Learning to Learn across Diverse Data Biases in Deep Face Recognition Supplementary Material

Chang Liu[1,2*] Xiang Yu[2] Yi-Hsuan Tsai[2] Masoud Faraki[2] Ramin Moslemi[2]
Manmohan Chandraker[2,3] Yun Fu[1]
[1]Northeastern University [2]NEC Labs America [3]University of California, San Diego
[1]{liu.chang6, yunfu}@ece.neu.edu, [2]{xiangyu, ytsai, mfaraki ,rmoslemi, manu}@nec-labs.com

In this material, we firstly show in Sec. A that the re-weighting method Eqn. 1 can be interpreted as the CosFace loss with new scale $\sigma_{y_j}\boldsymbol{s}$ and new margin $\sigma_{y_j}\bar{\boldsymbol{m}}$. Then, we illustrate our proposed meta-updating module in Sec B with more details. A complement implementation detail to our main submission experiment section, is in Sec C to provide more information for the reproduction purpose. To provide a finer-scale visualization of our Fig. 1 in the main submission, we show the multiple long-tailed distribution plots regarding per-class volume, head pose and ethnicty in Sec. D. In Sec. E, we visualize the instance-level variation-aware margin with more samples as an extension of our main submission Fig. 3, which is expected to provide a more complete view of the visualization. Sec. F re-iterates our claiming regarding ethics concern and finally Sec. G abstracts the reproducibility of our method.

A. Sample Importance Interpreted as Cosine Loss Margin

In our main submission Sec. 3.1, we show that the sample importance from the re-weighting method (Eqn. 1) is equivalent to the Cosine Loss. In this section, we provide a more detailed proof showing that actually the re-weighting objective (Eqn. 1) can be mathematically approximated by the Cosine Loss.

Connecting to the main submission, we introduce the importance weight σ_{y_j} to re-weight each sample loss as the following:

$$\min_{\Omega} \quad \frac{1}{N}\sum_{j=1}^{N}\sigma_{y_j}\mathcal{L}_{cos}(f(x_j;\Omega),y_j,\mathbf{s},\bar{\mathbf{m}}) \tag{1}$$

In the Cosine Loss, we omit the input feature $f(x_j;\Omega)$, y_j and only reserve $\mathbf{s},\bar{\mathbf{m}}$ in the loss notation:

$$\mathcal{L}_{cos}(\mathbf{s},\bar{\mathbf{m}}) = -log\frac{e^{\mathbf{s}\cdot\cos\theta_{y_j}-\bar{\mathbf{m}}}}{e^{\mathbf{s}\cdot\cos\theta_{y_j}-\bar{\mathbf{m}}}+\sum_{y_k\neq y_j}^{C}e^{\mathbf{s}\cdot\cos\theta_{y_k}}} \tag{2}$$

*This work was conducted as part of a summer internship at NEC Labs America.

Combining Eqn. 1 with Cosine Loss Eqn. 2, we obtain:

$$\min_{\Omega}\frac{1}{N}\sum_{j=1}^{N}-log\frac{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}\right]^{\sigma_{y_j}}}{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}+\sum_{k\neq y_j}^{C}e^{\mathbf{s}\cdot cos\theta_{y_k}}\right]^{\sigma_{y_j}}} \tag{3}$$

When the training with Cosine Loss towards convergence, we know that $\theta_{y_j}\approx 0$ and $\theta_{y_k}\approx\frac{\pi}{2}$ and thus the denominator is close to:

$$\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}+\sum_{k\neq y_j}^{C}e^{\mathbf{s}\cdot cos\theta_{y_k}}\right]^{\sigma_{y_j}} = \left[e^{\mathbf{s}-\bar{\mathbf{m}}}+C-1\right]^{\sigma_{y_j}} \tag{4}$$

Similarly, when consider another Cosine Loss whose scale and margin are $\sigma_{y_j}\mathbf{s}$ and $\sigma_{y_j}\bar{\mathbf{m}}$, we obtain:

$$\mathcal{L}_{cos}(\sigma_{y_j}\mathbf{s},\sigma_{y_j}\bar{\mathbf{m}}) = -log\frac{e^{\sigma_{y_j}(\mathbf{s}\cdot\cos\theta_{y_j}-\bar{\mathbf{m}})}}{e^{\sigma_{y_j}(\mathbf{s}\cdot\cos\theta_{y_j}-\bar{\mathbf{m}})}+\sum_{y_k\neq y_j}^{C}e^{\sigma_{y_j}\mathbf{s}\cdot\cos\theta_{y_k}}} \tag{5}$$

The denominator part can be similarly approximated as Eqn. 4 when converging:

$$e^{\sigma_{y_j}(\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}})}+\sum_{k\neq y_j}^{C}e^{\sigma_{y_j}\mathbf{s}\cdot cos\theta_{y_k}} = e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}+C-1 \tag{6}$$

Then, consider the ratio between Eqn. 4 and Eqn. 6, $F(\sigma_{y_j}) = \frac{\left[e^{\mathbf{s}-\bar{\mathbf{m}}}+C-1\right]^{\sigma_{y_j}}}{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}+C-1}$, we investigate the ratio limit when σ_{y_j} ranges from negative infinite to positive infinite:

$$\lim_{\sigma_{y_j}\to-\infty}\frac{\left[e^{\mathbf{s}-\bar{\mathbf{m}}}+C-1\right]^{\sigma_{y_j}}}{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}+C-1} = \frac{0}{C-1} = 0 \tag{7}$$

$$\lim_{\sigma_{y_j}\to+\infty}\frac{\left[e^{\mathbf{s}-\bar{\mathbf{m}}}+C-1\right]^{\sigma_{y_j}}}{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}+C-1} = \frac{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}}{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}} = 1 \tag{8}$$

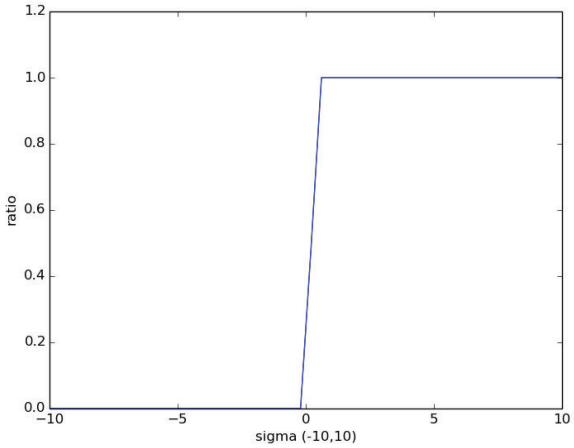

Figure 1. Ratio of $\frac{\left[e^{\mathbf{s}-\bar{\mathbf{m}}}+C-1\right]^{\sigma_{y_j}}}{e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})}+C-1}$ with respect to σ_{y_j}. $\mathbf{s} = 64, C = 8e4, \bar{\mathbf{m}} = 64 * 0.3 = 19.2$.

Further, the derivative of function F with respect to σ_{y_j} is:

$$F'(\sigma) = \frac{(e^u + c)^\sigma \left((\ln(e^u + c) - u)e^{u\sigma} + c\ln(e^u + c)\right)}{\left[e^{u\sigma} + c\right]^2} > 0, since\, \ln(e^u + c) \geq u, \ln(e^u + c) > 0 \tag{9}$$

where we use σ to denote σ_{y_j}, $u = \mathbf{s} - \bar{\mathbf{m}}$, $c = C - 1$ for simplicity. We see that $F(\sigma)$ is a monotonically increasing function in $(-\infty, +\infty)$.

Hence, for a given $\sigma_{y_j} > 0$, we have:

$$\left[e^{\mathbf{s}-\bar{\mathbf{m}}} + C - 1\right]^{\sigma_{y_j}} \leq \left[e^{\sigma_{y_j}(\mathbf{s}-\bar{\mathbf{m}})} + C - 1\right] \tag{10}$$

We also numerically plot the ratio curve $F(\sigma_{y_j})$ with respect to σ_{y_j} in Fig. 1, which verifies that the function is monotonically increasing between 0 and 1.

From Eqn. 3, we consider Eqn. 10 and replace the original denominator of Eqn. 3 as:

$$\begin{aligned}
&\frac{1}{N}\sum_{j=1}^{N}\sigma_{y_j}\mathcal{L}_{cos}(f(x_j;\Omega), y_j, \mathbf{s}, \bar{\mathbf{m}}) \\
&= \frac{1}{N}\sum_{j=1}^{N} -log\frac{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}\right]^{\sigma_{y_j}}}{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}} + \sum_{k\neq y_j}^{C} e^{\mathbf{s}\cdot cos\theta_{y_k}}\right]^{\sigma_{y_j}}} \\
&\leq \frac{1}{N}\sum_{j=1}^{N} -log\frac{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}\right]^{\sigma_{y_j}}}{\left[e^{\mathbf{s}cos\theta_{y_j}-\bar{\mathbf{m}}}\right]^{\sigma_{y_j}} + \sum_{k\neq y_j}^{C} e^{\sigma_{y_j}\mathbf{s}\cdot cos\theta_{y_k}}} \\
&= \frac{1}{N}\sum_{j=1}^{N}\mathcal{L}_{cos}(f(x_j;\Omega), y_j, \sigma_{y_j}\mathbf{s}, \sigma_{y_j}\bar{\mathbf{m}})
\end{aligned} \tag{11}$$

The inequality step is based on Eqn. 10. Thus, we proved that the re-weighting objective with the original Cosine Loss Eqn. 3 is upper bounded by a new Cosine Loss in Eqn. 5. Further, by minimizing the new Cosine Loss (Our MvCoM), we can guarantee that the original re-weighting objective Eqn. 3 is minimized. □

B. Meta-learning Update Details

As shown in Fig. 2 (main submission Fig. 2), there are mainly three updating steps in our meta-updating procedure. We hereby specify one typical loop of the proposed meta learning framework from the gradient perspective as the following:

$$\tilde{\Omega}^{t+1}(\boldsymbol{r}^t) : \Omega^t - \eta\frac{\partial\sum_{j\in T}\mathcal{L}_{MvCoM}(f(x_j;\Omega^t), y_j; m_{y_j} + \boldsymbol{r}_j^t)}{\partial\Omega} \tag{12}$$

$$\boldsymbol{r}^{t+1} : \boldsymbol{r}^t - \tau\frac{\partial\sum_{k,j\in V}\mathcal{L}_{var}^k(f(x_j;\tilde{\Omega}^{t+1}(\boldsymbol{r}^t)), \mu_j^k)}{\partial\boldsymbol{r}} \tag{13}$$

$$\Omega^{t+1} : \Omega^t - \eta\frac{\partial\sum_{j\in T}\mathcal{L}_{MvCoM}(f(x_j;\Omega^t), y_j; m_{y_j} + \boldsymbol{r}_j^{t+1})}{\partial\Omega} \tag{14}$$

It corresponds to three steps of the update in Sec. 3.2.2 of the main submission. First, we feed a training batch into the deep face recognition model to compute identification loss with a prior class-aware margin where margin residual is initialized as zero and update the model in Eqn. 12 to a temporary model which is termed as "Pseudo updated" model in the main submission. Then, following our hard sampling rule, we sample a meta-learning batch which is the most distinct from the training batch, feed it into the pseudo updated model, compute the classification loss for 4 variation tasks and update the margin residual in Eqn. 13. It is worth noting that the meta-updated margin residual carries the compensating information to each long-tailed variation factors compared to the previous margin. Last, we add the meta-updated margin residual on top of previous margin prior, go back to the first step to compute the identification loss and update the original pre-updated model with newly meta-learned margin in Eqn. 14.

In a nutshell, the purpose of meta-learning framework is to learn the best margin (or margin residual $\boldsymbol{r}$) in the sense that it results in smaller classification errors in variation tasks on meta-learning set by balancing multiple variation factors.

C. More Implementation Details

The whole code base is implemented with Pytorch v1.1. We use the clean list from ArcFace [2] for MS-Celeb-1M [3]

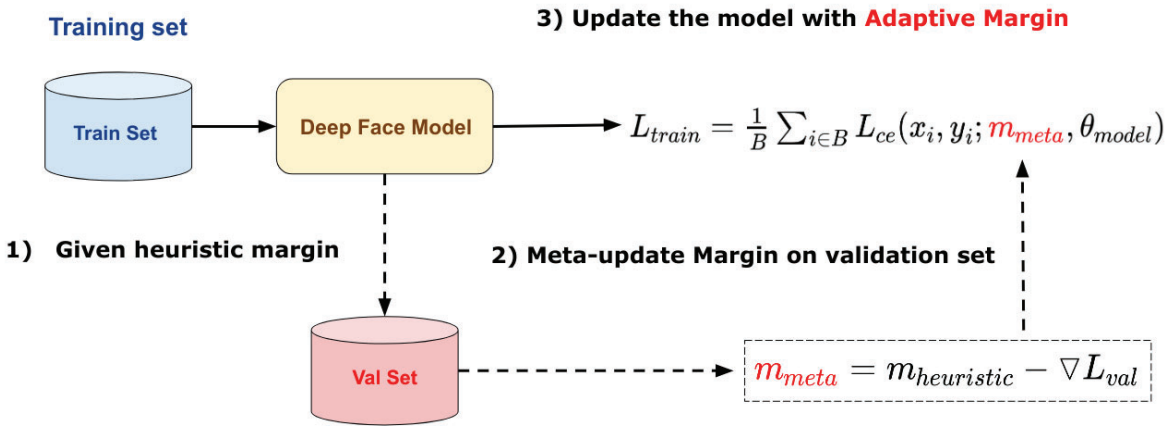

Figure 2. The meta-learning update three stages. (1) recognition model pseudo update. (2) MvCoM meta-update with pseudo recognition model. (3) recognition model real update with updated MvCoM.

as the training data. For the meta-training set, we adopt VGGFace2 [1] and exclude the duplicate identities since it contains multiple variation factors that can potentially benefit the training dataset. The baseline models in the experiments are trained with CosFace loss [4] for 30 epochs with empirically fixed margin $m = 0.35$. After pre-training, we discard the classifier and fine-tune the models with the proposed framework for 18 epochs to ensure convergence.

Pre-training Specifically, we adopt the 100-layer ResNet proposed in [4] as our embedding network Ω. As introduced in the main paper, our training of the face recognition engine is divided into two steps. For the first step, we pre-train a CosFace-like engine backbone. The training data is the cleaned MS-Celeb-1M [3] with 84K identities and around $4.8M$ images. We train the CosFace model with 30 epochs with initial learning rate 0.1 Then, the learning rate is multiplied by 0.1 at the 14^{th}, 20^{th} and 23^{th} epoch each time. The momentum is set as 0.9 with weight decay as $5e^{-4}$. We use SGD as the optimizer. Batch size is set to 512. The overall training is conducted on a 8-core Titan-X gpu with pytorch parallel model training.

Alternative Meta-Optimization After pre-training, we keep the whole backbone engine from the first step. For the identification classifier (a fully-connected layer with 512×84K dimension), we also keep it for our main task identification. Meanwhile, as introduced in the main paper, we introduce 4 variation task classifiers, namely a pose classifier (a fully-connected layer with 512×7 dimension), an occlusion classifier (a fully-connected layer with 512×5 dimension), a blur classifier (a fully-connected layer with 512×4 dimension) and an ethnicity classifier (a fully-connected layer with 512×4 dimension). There are typically two rounds of forward pass and backward pass in one iteration of the fine-tuning. For the first round, a training batch is fed to the recognition backbone and calculate the identification loss via identification classifier, which is known as "Pseudo Update" explained in the methodology part of the main paper. Then, a backward pass is conducted to update the network parameter Ω. After the first round, we meta-update the residual of the margin for each instance in the training batch. For the second round, the updated margin is contributed to calculate the loss and the recognition model is updated. Such process lasts 18 epochs to ensure convergence. We initialize the learning rate as 0.001 and reduce by 0.1 at every 8^{th}, 14^{th} and 16^{th} epochs. The momentum is set as 0.9 with weight decay as $5e^{-4}$. We use SGD as the optimizer. Batch size is set to 512, the same as the pre-training step.

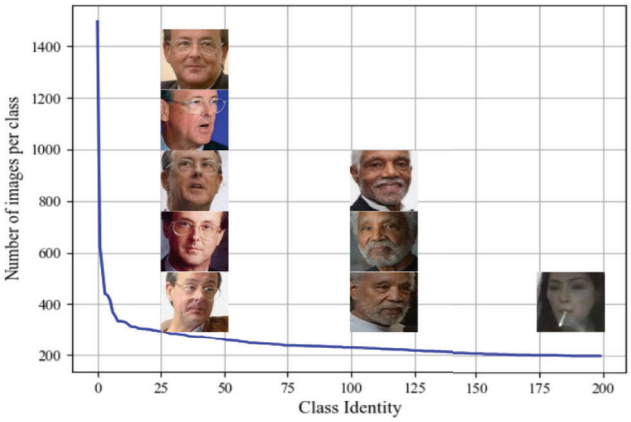

Figure 3. Volume per-class long-tailed distribution.

D. Long-tailed Distribution Visualization

We exhibit the long-tailed distribution in a finer scale besides the Fig. 1 in our main submission. The long-tailed

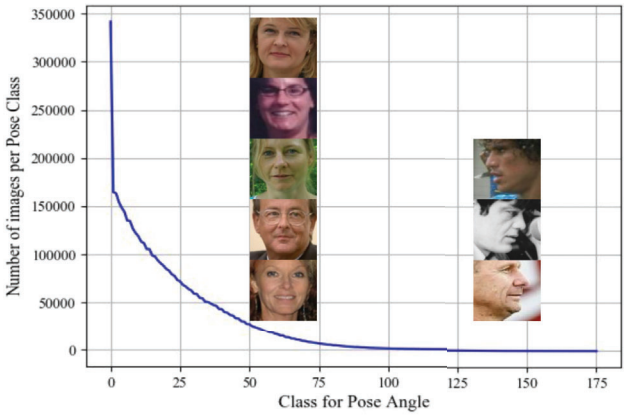

Figure 4. Pose long-tailed distribution.

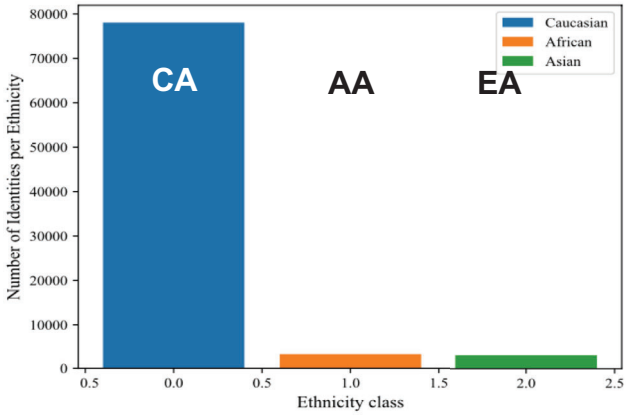

Figure 5. Ethnicity long-tailed distribution.

distribution is visualized in terms of class volume, head pose and ethnicity in Fig. 3, Fig. 4 and Fig. 5, respectively. We observe that the training distribution is not only long-tailed in class volume, but also in other variation factors such as ethnicity and head poses. This motivates us to consider multiple long-tailed factors into a single framework.

E. Per-Sample Margin Visualization

In this section, we visualize the per-sample margin for more instances from MS-Celeb-1M [3] to verify our learned margin is capable of compensating the distribution imbalance in terms of multiple variation factors. In Fig. 6, we observe that our model consistently assigns larger margin weight to the samples from the tailed class of each variation such as non-Caucasian, large pose, blur and occluded images.

F. Ethics Statement

As mentioned in our main submission "Discussions and Conclusions" section, we would like to acknowledge that all the subjects' images utilized in our paper, have obtained consent from all the subjects by the public dataset providers. We will remove the subjects' face images if required from any privacy concern that is not properly defined in the dataset providers' consent. We acknowledge that the potential use of face recognition technology can lead to unlawful surveillance and discrimination, which should be regulated by the laws. A famous example is a regular African American face image is mistakenly recognized as a gorilla by Google inc. face recognition engine. This exemplifies that bias in deep learning models indeed can result in negative unexpected social and ethical impacts.

Exactly to mitigate this, our work has the positive benefit of alleviating such critical concern with face recognition and its biases, which have been observed to have detrimental consequences in health, hiring, policing and judicial outcomes. For instance, the advocation of the proposed method can help to improve face recognition technology in less minority ethnicity discrimination, lower risk of improperly recognizing faces due to low quality images, large head poses and heavy face occlusion. We believe our work is in the venue of aligning the research directions to be consistent to the social and ethical requirements, such as towards less bias in ethnicity or geographic factor, which is a key existing problem across almost all of the public face datasets.

G. Reproducibility

We highlight all the components within this submission that effectively support the reproducibility of this work. **1)** Implementation details are provided in main submission Sec. 4, as well as a more elaborated implementation details in Sec. C, where the training dataset, backbone architecture, training schemes, variation label preparation and the running complexity are carefully analyzed. **2)** A more detailed theoretical analysis regarding our methodology is presented in main Submission Sec. 3.1 and Sec. A. **3)** A better understanding of our method, that is, more visualizations on the long-tailed distributions (Fig. 3, 4 and 5), as well as per-sample level multi-variation cosine margin (MvCoM)(Fig. 6) are visualized.

References

[1] 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. 3

[2] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CVPR*, 2019. 2

[3] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016. 2, 3, 4

[4] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *CVPR*, 2018. 3

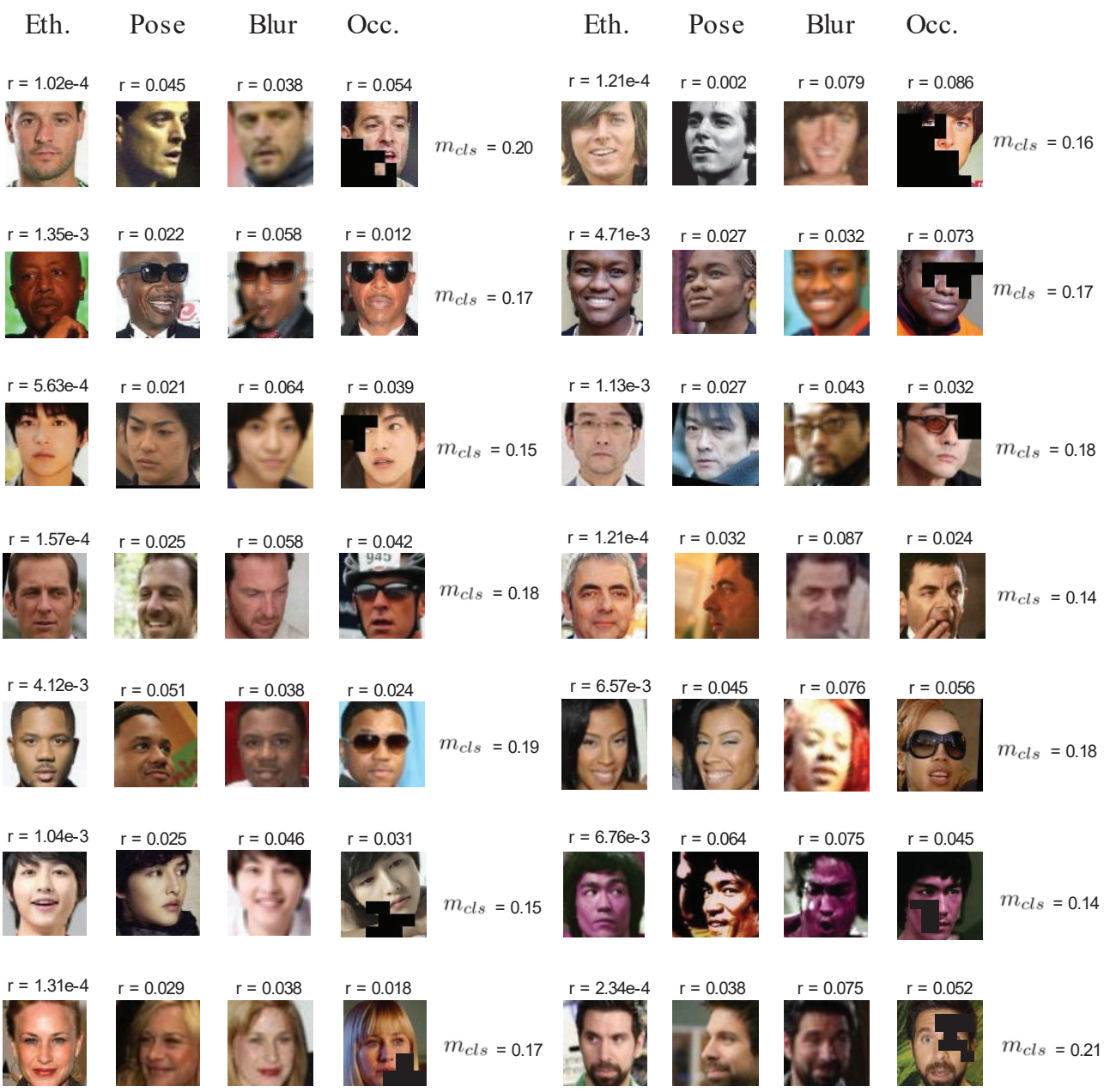

Figure 6. Multi-variation Cosine Margin (MvCoM) visualization across all the factors,i.e. ethnicity, quality, pose, blur and occlusion.