Unified Negative Pair Generation toward Well-discriminative Feature Space for Face Recognition

Junuk Jung
rnans33@koreatech.ac.kr
Seonhoon Lee
seonhoon1002@koreatech.ac.kr
Heung-Seon Oh
ohhs@koreatech.ac.kr
Yongjun Park
qkr2938@koreatech.ac.kr
Joochan Park
green669@koreatech.ac.kr
Sungbin Son
sbson0621@koreatech.ac.kr

School of Computer Science and Engineering
Korea University of Technology and Education (KOREATECH)

Abstract

The goal of face recognition (FR) can be viewed as a pair similarity optimization problem, maximizing a similarity set $\mathcal{S}^p$ over positive pairs, while minimizing similarity set $\mathcal{S}^n$ over negative pairs. Ideally, it is expected that FR models form a well-discriminative feature space (WDFS) that satisfies $\inf \mathcal{S}^p > \sup \mathcal{S}^n$. With regard to WDFS, the existing deep feature learning paradigms (i.e., metric and classification losses) can be expressed as a unified perspective on different pair generation (PG) strategies. Unfortunately, in the metric loss (ML), it is infeasible to generate negative pairs taking all classes into account in each iteration because of the limited mini-batch size. In contrast, in classification loss (CL), it is difficult to generate extremely hard negative pairs owing to the convergence of the class weight vectors to their center. This leads to a mismatch between the two similarity distributions of the sampled pairs and all negative pairs. Thus, this paper proposes a unified negative pair generation (UNPG) by combining two PG strategies (i.e., MLPG and CLPG) from a unified perspective to alleviate the mismatch. UNPG introduces useful information about negative pairs using MLPG to overcome the CLPG deficiency. Moreover, it includes filtering the similarities of noisy negative pairs to guarantee reliable convergence and improved performance. Exhaustive experiments show the superiority of UNPG by achieving state-of-the-art performance across recent loss functions on public benchmark datasets. Our code and trained models are publicly available[1].

[1] https://github.com/Jung-Jun-Uk/UNPG.git

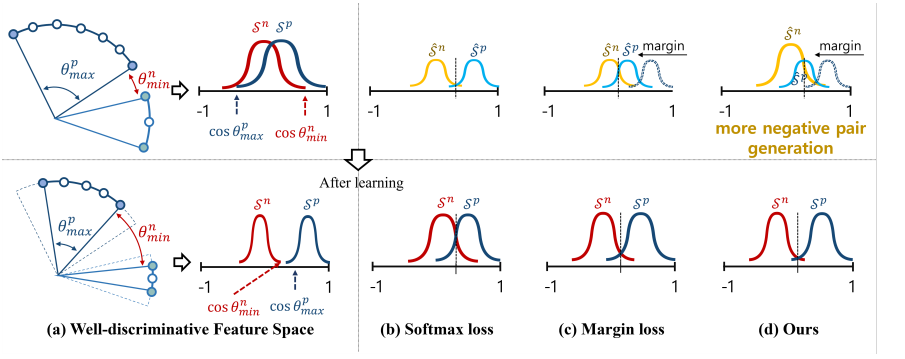


Figure 1: Similarity distributions viewed from pair generation perspective for approximating WDFS. The bottom line presents similarity distributions in feature space after sufficiently learning in their own ways with the top line. $\mathcal{S}^p$ and $\mathcal{S}^n$ are positive and negative similarity sets and $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ are subsets of $\mathcal{S}^p$ and $\mathcal{S}^n$, respectively. (a) The ideal similarity sets satisfying $\inf \mathcal{S}^p > \sup \mathcal{S}^n$ after learning with $\mathcal{S}^p$ and $\mathcal{S}^n$. θ^p_{max} and θ^n_{min} are the max and min angles among positive and negative pairs. (b) Using a softmax loss, no overlap between $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ results in an overlap between $\mathcal{S}^p$ and $\mathcal{S}^n$. (c) Using a marginal loss, an overlap between $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ by shifting $\hat{\mathcal{S}}^p$ reduces an overlap after learning. (d) Using more negative pairs, an overlap between $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ by shifting $\hat{\mathcal{S}}^p$ and enlarging $\hat{\mathcal{S}}^n$ significantly reduces the overlap after learning.

1 Introduction

The goal of face recognition (FR) can be viewed as a pair similarity optimization problem that maximizes a similarity set $\mathcal{S}^p$ over positive pairs while minimizing a similarity set $\mathcal{S}^n$ over negative pairs. Regardless of FR tasks such as face verification (1:1) and face identification (1:N), it is expected that FR models form a well-discriminative feature space (WDFS) that satisfies $\inf \mathcal{S}^p > \sup \mathcal{S}^n$ as shown in Fig. 1 (a). To this end, previous research advances pair similarity optimization[5, 18, 20, 30, 36, 37, 41] by enhancing intra-class compactness and inter-class dispersion.

In deep feature learning paradigms for pair similarity optimization, loss functions in FR can be categorized based on two approaches: metric loss (ML; e.g., triplet loss[9, 25] and N-pair loss[28]) and classification loss (CL; e.g., softmax loss[7, 23, 32]). The former directly performs the optimization with a pair of deep feature vectors using a pair-wise label whereas the latter performs indirectly with a pair of deep feature and class weight vectors using a class-level label. Recently, in circle loss[30], two different approaches were expressed as a unified loss function since their intent and behaviors pursued the same objective of maximizing a similarity set $\mathcal{S}^p$ over positive pairs, while minimizing it over negative pairs. We decomposed the unified loss function into pair generation (PG) and similarity computation (SC) without loss of generality. While SC focuses on computing the similarity between two samples in a pair, PG focuses on generating a pair using vectors of deep features or class weights. In the unified loss function, the only difference between ML and CL is PG, because various methods in SC can be applied to both ML and CL in the same manner. Consequently, the core of FR research from a unified perspective is the generation of informative pairs, i.e., PG. This is crucial because only a limited number of pairs are trainable in each iteration

owing to the large computational costs incurred. Based on the assumption that pairs sampled from mini-batches can represent the entire feature space, the existing methods decrease the loss to a pair as it approaches WDFS, whereas they increase the loss in the opposite case.

We observed the reason why FR models trained sufficiently fail to approach WDFS. This stems from the mismatch of similarity distributions between the sampled pairs and all pairs. Fig. 1 (b) shows an example of two similarity sets $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ of positive and negative pairs, respectively, sampled from mini-batches. Even though feature space is far from WDFS by highly overlapped $\mathcal{S}^p$ and $\mathcal{S}^n$, a FR model is rarely trainable with $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ because they are well-separated with almost no overlap so that an only slight loss is assigned. To deal with this problem, a line of research[5, 18, 20, 30, 36, 37, 41] devises marginal loss functions to reduce the gap by shifting $\hat{\mathcal{S}}^p$, as shown in Fig. 1 (c). In general, marginal CL functions show better performance than ML functions on large-scale datasets[6]. However, there still exists a mismatch between the sampled negative pairs and all negative pairs because it is difficult to obtain too-hard negative pairs.

This paper proposes unified negative pair generation (UNPG) by combining two PG strategies (i.e., MLPG and CLPG) from a unified perspective to alleviate the mismatch. Moreover, it includes filtering noise-negative pairs, such as too-easy/hard negative pairs, in order to guarantee reliable convergence and improve performance. Consequently, UNPG helps approach WDFS, as shown in Fig. 1 (d). Through experiments, we demonstrate the superiority of UNPG by achieving state-of-the-art performance using recent loss functions equipped with UNPG on public benchmark datasets (IJB-B[39], IJB-C[19], MegaFace[14], LFW[11], CFPFP[26], AgeDB-30[21], CALFW[43], and K-FACE[3]) and deliver an in-depth analysis of the reasons behind UNPG.

2 Related Works

FR is one of the most promising computer-vision tasks. In recent times, the combination of the following three factors has contributed to the rapid growth of this technology: 1) introduction to large-scale face datasets[6, 29, 40], 2) development of effective backbone models[8, 10, 16, 27, 31], 3) novel loss functions[5, 20, 36]. Among them, loss functions have been actively developed and can be categorized into metric and classification losses.

Metric Loss. Early direct optimization methods include contrastive loss[4, 7] and triplet loss[9, 25], which use the similarity between pairs or triplets in the feature space. They try to make positive samples close and push negative samples far away, but often suffer from slow convergence and poor local optima because they only learn 1:1 relationships in positive and negative pairs. Thus, lifted-structure loss[22] and N-pair loss[28] were designed to address this issue. They build massive negative samples and one positive sample based on the same anchor point and learn their relationship simultaneously. Subsequently, other methods have been developed to create more informative pairs. Multi-similarity loss[37] classifies existing studies into three types of similarities and devises pair mining and pair weighting methods that satisfy them simultaneously. Tuplet-margin loss[41] provides a slack margin to prevent overfitting from hard triplets. Despite these efforts, ML still faces a problem: The negative pairs generated by each iteration cannot represent all identities because FR datasets[6, 29, 40] usually have more classes than a mini-batch size.

Classification Loss. Early indirect optimization methods include softmax loss[2, 23, 32], which uses the similarity between the deep feature and class weight vectors. Softmax loss has been widely applied in classification problems, but it is not appropriate for FR because

testing is done by similarity comparison, not classification. Hence, two methods are being investigated to modify the softmax logit to form a feature space suitable for FR. The first is the normalization of the deep feature and class weight vectors[24, 35, 36] to reduce the gap between the training and test phases mapped to the cosine similarity space. This is motivated by the interpretation of the feature space of studies such as center-loss[38], L-softmax[17], and NormFace[34]. The second is a margin assignment technique[1, 5, 12, 15, 18, 20, 35, 36], which is performed in various ways to enhance intra-class compactness and inter-class dispersion. CosFace[36] and ArcFace[5], which are typical margin-based loss functions, add external and internal margins to cosine angles, respectively. ElasticFace[1] extended CosFace and ArcFace by using random margin penalty values derived from a Gaussian distribution rather than a fixed margin. CurricularFace[12] adopted curriculum learning that automatically injects adaptive margins based on the difficulty level of samples and training time. MagFace[20] introduced a new margin and regularizer technique within several constraints that assumed a positive correlation between magnitude and face quality and ensured convergence. AdaFace[15] is similar to MagFace but different in using normalized image quality indicators when calculating adaptive margins. They improved FR performance by creating discriminative features. However, in our interpretation, there is a problem that extremely hard negative similarities in the feature space cannot be expressed by the indirect optimization method alone.

Multi-Objective Loss & Unified Loss. Multi-objective loss tried to combine two different losses with a mixture weight at the surface level. MixFace[13] attempted to combine the metric and classification losses (i.e., $\mathcal{L}_{mix} = \mathcal{L}_{arc} + \mathcal{L}_{sn-pair}$) with an analysis of their advantages and disadvantages. However, it is a mixture at the surface level and not a unified loss function. According to the Circle-loss[30], the two existing approaches (i.e., metric and classification losses) can be expressed as a unified loss function. It also adds independent weight factors to deal with ambiguous convergence but is limited in generating pairs (e.g., Circle-loss[30] used only MLPG).

3 Methodology

Unified Loss. According to [28, 30], the classification and metric losses can be expressed as a unified loss function (i.e., cross-entropy loss). Suppose that $\hat{\mathcal{S}}^p = \{s_i^p | i = 1,2,...,K\}$ and $\hat{\mathcal{S}}^n = \{s_j^n | j = 1,2,...,L\}$ are the similarity scores for K positive and L negative pairs, respectively. Then, the unified loss function is defined as:

$$\begin{aligned} \mathcal{L}^{uni} &= \frac{1}{K}\sum_{i=1}^{K}\mathcal{L}_i^{uni}, \\ \mathcal{L}_i^{uni} &= -\log\frac{e^{\gamma s_i^p}}{e^{\gamma s_i^p}+\sum_{j=1}^{L}e^{\gamma s_j^n}} = \log\left[1+\sum_{j=1}^{L}e^{\gamma(s_j^n - s_i^p)}\right] \end{aligned} \tag{1}$$

where γ is a positive scale factor.

The only difference between the two losses is the method of computing $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$. We break down this step into PG and SC to clearly explain our proposed method without loss of generality.

Pair Generation (PG). In a feature space, let us assume that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are i-th and j-th samples in N-sized mini-batch and y_i and y_j are the corresponding indexes of target classes

in a total of C classes. $\boldsymbol{w}_c$ is a weight vector of c-th class. Then, we generate positive and negative pair sets $\mathcal{P}$ and $\mathcal{N}$ for the metric (Eq. 2) and classification (Eq. 3) losses, respectively:

$$\begin{aligned}\mathcal{P}^{ml} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j = y_i\} \\ \mathcal{N}^{ml} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j \neq y_i\}\end{aligned} \tag{2}$$

$$\begin{aligned}\mathcal{P}_i^{cl} &= (\boldsymbol{x}_i, \boldsymbol{w}_{y_i}) \\ \mathcal{N}_i^{cl} &= \{(\boldsymbol{x}_i, \boldsymbol{w}_c) | c \neq y_i\}\end{aligned} \tag{3}$$

In ML, a pair is composed of two samples (e.g., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$) from a mini-batch, and is composed of a sample and a weight vector (e.g., $\boldsymbol{x}_i$ and $\boldsymbol{w}_c$) in CL. We denote MLPG and CLPG for PG of the metric and classification losses, respectively.

Similarity Computation (SC). We can compute the similarity sets $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ obtained from PG. The metric and classification losses employ the same similarity method for the same type of pair sets (i.e., positive sets $\mathcal{P}^{ml}$ and $\mathcal{P}^{cl}$ and negative sets $\mathcal{N}^{ml}$ and $\mathcal{N}^{cl}$). Recent research has focused on improving the cosine similarity using a margin. Let us define the angle between two vectors as $\Theta(\boldsymbol{a}, \boldsymbol{b}) = \arccos(\boldsymbol{a}^\top \boldsymbol{b} / \|\boldsymbol{a}\| \|\boldsymbol{b}\|)$. Then, SN-pair[13] computes $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ for $\mathcal{P}^{ml}$ and $\mathcal{N}^{ml}$ as:

$$\begin{aligned}\hat{\mathcal{S}}^p &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j = y_i\} \\ \hat{\mathcal{S}}^n &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j \neq y_i\}\end{aligned} \tag{4}$$

Note that the $\hat{\mathcal{S}}^n$ of Eq. 4 generates the similarities of all negative pairs in the mini-batch, not just the similarities between the anchor and the negative ones, to approach the WDFS (see. Eq. 9 and Eq. 10).

There is a line of research[5, 20, 36] that employs a margin in cosine similarity. In CosFace [33], margin m is placed outside cosine for $\hat{\mathcal{S}}^p$. Thus, $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ are computed for $\mathcal{P}_i^{cl}$ and $\mathcal{N}_i^{cl}$ as:

$$\begin{aligned}\hat{\mathcal{S}}_i^p &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{w}_{y_i}) + m\} \\ \hat{\mathcal{S}}_i^n &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{w}_c) | c \neq y_i\}\end{aligned} \tag{5}$$

On the other hand, ArcFace[5] places margin m inside cosine:

$$\hat{\mathcal{S}}_i^p = \{\cos(\Theta(\boldsymbol{x}_i, \boldsymbol{w}_{y_i}) + m)\} \tag{6}$$

In other research using margins such as SphereFace[18] and MagFace[20], $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ can be derived similarly without loss of generality.

Unified Negative Pair Generation (UNPG). We address the fact that PG is the only difference between metric and classification losses from a unified perspective. Previous studies[5, 18, 20, 36, 37, 41] attempted to reduce the gap between $\mathcal{S}^p$ and $\hat{\mathcal{S}}^p$ by devising various margin-based methods. Evidently, there is no concern about the gap between $\mathcal{S}^n$ and $\hat{\mathcal{S}}^n$ even though it is a critical component in computing a loss. There are several reasons that cause the gap between $\mathcal{S}^n$ and $\hat{\mathcal{S}}^n$. In ML, it is infeasible to generate negative pairs taking all classes into account in each iteration because of the limited mini-batch size.

In CL, it is difficult to generate too-hard negative pairs owing to the convergence of the class weight vectors to their center. This paper proposes unified negative pair generation (UNPG) by combining MLPG and CLPG strategies from a unified perspective to alleviate the mismatches of $(\mathcal{S}^n, \hat{\mathcal{S}}^n)$ and $(\mathcal{S}^p, \hat{\mathcal{S}}^p)$, together. UNPG introduces useful information about negative pairs using MLPG to overcome the CLPG deficiency. In UNPG, a negative pair set $\mathcal{N}_i^{uni}$ and the corresponding similarity set $\hat{\mathcal{S}}_i^n$ are defined as:

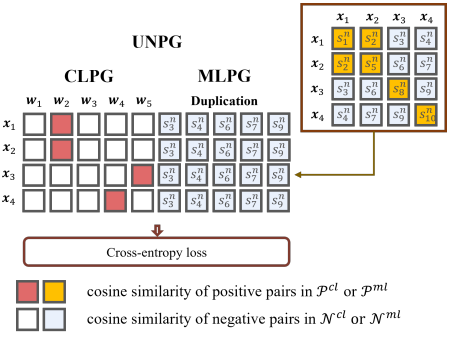


Figure 2: Unified loss with UNPG.

$$\begin{aligned} \mathcal{N}_i^{uni} &= \mathcal{N}_i^{cl} \cup \mathcal{N}^{ml} \\ \hat{\mathcal{S}}_i^n &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{w}_c) | c \neq y_i\} \cup \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j \neq y_i\} \end{aligned} \tag{7}$$

$\hat{\mathcal{S}}_i^n$ can be computed from $\mathcal{N}_i^{uni}$ using various methods such as Eqs. 4 and 5. Decomposing SC and PG can lead to wide research directions in FR. As a result, the unified loss with UNPG is defined as:

$$\mathcal{L}_i^{uni} = -\log P_i = -\log \frac{e^{\gamma s_i^p}}{e^{\gamma s_i^p} + \sum_{j=1}^{L^{cl}} e^{\gamma s_j^n} + \sum_{k=1}^{L^{ml}} e^{\gamma s_k^n}} \tag{8}$$

where $L^{cl} = |\mathcal{N}_i^{cl}|$ and $L^{ml} = |\mathcal{N}^{ml}|$. Note that the normalization term in Eq. 8 uses the scores from $\mathcal{N}^{ml}$. Fig. 2 visualizes Eq. 8, where UNPG uses the similarity score matrix obtained from $\mathcal{N}^{ml}$ at each mini-batch and then duplicates it by the size of the mini-batch.

Whether CL or ML, many loss functions (ArcFace[5], CosFace[36], triplet-loss[9], N-pair loss[28] etc.,) induce the similarity between the anchor and the positive sample to be higher than the similarity between the anchor and the negative sample. These methods might be useful in image retrieval tasks that employ recall@k between anchors and other samples as an evaluation protocol to reduce the gap between training and testing. However, it is not appropriate for tasks such as face verification (WDFS: $\forall s_i^p > \forall s_j^n \iff \inf \mathcal{S}^p > \sup \mathcal{S}^n$), which intends the similarity between any two positive samples to be higher than any two negative samples. Therefore, the WDFS approach for FR needs to be intended from Eq. 10 as the proposed method, not Eq. 9.

$$\cos\Theta(\boldsymbol{x}_a, \boldsymbol{w}_{y_a}) > \{\cos\Theta(\boldsymbol{x}_a, \boldsymbol{w}_c) | c \neq y_a\} \tag{9}$$

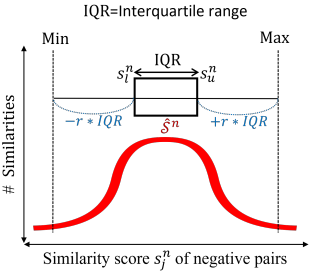
$$\cos\Theta(\boldsymbol{x}_a, \boldsymbol{w}_{y_a}) > \{\cos\Theta(\boldsymbol{x}_a, \boldsymbol{w}_c) | c \neq y_a\} \cup \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_j \neq y_i\} \tag{10}$$

Note that $\boldsymbol{x}_a$ is an anchor example and $\cos\Theta(\boldsymbol{x}_a, \boldsymbol{w}_{y_a})$ is deformable through various margin injecting techniques.

Noise Negative Pair Filtering. According to our preliminary experiments, directly utilizing $\mathcal{N}^{ml}$ produced by MLPG causes performance degradation and divergence of a loss because many noise-negative pairs cause a side-effect. We assumed that there are two types of noise pairs: too-easy and too-hard pairs. In the former case, FR models need not pay attention to the pairs but they do, owing to the size of the pairs. In the latter case, FR models cannot

Algorithm 1: Noise Negative Pair Filtering.

Input: $s_j^n \in \hat{\mathcal{S}}^n$ from $\mathcal{N}^{ml}$, wisker size r
Extract the lower 25% similarity s_l^n
Extract the upper 25% similarity s_u^n
$\text{IQR} = s_u^n - s_l^n$
$\text{Min} = s_l^n - r*\text{IQR}$, $\text{Max} = s_u^n + r*\text{IQR}$
$\tilde{\mathcal{N}}^{ml} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | (y_i \neq y_j) \wedge (\text{Min} \leq s_j^n \leq \text{Max})\}$
Output: $\tilde{\mathcal{N}}^{ml}$



allow them because they exceed the representation power of the models. To address this problem, we developed noise-negative pair filtering using a box and whisker algorithm[43].

As a result, UNPG adopting Algorithm 1 is defined as:

$$\begin{aligned} \mathcal{N}_i^{uni} &= \mathcal{N}_i^{cl} \cup \tilde{\mathcal{N}}^{ml} \\ \hat{\mathcal{S}}_i^n &= \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{w}_c) | c \neq y_i\} \cup \{\cos\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) | (y_j \neq y_i) \wedge (\text{Min} \leq s_j^n \leq \text{Max})\} \end{aligned} \tag{11}$$

4 Experiments

Datasets. For training, MS1M-V2[5] and K-FACE:T4[13] datasets were employed. MS1M-V2, a semi-automatically refined version of MS-Celeb-1M[6], has 5.8M images and 85K identities. K-FACE:T4 is a preprocessed version of K-FACE[3] utilized in MixFace[13] and has 3.8M images and 370 identities. For testing, several benchmark datasets (IJB-B[39], IJB-C[19], MegaFace[14], LFW[11], CFPFP[26], AgeDB-30[21], CALFW[43], and K-FACE:Q1-Q4[13]) were used to evaluate FR models. The implementation details are provided in the supplementary material.

4.1 Evaluation Results

Results on IJB-B and IJB-C. IJB-B consists of 21.8K images of 1,845 subjects and 55K frames of 7,011 videos. IJB-C, an extended version of IJB-B, contains 31.3K images of 3,531 subjects and 117.5K frames of 11,799 videos. 10K/8M and 19K/15M of positive/negative pairs in IJB-B and IJB-C were used for 1:1 verification. Owing to the severe imbalance between positive and negative pairs, performance was measured by TAR@FAR at different intervals such as [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]. As shown in Table 1, all FR models with UNPG improved at almost every interval compared to those without UNPG. In particular, TAR@(FAR=1e-4), an interval widely used in FR improved consistently. For example, Mag+UNPG obtained gains of 1.22% and 0.84% in IJB-B and IJB-C, respectively, compared to MagFace, and gains of 0.7% and 0.41%, respectively, compared to MagFace*.

Results on LFW, CFP-FP, AgeDB-30, and CALFW. FR on LFW, CFP-FP, AgeDB-30, and CALFW is straightforward. Thus, the performance was saturated. LFW, AgeDB-30, and CALFW contain 6,000 images, and CFP-FP has 6,000 images. They have 1:1 ratios between the positive and negative pairs. Verification accuracy was employed with the best threshold separating the positive and negative pairs. In Table 2, the FR models with UNPG obtained competitive performance on the four datasets.

Results on MegaFace. MegaFace consists of a gallery set of 1M images with 690K classes and probe photos of 100K images with 530 classes. We followed the test protocol of ArcFace[5]. We removed noisy images and measured rank-1 accuracy for the 1M distractor after following the identification scenarios using the devkit provided by MegaFace. Table 3 presents the results of this study. FR models with UNPG performed better than those without it. ArcFace and CosFace using UNPG obtained gains of 0.26% and 0.19%, respectively, compared to those without it.

Method	Backbone	IJB-B(TAR@FAR)					IJB-C(TAR@FAR)				
		1e-6	1e-5	1e-4	1e-3	1e-2	1e-6	1e-5	1e-4	1e-3	1e-2
VGGFace2*[3]	R50	-	67.10	80.00	-	-	-	74.70	84.00	-	-
Circle-loss*[30]	R34	-	-	-	-	-	-	86.78	93.44	96.04	-
Circle-loss*[30]	R100	-	-	-	-	-	-	89.60	93.95	**96.29**	-
ArcFace*[5]	R100	-	-	94.20	-	-	-	-	95.60	-	-
MagFace*[21]	R100	**42.32**	**90.36**	**94.51**	-	-	**90.24**	**94.08**	**95.97**	-	-
Triplet-loss	R34	4.42	12.57	32.65	61.33	88.78	4.04	15.32	36.86	66.46	90.77
contrastive-loss	R34	33.10	59.40	72.18	81.98	90.11	57.84	66.41	76.16	85.03	92.21
CosFace[28]	R34	39.70	87.47	93.55	95.71	97.05	85.95	92.57	95.23	96.81	97.94
Cos+UNPG	R34	**43.33**	87.51	93.58	**95.96**	97.24	87.84	92.49	**95.33**	**96.94**	**98.06**
ArcFace	R34	40.61	86.28	93.38	95.74	97.22	85.47	92.21	95.08	96.79	97.94
Arc+Triplet	R34	38.31	86.46	93.22	95.72	97.28	86.40	92.19	94.97	96.68	97.94
Arc+Contrastive	R34	38.07	86.54	93.03	95.61	**97.33**	85.21	92.54	94.86	96.60	98.01
Arc+UNPG	R34	40.27	**88.05**	**93.66**	**95.96**	97.17	**87.99**	**93.02**	**95.33**	96.88	97.92
CosFace	R100	42.27	89.38	94.39	96.17	97.35	86.56	94.42	96.35	**97.57**	98.26
Cos+UNPG	R100	**49.13**	90.61	94.99	96.50	97.36	86.95	94.48	**96.39**	**97.57**	98.24
ArcFace	R100	40.68	89.99	94.89	96.40	97.59	86.57	93.93	96.25	97.43	98.31
Arc+UNPG	R100	42.08	91.76	95.16	96.47	97.62	89.64	94.73	96.37	97.51	**98.32**
MagFace	R100	43.71	89.03	93.99	96.11	97.32	87.19	93.30	95.54	97.00	98.05
Mag+UNPG	R100	46.33	**90.93**	**95.21**	96.50	97.63	**90.01**	**94.70**	96.38	97.51	**98.32**

Table 1: Verification accuracy of TAR@FAR on IJB-B and IJB-C. "*" indicates results from the original paper.

Method	LFW	CFP-FP	AgeDB	CALFW
Circle-loss*	99.73	96.02	-	-
ArcFace*	99.82	-	-	95.45
MagFace*	**99.83**	**98.46**	**98.17**	**96.15**
CosFace	**99.83**	97.72	98.11	96.11
Cos+UNPG	99.81	98.50	98.31	96.15
ArcFace	**99.83**	98.60	98.23	96.11
Arc+UNPG	**99.83**	98.60	98.25	96.13
MagFace	99.81	**98.62**	98.30	96.15
Mag+UNPG	99.81	98.52	**98.38**	**96.21**

Table 2: Verification accuracy on LFW, CFP-FP, AgeDB-30, and CALFW with ResNet-100 backbone.

Results on K-FACE. K-FACE focuses on FR under fine-grained conditions. It consists of 4.3M images with 6 accessories, 30 illuminations, 3 expressions, and 20 poses for 400 persons. We adopted the same training and test splits used in MixFace[13]. The training split was composed of 3.8M images with 370 persons. In particular, the test split, including the remaining 30 persons, was partitioned into Q1, Q2, Q3, and Q4. The number next to Q indicates the variance of conditions where it increases as more conditions are included. Q4 is the most challenging task, whereas Q1 is the most straightforward task among the four. Table 4 presents the results of the FR models. Surprisingly, ArcFace with UNPG outperformed the other FR models on Q1, Q2, Q3, and Q4. Specifically, it obtained gains of 25.38%, 19.34%,

Method	Rank-1 accuracy (%)
AdaCos*[41]	97.41
ArcFace*	98.35
Circle-loss*	98.50
MagFace	98.51
Mag+UNPG	98.03
ArcFace	98.56
Arc+UNPG	98.82
CosFace	99.08
Cos+UNPG	**99.27**

Table 3: Identification results on MegaFace datasets with ResNet-100 backbone except for AdaCos. "*" indicates the results from the original paper.

Method	Q4(TAR@FAR)		Q3(TAR@FAR)		Q2(TAR@FAR)		Q1(TAR@FAR)	
	1e-4	1e-3	1e-4	1e-3	1e-4	1e-3	1e-3	1e-2
ArcFace	0.29	4.04	4.40	18.27	41.29	63.91	94.00	**100**
AdaCos	2.57	16.68	9.94	34.57	26.31	66.88	94.00	**100**
SN-pair[3]	7.21	17.45	21.16	30.85	33.26	55.92	91.80	97.60
MS-loss[5]	8.70	25.01	18.74	38.36	46.64	66.63	94.60	99.20
MixFace[3]	10.92	19.92	22.55	37.67	44.48	67.44	97.00	**100**
Circle-loss[40]	**25.05**	**43.46**	**41.54**	**64.88**	**77.93**	**89.97**	**100**	**100**
Arc+UNPG	**50.43**	**64.05**	**60.88**	**78.68**	**93.26**	**95.68**	**100**	**100**

Table 4: Verification accuracy of TAR@FAR on K-FACE with ResNet-34 backbone.

and 15.33% in TAR@(FAR=1e-4) compared to Circle-loss.

4.2 Analysis

Does it sufficiently satisfy WDFS? We conclude that UNPG helps FR models to form WDFS by reducing the gap between $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$. As shown in Fig. 3, we measured the number of overlapping similarity scores between $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$ using ArcFace, CosFace, and MagFace with and without UNPG after training with R100. We randomly sampled 256 positive pairs and 256 most hard negative pairs at each iteration from MS1M-V2. After 1000 iterations, we generated $\hat{\mathcal{S}}^p$ and $\hat{\mathcal{S}}^n$, each with a total of 257,992, and calculated the

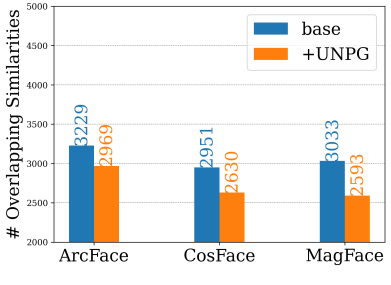


Figure 3: Comparison of overlapping similarities for positive and negative pairs with and without UNPG.

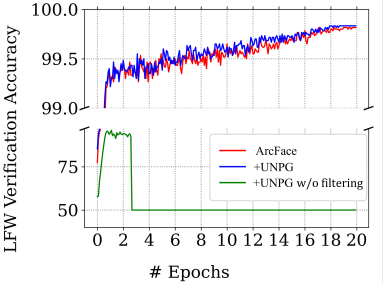


Figure 4: Effects of noise negative pair filtering in UNPG with ResNet-34.

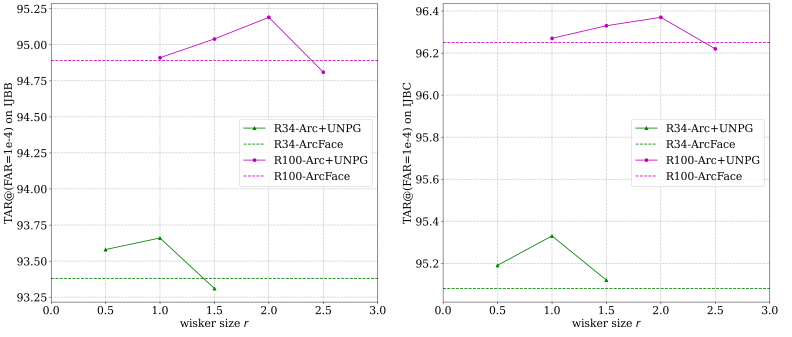


Figure 5: Effects of backbone capacity and whisker size on IJB-B (left) and IJB-C (right).

overlap between them using a histogram. Obviously, applying UNPG reduced the gaps of 260 (ArcFace), 321 (CosFace), and 440 (MagFace) consistently. This proves the effect of UNPG, as expected.

Effect of Noise-negative Pair Filtering. To approximate WDFS, $\mathcal{N}^{ml}$ was assumed to include extremely hard negative pairs because it can produce similarity scores similar to $\sup\mathcal{S}^n$. In Fig. 4, we observed that an FR model using $\mathcal{N}^{ml}$ without filtering (+UNPG w/o filtering) at each iteration leads to performance degradation and the divergence of a loss on LFW, whereas it achieved better performance and convergence of a loss with filtering (+UNPG). Although FR models should adequately distinguish too-hard negative pairs ultimately, we argue that it causes adverse effects using a model lacking representation power to cover them.

We can deal with too-hard pairs by enlarging the model capacity, as depicted in Fig. 5. We conducted experiments using ArcFace with different backbones, R34 and R100, on IJB-B, IJB-C and MegaFace for verification and identification. In R34, the highest performance was obtained at whisker size $r = 1.0$ on all datasets, whereas it was obtained at $r = 2.0$ in R100. This reveals that the informative range determined by whisker size r also increases as a model has a large representation power. More analysis can be found in the supplementary material.

5 Conclusion

This paper is based on two insights. First, from a unified perspective, CL and ML have the same purpose of approaching WDFS, except for PG. Second, CL and ML show a mismatch between two similarity distributions of sampled pairs and all negative pairs. Based on these insights, we developed UNPG by combining two PG strategies (MLPG and CLPG) to alleviate the mismatch. Filtering was also applied to remove negative pairs in both too-easy and too-hard pairs. It was observed that UNPG increases the ability to learn existing FR models compared to MLPG and CLPG by providing more informative pairs. Finally, we suggest two research directions in FR: 1) pair generation strategies in the qualitative aspect and 2) loss functions considering the capability of representation power.

Acknowledgement This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1I1A3052815) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00921).

References

[1] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Elasticface: Elastic margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1587, 2022.

[2] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[3] Yeji Choi, Hyunjung Park, Gi Pyo Nam, Haksub Kim, Heeseung Choi, Junghyun Cho, and Ig-Jae Kim. K-face: A large-scale kist face database in consideration with unconstrained environments. *arXiv preprint arXiv:2103.02211*, 2021.

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546. IEEE, 2005.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[6] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[7] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[11] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[12] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5910, 2020.

[13] Junuk Jung, Sungbin Son, Joochan Park, Yongjun Park, Seonhoon Lee, and Heung-Seon Oh. Mixface: Improving face verification focusing on fine-grained conditions. *arXiv preprint arXiv:2111.01717*, 2021.

[14] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[15] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18750–18759, 2022.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[17] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, page 507–516, 2016.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017.

[19] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics*, pages 158–165. IEEE, 2018.

[20] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14225–14234, 2021.

[21] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 51–59, 2017.

[22] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

[23] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Association*, pages 41.1–41.12, 2015.

[24] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[26] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9. IEEE, 2016.

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

[28] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in Neural Information Processing Systems*, 29, 2016.

[29] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[30] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[32] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[33] John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, MA, 1977.

[34] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1041–1049, 2017.

[35] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[36] 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 *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[37] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[38] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

[39] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017.

[40] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[41] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6490–6499, 2019.

[42] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019.

[43] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.