Feature Transfer Learning for Face Recognition with Under-Represented Data

Xi Yin†∗, Xiang Yu‡, Kihyuk Sohn‡, Xiaoming Liu† and Mannohan Chandraker§‡
†Michigan State University
‡NEC Laboratories America
§University of California, San Diego
{yinxi1,liuxm}@cse.msu.edu, {xiangyu,ksohn,manu}@nec-labs.com

Abstract

Despite the large volume of face recognition datasets, there is a significant portion of subjects, of which the samples are insufficient and thus under-represented. Ignoring such significant portion results in insufficient training data. Training with under-represented data leads to biased classifiers in conventionally-trained deep networks. In this paper, we propose a center-based feature transfer framework to augment the feature space of under-represented subjects from the regular subjects that have sufficiently diverse samples. A Gaussian prior of the variance is assumed across all subjects and the variance from regular ones are transferred to the under-represented ones. This encourages the under-represented distribution to be closer to the regular distribution. Further, an alternating training regimen is proposed to simultaneously achieve less biased classifiers and a more discriminative feature representation. We conduct ablative study to mimic the under-represented datasets by varying the portion of under-represented classes on the MS-Celeb-1M dataset. Advantageous results on LFW, IJB-A and MS-Celeb-1M demonstrate the effectiveness of our feature transfer and training strategy, compared to both general baselines and state-of-the-art methods. Moreover, our feature transfer successfully presents smooth visual interpolation, which conducts disentanglement to preserve identity of a class while augmenting its feature space with non-identity variations such as pose and lighting.

1. Introduction

Face recognition is one of the ongoing success stories in the deep learning era, yielding very high accuracy on several benchmarks [12,20,21]. However, it remains undetermined how deep learning classifiers for fine-grained recognition are trained to maximally exploit real-world data. While it is known that recognition engines are data-hungry and keep improving with more volume, mechanisms to derive benefits from the vast diverse data are relatively unexplored. In particular, as discussed by [18], there is a non-negligible part of data that is under-represented (UR), where only a few samples are available for each class.

It is evident that classifiers that ignore this UR data likely imbibe hidden biases. Consider CASIA-WebFace [47] dataset as an example (Figure 1 (a)). About 39% of the 10K subjects have less than 20 images. A simple solution is to discard the UR classes, which results in insufficient training data. Besides reduction in the volume of data, the inherently uneven sampling leads to bias in the weight norm distribution across regular and UR classes (Figure 1 (b,c)). Sampling
UR classes at a higher frequency alleviates the problem, but still leads to biased decision boundaries due to insufficient intra-class variance in UR classes (Figure 1 (d)).

In this paper, we propose Feature Transfer Learning (FTL) to train less biased face recognition classifiers by adapting the feature distribution of UR classes to mimic that of regular classes. Our FTL handles such UR classes during training by augmenting their feature space using a center-based transfer. In particular, assuming a Gaussian prior on features with class-specific mean and the shared variance across regular and UR classes, we generate new samples of UR classes at feature space, by transferring the linear combination of the principal components of variance that are estimated from regular classes to the UR classes.

Our feature transfer addresses the issue of imbalanced training data. However, using the transferred data directly for training is sub-optimal as the transfer might skew the class distributions. Thus, we propose a training regimen that alternates between carefully designed choices to solve for feature transfer (with the goal of obtaining a less biased decision boundary) and feature learning (with the goal of learning a more discriminative representation) simultaneously. Besides, we propose a novel and effective metric regularization which contributes to the general deep training in an orthogonal way.

To study the empirical properties of our method, we construct UR datasets by limiting the number of samples for various proportions of classes in MS-Celeb-1M [12], and evaluate on LFW [20], IJB-A [21] and the hold-out test set from MS-Celeb-1M. We observe that our FTL consistently improves upon baseline method that does not specifically handle UR classes. Advantageous results over state-of-the-art methods on LFW and IJB-A further confirm the effectiveness of the feature transfer module. Moreover, our FTL can be applied to low-shot or one-shot scenarios, where a few samples are available for some classes. Competitive record on MS-celeb-1M one-shot challenge [11] evidences the advantage. Finally, we visualize our feature transfer module through smooth feature interpolation. It shows that for our feature representation, identity is preserved while non-identity aspects are successfully disentangled and transferred to the target subject.

We summarize our contributions as the following items.

- A center-based feature transfer algorithm to enrich the distribution of UR classes, leading to diversity without sacrificing volume. It also leads to an effective disentanglement of identity and non-identity representations.
- A two-stage alternative training scheme to achieve a less biased classifier and retain discriminative power of the feature representation.
- A simple but effective metric regularization to enhance performance for both our method and baselines, which is also applicable to other recognition tasks.
- Extensive ablation experiments demonstrate the effectiveness of our FTL framework. Combining with the proposed m-L2 regularization and other orthogonal metric learning methods, we achieve top performance on LFW and IJB-A.

2. Related Work

Imbalanced data classification Classic works study data re-sampling methods [1, 15], which learn unbiased classifiers by changing the sampling frequency. By applying deep neural networks [16, 22], the frontier of face recognition research has been significantly advanced [24, 32, 42]. However, there are only few works that discuss about learning from UR data. Huang et al. [19] propose quintuplet sampling based hinge loss to maintain both inter-cluster and inter-class margins. Zhang et al. [50] propose the range loss that simultaneously reduces intra-class variance and enlarges the inter-class variance. However, UR classes are treated in the same way as regular classes in the above methods. Guo and Zhang [11] propose UR class promotion loss that regularizes the norm of weight vectors of UR classes, which can solve the unbalance issue to some extent. Other than designing data sampling rules or regularization on UR classes, we augment UR classes by generating feature-level samples through transfer of intra-class variance from regular classes, which solves the fundamental problem of UR data.

One-shot and low-shot learning Low-shot learning aims at recognizing an image for a specific class with very few or even one image available at training. Some efforts are made by enforcing strong regularization [14] or utilizing non-parametric classification methods based on distance metric learning [34, 39]. Generative model based methods have also been studied in recent years. Dixit et al. [9] propose a data augmentation method using attribute-guided feature descriptor for generation. The method in [14] proposes non-parametric generation of features by transferring within class pair-wise variation from regular classes in object classification task. Compared to their task on ImageNet [30] with 1K classes, face recognition is a fine-grained classification problem that incorporates at least two orders of magnitude more classes with low inter-class variance.

Feature transfer learning Transfer learning applies information from a known domain to an unknown one [3, 4]. We refer to [27] for further discussion. Attributes are used in [9] to synthesize feature-level data. In [35], features are transferred from web images to video frames via a generative adversarial network (GAN) [10]. Our method shares the same flavor in terms of feature transfer concept. However, compared to [35], no additional supervision is provided in our method as it may introduce new bias. We model the intra-class variance in a parametric way, assuming the regular classes and UR classes share the same feature variance distribution. By transferring this shared variance, we transfer sample features from regular classes to UR classes.

5705
3. The Proposed Approach

In this section, we first introduce the problems caused by training with UR classes for face recognition (Sec. 3.1). Then, we present the recognition backbone framework with our proposed metric regularization (Sec. 3.2), our proposed feature transfer framework (Sec. 3.3), and the alternating training scheme to solve these problems (Sec. 3.4).

3.1. Limitations of Training with UR Classes

A recent work [50] shows that directly learning face representation with UR classes results in degraded performance. To demonstrate the problems of training with UR classes, we train a network (CASIA-Net) on CASIA-Webface [47], of which the data distribution is shown in Figure 1 (a). We mainly observe two issues: (1) wildly variant classifier weight norms; and (2) imbalanced intra-class variances between regular and UR classes.

**Imbalance on classifier weight norm** As shown in Figure 1 (b), the norms of the classifier weights (i.e., the weights in the last fully connected layer) of regular classes are much larger than those of UR classes, which causes the decision boundary biases towards the UR classes [11]. This is because the much larger volume of regular classes lead to more frequent weight updates than those of UR classes. To alleviate this problem, there are typical solutions such as data re-sampling or weight normalization [11]. However, such strategies can not solve the fundamental problem of lacking sufficient and diversified samples in UR classes, which is demonstrated in the following.

**Imbalance on intra-class variance** As an illustrative example, we randomly pick two classes, one regular class (ID=1008) and one UR class (ID=10449). We visualize the features from two classes projected onto 2D space using t-SNE [38] in Figure 1(c). Further, the feature space after weight norm regularization is shown in Figure 1(d). Although the weight norms are regularized to be similar, the low intra-class variance of the UR class still causes the decision boundary bias problem.

Based on these observations, we posit that enlarging the intra-class variance for UR classes is the key to alleviate these imbalance issues. Therefore, we propose a feature transfer learning approach that generates extra samples for UR classes to enlarge the intra-class variance. As illustrated in Figure 1(d), the feature distribution augmented by the virtual samples (blue triangles) helps to rectify the classifier decision boundary and learn a better representation.

3.2. The Proposed Framework

Most recent success in deep face recognition works on novel losses or regularizations [7,24,31,32,34], which aim at improving model generalization. In contrast, our method focuses on enlarging intra-class variance of UR classes by transferring knowledge from regular classes. At first glance, our goal of diversifying features seems to contradict with the general premise of face recognition frameworks, i.e., pursuing compact features. In fact, we enlarge the intra-class variance of UR classes at a lower level feature space, which we term as rich-feature layer [13]. The subsequent filtering layers will learn a more discriminative representation.

As illustrated in Figure 2, the proposed framework is composed of several modules including an encoder, decoder, feature transfer module followed by filtering module and a classifier layer. An encoder $E_{nc}$ computes rich features $g = E_{nc}(x) \in \mathbb{R}^{320}$ from an input image $x \in \mathbb{R}^{100 \times 100}$ and reconstructs the input with a decoder $D_{ec}$, i.e., $x' = D_{ec}(g) = D_{ec}(E_{nc}(x)) \in \mathbb{R}^{100 \times 100}$. This pathway is trained with the following pixel-wise reconstruction loss:

$$L_{recon} = \|x' - x\|_2^2. \quad (1)$$

The reconstruction loss allows $g$ to contain diverse non-identity variations such as pose, expression, and lighting. Therefore, we denote $g$ as the rich feature space.

A filtering network $R$ is applied to generate discriminative identity features $f = R(g) \in \mathbb{R}^{320}$ that are fed to a linear classifier layer $FC$ with weight matrix $W = \{w_j\}_{j=1}^{N_f} \in \mathbb{R}^{320 \times 5706}$.
where $N_c$ is the total number of classes. This pathway optimizes the softmax loss:

$$L_{sfmx} = - \log \frac{\exp(w_i^T f)}{\sum_{j=1}^{N_c} \exp(w_j^T f)},$$

where $y_i$ is the ground-truth identity label of $x$.

Note that softmax loss is scale-dependent where the loss can be made arbitrarily small by scaling the norm of the weights $w_j$ or features $f$. Typical solutions to prevent this problem are to either regularize the norm of weights or features, or to normalize both of them. However, we argue that these methods are too stringent since they penalize norms individually.

Finally, we formulate the training loss in Eqn. (4), with the following coefficients $\alpha_{sfmx} = \alpha_{recon} = 1$, $\alpha_{reg} = 0.25$ unless otherwise stated:

$$L = \alpha_{sfmx} L_{sfmx} + \alpha_{recon} L_{recon} + \alpha_{reg} L_{reg}. \tag{4}$$

### 3.3. Feature Transfer for UR Classes

Following the Joint Bayesian face model [2], we assume that the rich feature $g_{ik}$ from class $i$ lies in a Gaussian distribution with a class mean $c_i$ and a covariance matrix $\Sigma_i$. The class mean or center is estimated as an arithmetic average over all features from the same class. As shown in the left of Figure 3, the center representation of regular classes is identity-specific while removing non-identity factors such as pose, expression and illumination. However, as in the right of Figure 3, due to the lack of samples, the center estimation of UR classes is not accurate and often biased towards certain identity-irrelevant factors like pose, which we find to be dominant in practice.

To improve the quality of center estimation for UR classes, we discard samples with extreme pose variation. Furthermore, we consider averaging features from both the original and horizontally flipped images. With $g_{ik} \in \mathbb{R}^{320}$ denoting the rich feature extracted from the flipped image, the feature center is estimated as follows:

$$c_i = \frac{1}{2|\Omega_i|} \sum_{k \in \Omega_i} (g_{ik} + \bar{g}_{ik}), \quad \Omega_i = \{ k \mid |p_{ik}| + |\bar{p}_{ik}| \leq \tau \},$$

where $p_{ik}$ and $\bar{p}_{ik}$ are the estimated poses of the original and flipped images, respectively. By bounding the summation, we expect the yaw angle $p_{ik}$ to be an inlier.

To transfer the intra-class variance from regular classes to UR classes, we assume the covariance matrices are shared across all classes, i.e., $\Sigma_i = \Sigma$. In theory, one can draw feature samples of UR classes by adding a noise vector $\epsilon \sim \mathcal{N}(0, \Sigma)$ to its center $c_i$. However, the direction of the noise vector might be too random and does not reflect the true factors of variations found in the regular classes. Therefore, we transfer the intra-class variance evaluated from the samples of regular classes. First, we calculate the covariance matrix $V$ via:

$$V = \sum_{i=1}^{N_c} \sum_{k=1}^{m_i} (g_{ik} - c_i)\Sigma (g_{ik} - c_i)^T,$$

where $m_i$ is the total number of samples for class $i$. We perform PCA to decompose $V$ into major components and take the first 150 Eigenvectors as $Q \in \mathbb{R}^{320 \times 150}$, which preserves 95% energy. Our center-based feature transfer is achieved via:

$$\bar{g}_{ik} = c_i + Q Q^T (g_{ik} - c_i),$$

where $g_{jk}$ and $c_j$ are the feature-level sample and the center of a regular class $j$. $c_i$ is the feature center of an UR class $i$ and $\bar{g}_{ik}$ is the transferred features for class $i$. Here, $\bar{g}_{ik}$ preserves the same identity as $c_i$, with similar intra-class variance as $g_{ik}$. By sufficiently sampling $g_{jk}$ across different regular classes, we expect to obtain an enriched distribution of the UR class $i$, which consists of both the original features $g_{ik}$ and the transferred features $\bar{g}_{ik}$.
Stage 1: Decision boundary reshape.
Fixed models: Enc and Dec.
Training models: R and FC, using Eqn. 2 and 3.
Init \([C, Q, h] = \text{UpdateStats}()\), \(N_{iter} = \#\) iterations.
for \(i = 1, \ldots, N_{iter}\) do
  Train 1st batch sampled from \(h\) in \(D_{\text{reg}}\): \([x', y']\).
  Train 2nd batch sampled from \(D_{\text{UR}}\): \([x^u, y^u]\).
  Feature transfer: \(\tilde{g}^u = \text{Transfer}(x', y', y^u)\).
Train 3rd batch: \([g^u, y^u]\).
Stage 2: Compact feature learning.
Fixed models: FC.
for \(i = 1, \ldots, N_{iter}\) do
  train batch sampled from \(D\): \([x, y]\).
Alternate stage 1 and 2 every \(N_{iter}\) until convergence.

Function \([C, Q, h] = \text{UpdateStats}()\)
Init \(C = [], V = [], h = [], m_i = \#\) samples in class \(i\), \(N_c = \#\) classes, \(N_s = \#\) samples in each batch.
for \(i = 1, \ldots, N_c\) do
  for \(j = 1, \ldots, m_i\) do
    \(g_{ij} = \text{Enc}(x_{ij}), g_{ij} = \text{Enc}(x_{ij})\)
    \(c_i = \frac{1}{m_i} \sum_{b \in D_i} (g_{ik} + g_{ik})\)
    C.append(c_i)
  if \(i\) in \(D_{\text{reg}}\) then
    \(d_i = \frac{1}{m_i} \sum_k ||g_{ik} - c_i||_2\)
    for \(j = 1, \ldots, m_i\) do
      \(V += (g_{ij} - c_j)(g_{ij} - c_i)\)
      if \(||g_{ij} - c_j||_2 > d_i\) then
        \(h.append([ij])\)
  \(Q = \text{PCA}(V)\)
Function \(\tilde{g}^u = \text{Transfer}(x', y', y^u)\)
\(g^r = \text{Enc}(x')\)
for \(k = 1, \ldots, N_s\) do
  \(c_j = C(y_k^u), c_i = C(y_u^i)\)
  \(\tilde{g}^u_k = c_i + QQ^T(g^r_k - c_i)\)

Algorithm 1: Two-stage alternating training strategy.

3.4. Alternating Training Strategy
Given a training set of both regular and UR classes \(D = \{D_{\text{reg}}, D_{\text{UR}}\}\), we first pre-train all modules \(M = \{\text{Enc}, \text{Dec}, R, FC\}\) using Eqn. 4 without feature transfer. Then, we alternate between the two stages of the classifier with our proposed feature transfer method for decision boundary reshape and learning a more discriminative feature representation with boundary-corrected classifier. The overview of our two-stage alternating training process is illustrated in Algorithm 1, which we describe in more details below.

Stage 1: Decision boundary reshape. In this stage, we train \(R\) and FC while fixing other modules (the rich feature space is fixed for stable feature transfer). The goal is to reshape the decision boundary by transferring features from regular classes to UR classes. We first update the statistics for each regular class including the feature centers, PCA basis, and an index list \(h\) of hard samples whose distances to the feature centers exceed the average distance. The PCA basis \(Q\) is achieved by decomposing the covariance matrix \(V\) computed with the samples from all regular classes \(D_{\text{reg}}\). Three batches are applied for training in each iteration: (1) a regular batch sampled from hard index list \(h\): \([g^u, y^u]\), to guarantee no degradation in the performance; (2) a UR batch sampled from UR classes \([g^u, y^u]\), to conduct the updating similar to class-balanced sampling; (3) a transferred batch \([g^u, y^u]\) by transferring the variances from regular batch to UR batch, to reshape the decision boundary.

Stage 2: Compact feature learning. In this stage, we train \(Enc, Dec\) and \(R\) using normal batches \([x, y]\) from both regular and UR classes without feature transfer. We keep FC fixed since it is already updated from the previous stage with decision boundary correction. The gradient directly back-propagates to \(R\) and \(Enc\) to learn a more compact representation that reduces the violation of crossing rectified classifier boundaries. We perform online alternation between stage 1 and 2 for every \(N_{iter}\) iterations until convergence.

4. Experiments
We use MS-Celeb-1M as our training set. Due to label noise, we adopt a cleaned version from [43] and remove the classes overlapped with LFW and IJB-A, which results in 4.8M images of 76.5K classes. A class with no more than 20 images is considered as a UR class, following [50]. A facial keypoint localization method [49] is applied as the face alignment and cropping.

We apply an encoder-decoder structure for model \(Enc\) and \(Dec\). Model \(R\) consists of a linear layer, two de-convolution layers, two convolution layers and another linear layer to obtain \(f \in \mathbb{R}^{320}\). Detail of the network structure is referred to the supplementary material. Adam solver with a learning rate of \(2e^{-4}\) is used in model pre-training. A learning rate of \(1e^{-5}\) is used in stage 1 and 2, which alternate for every 5K iterations until convergence. The hyper-parameter setting is determined by an off-line parameter search based on a hold-out validation set.

4.1. Feature Center Estimation
Feature center estimation is a key step for feature transfer. To evaluate center estimation for UR classes, 1K regular classes are selected from MS-Celeb-1M and features are extracted using a pre-trained recognition model. We randomly choose a subset of 1, 5, 10, 20 images to mimic an UR class. Three methods are compared: (1) “PickOne”, randomly pick one sample as center. (2) “AvgAll”, average features of all images. (3) “AvgFlip”, proposed method in Eqn. 5. We compute the error as the difference between the center of
Figure 4. (a) Center estimation error comparison. (b) Illustration of intra- and inter-class variances. Circles from small to large show the minimum, mean and maximum distances from intra-class samples to center. Distances are averaged across 1K classes.

Figure 5. Toy example on MNIST to show the effectiveness of our m-$L_2$ regularization. Figure shows the feature distributions for models trained without (a) and with (b) m-$L_2$ regularization.

4.2. Effects of m-$L_2$ Regularization

To study the effects of the proposed m-$L_2$ regularization, we show a toy example on the MNIST dataset [23]. We use LeNet++ network (following [42]) to learn a 2D feature space for better visualization. Two models are compared: one trained with softmax loss only; the other trained with softmax loss and m-$L_2$ regularization ($\alpha_{reg} = 0.001$).

We have the following observations: (1) m-$L_2$ effectively avoids over-fitting. In Figure 5, the norm of the features in (a) is much larger than that in (b), as increasing the feature norm can reduce softmax loss, which may cause over-fitting. (2) m-$L_2$ enforces a more balanced feature distribution, where Figure 5 (b) shows a more balanced angular distribution than that in (a). On the MNIST testing set, the performance with m-$L_2$ improves $sfmx$ from 99.06% to 99.35%. Moreover, the testing accuracy with m-$L_2$ improves $sfmx$ and $sfmx + L_2$ from 98.60% and 98.53% to 99.37% on LFW as in Table 3. Note that m-$L_2$ is a general regularization which is orthogonal to our main claim in this paper, that can be easily adapted to other recognition frameworks.

4.3. Ablation Study

We study the impact of the ratio between the portion of regular classes and the portion of UR classes on training a face recognition system. To construct the exact regular and UR classes, we use the top 60K regular classes, which contain the most images from MS-Celeb-1M. Further, the top 10K classes are selected as regular classes which are shared among all training sets. We regard the 10K and 60K sets as the lower and upper bounds. Among the rest 50K classes sorted by the number of images, we select the first 10K, 30K and 50K and randomly pick 5 images per class. In this way, we form the training set of 10K10K, 10K30K, and 10K50K, of which the first 10K are regular and the last 10K or 30K or 50K are called faked UR classes. A hold-out testing set is formed by selecting 5 images from each of the shared 10K regular classes and 10K UR classes.

The evaluation on the hold out test set from MS-Celeb-1M is to mimic low-shot learning, where we use the feature center from the training images as the gallery and nearest neighbor (NN) for face matching. The rank-1 accuracy for both regular and UR classes are reported. We also evaluate the recognition performance on LFW and IJB-A. The results are shown in Table 1 and we draw the following observations.

- The rich feature space $g$ is less discriminative than the feature space $f$, which validates our intuition that $g$ is rich in intra-class variance for feature transfer while $f$ is more discriminative for face recognition.
- The proposed m-$L_2$ regularization boosts the performance with a large margin over the baseline softmax loss.
- The proposed FTL method consistently improves over softmax and sfmx+m-$L_2$ with significant margins.
- Our method is more beneficial when more UR classes are used for training as more training data usually lead to better face recognition performance.

4.4. One-Shot Face Recognition

As our method has tangential relation to low-shot learning, we evaluate on the MS-celeb-1M one-shot challenge [11]. The training data consists of a base set with 20K classes each with 50~100 images and a novel set of 1K classes each with only 1 image. The test set consists of 1 image per base (regular) class and 5 images per novel (UR) class. The goal is to evaluate the performance on the novel
<table>
<thead>
<tr>
<th>Test →</th>
<th>LFW</th>
<th>IJB-A: Verif.</th>
<th>IJB-A: Identif.</th>
<th>MS1M: NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train↓</td>
<td>Method↓</td>
<td>g</td>
<td>f</td>
<td>FAR@.01</td>
</tr>
<tr>
<td>10K0K</td>
<td>sfmx</td>
<td>97.15</td>
<td>97.45</td>
<td>69.39</td>
</tr>
<tr>
<td></td>
<td>sfmx+m-L₂</td>
<td>97.00</td>
<td>97.88</td>
<td>73.00</td>
</tr>
<tr>
<td>10K10K</td>
<td>sfmx</td>
<td>97.08</td>
<td>97.85</td>
<td>72.96</td>
</tr>
<tr>
<td></td>
<td>sfmx+m-L₂</td>
<td>97.13</td>
<td>98.08</td>
<td>74.07</td>
</tr>
<tr>
<td></td>
<td>FTL (Ours)²</td>
<td>96.72</td>
<td>98.33</td>
<td>80.25</td>
</tr>
<tr>
<td>10K30K</td>
<td>sfmx</td>
<td>97.13</td>
<td>98.08</td>
<td>74.03</td>
</tr>
<tr>
<td></td>
<td>sfmx+m-L₂</td>
<td>97.13</td>
<td>98.08</td>
<td>76.92</td>
</tr>
<tr>
<td></td>
<td>FTL (Ours)²</td>
<td>96.87</td>
<td>98.42</td>
<td>81.80</td>
</tr>
<tr>
<td>10K50K</td>
<td>sfmx</td>
<td>97.02</td>
<td>97.93</td>
<td>72.87</td>
</tr>
<tr>
<td></td>
<td>sfmx+m-L₂</td>
<td>97.32</td>
<td>98.10</td>
<td>78.52</td>
</tr>
<tr>
<td></td>
<td>FTL (Ours)²</td>
<td>96.95</td>
<td>98.48</td>
<td>82.60</td>
</tr>
<tr>
<td>60K0K</td>
<td>sfmx</td>
<td>97.52</td>
<td>98.30</td>
<td>82.75</td>
</tr>
<tr>
<td></td>
<td>sfmx+m-L₂</td>
<td>97.90</td>
<td>98.85</td>
<td>86.38</td>
</tr>
</tbody>
</table>

Table 1. Controlled experiments by varying the ratio between regular and UR classes in training sets. FTL (Ours)²: model trained on subsets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ext</th>
<th>#Models</th>
<th>Base</th>
<th>Novel</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCSM [45]</td>
<td>YES</td>
<td>3</td>
<td>–</td>
<td>61.0</td>
</tr>
<tr>
<td>Cheng et al. [5]</td>
<td>YES</td>
<td>4</td>
<td>99.74</td>
<td>100</td>
</tr>
<tr>
<td>Choe et al. [6]</td>
<td>NO</td>
<td>1</td>
<td>≥ 95.00</td>
<td>11.17</td>
</tr>
<tr>
<td>UP [11]</td>
<td>NO</td>
<td>1</td>
<td>99.80</td>
<td>77.48</td>
</tr>
<tr>
<td>Hybrid [44]</td>
<td>NO</td>
<td>2</td>
<td>99.58</td>
<td><strong>92.64</strong></td>
</tr>
<tr>
<td>DM [33]</td>
<td>NO</td>
<td>1</td>
<td>–</td>
<td>73.86</td>
</tr>
<tr>
<td>FTL (Ours)</td>
<td>NO</td>
<td>1</td>
<td>99.21</td>
<td><strong>92.60</strong></td>
</tr>
</tbody>
</table>

Table 2. Comparison on one-shot learning challenge. Result on base classes are reported as rank-1 accuracy and on novel classes as Coverage@Precision = 0.99. “Ext” means “External Data”.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>Method</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG Face [28]</td>
<td>98.95</td>
<td>FaceNet [32]</td>
<td><strong>99.63</strong></td>
</tr>
<tr>
<td>DeepID2 [36]</td>
<td>99.15</td>
<td>CosFace [41]</td>
<td><strong>99.73</strong></td>
</tr>
<tr>
<td>NormFace [40]</td>
<td>99.19</td>
<td>sfmx</td>
<td>98.60</td>
</tr>
<tr>
<td>CenterLoss [42]</td>
<td>99.28</td>
<td>sfmx + L₂</td>
<td>98.53</td>
</tr>
<tr>
<td>SphereFace [24]</td>
<td>99.42</td>
<td>sfmx + m-L₂ (Ours)</td>
<td><strong>99.18</strong></td>
</tr>
<tr>
<td>RangeLoss [50]</td>
<td>99.53</td>
<td>FTL (Ours)</td>
<td><strong>99.55</strong></td>
</tr>
</tbody>
</table>

Table 3. Performance comparisons on LFW. Methods of sfmx, sfmx+L₂, sfmx+m-L₂ are our implementations.

We use the output from softmax layer as the confidence score and achieve 92.60% coverage at precision of 0.99 with single-model single-crop testing, as in Table 2. Note that both methods [5, 44] use model ensemble and multi-crop testing. Compared to methods [6, 11] with similar setting, we achieve competitive performance on the base classes and much better accuracy on the novel classes by 15%.

4.5. Large-Scale Face Recognition

In this section, we train our model on the full MS-celeb-1M dataset and evaluate on LFW and IJB-A. On LFW (Table 3), our performance is strongly competitive, achieving 99.55% whereas the state-of-the-arts show 99.63% from FaceNet [32] and 99.73% from CosFace [41]. On IJB-A (Table 4), the softmax loss with our proposed m-L₂ regularization already provides good results denoted as sfmx+m-L₂. Our FTL improves the performance significantly, with margins varying from 0.6% to 2.8%. We further combine media pooling (MP) and template adaptation (TA) [7] metric learning with our proposed method (FTL + MP + TA), and achieve consistently better results than state-of-the-art methods [46].

4.6. Qualitative Results

We apply decoder Dec in our framework for feature visualization. While skip link between encoder and decoder improves the visual quality [48], we do not apply it to encourage the rich features g to encode intra-class variance. **Center visualization** We compute a feature center for a given class, on which the Dec is applied to generate a center face. As shown in Figure 6, we confirm the observation that the center is mostly an identity-preserved frontal neutral face. It also applies to portrait and cartoon figures.

**Feature transfer** The transferred features are visualized by
Figure 7. Feature transfer visualization between two classes for every two columns. The first row are the input, in which odd column denotes class 1: $x_1$, and the even column denotes class 2: $x_2$. The second row are the reconstructed images $x'_1$ and $x'_2$. In the third row, odd column image is the decoded image of the transferred feature from class 1 to class 2 and even column image is the decoded image of the transferred feature from class 2 to class 1. It is clear that the transferred features share the same identity as the target class while obtain the source image’s non-identity variance including pose, expression, illumination, and etc.

$\alpha \rightarrow 0$. 10. 20. 30. 40. 50. 60. 70. 80. 90. 0

Figure 8. Transition from top-left image to top-right image via feature interpolation. First row shows traditional feature interpolation; second row shows our transition of non-identity variance; third row shows our transition of identity variance.

**Dec.** Let $x_{1,2}, x'_{1,2}, g_{1,2}, c_{1,2}$ denote the input images, reconstructed images, encoded rich features and feature centers of two classes, respectively. We transfer feature from class 1 to class 2 by: $g_{12} = c_2 + QQ^T(g_1 - c_1)$, and visualize the decoded images. We also transfer from class 2 to class 1 and visualize the decoded images. As shown in Figure 7, the transferred images preserve the target class’s identity while retaining intra-class variance of the source image in terms of pose, expression and lighting, which shows that our feature transfer is effective in enlarging the intra-class variance.

**Feature interpolation** The interpolation between two representations shows the appearance transition from one to the other [29, 37]. Let $g_{1,2}, c_{1,2}$ denote the encoded features and the centers of two classes. Previous work generates a new representation as $g = g_1 + \alpha(g_2 - g_1)$ where identity and non-identity changes are mixed together. In our work, we can generate transitions of non-identity change as $g = c_1 + \alpha QQ^T(g_2 - c_2)$ and identity change as $g = g_1 + \alpha(c_2 - c_1)$. Figure 8 shows an interpolation example of a female with left pose and a male with right pose, where the illumination changes significantly. Compared to traditional interpolation that generates undesirable artifacts, our method shows smooth transitions, which verifies that the proposed model is effective at disentangling identity and non-identity features.

5. Conclusions

In this paper, we propose a novel feature transfer approach for deep face recognition training which explores the imbalance issue with $UR$ classes. We observe that generic face recognition approaches encounter classifier bias due to imbalanced distribution of training data across classes. By applying the proposed feature transfer approach, we enrich the feature space of the $UR$ classes, while retaining identity. Utilizing the generated data, our alternating feature learning method rectifies the classifier and learns more compact feature representations. Our proposed $m$-$L_2$ regularization demonstrates consistent advantages which can potentially boost performance across different recognition tasks. The disentangled nature of the augmented feature space is visualized through smooth interpolations. Experiments consistently show that our method can learn better representations to improve the performance on regular, $UR$, and unseen classes. While this paper focuses on face recognition, our future work will also derive advantages from the proposed feature transfer for other recognition applications, such as $UR$ natural species [17].
References


