Feature Transfer Learning for Face Recognition with Under-Represented Data

Xi Yin[†,*], Xiang Yu[‡], Kihyuk Sohn[‡], Xiaoming Liu[†] and Manmohan 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.

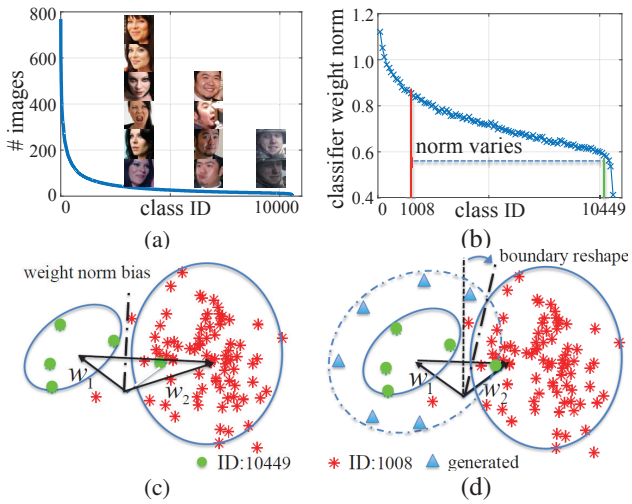


Figure 1. Illustration of the *UR* data problem and our proposed solution. (a) The data distribution of CASIA-WebFace dataset [47]. (b) Classifier weight norm varies across classes in proportion to their volume. (c) Weight norm for regular class 1008 is larger than *UR* class 10449, causing a bias in the decision boundary (dashed line) towards ID 10449. (d) Data re-sampling solves the classifier bias to some extent. However, the variance of ID 1008 is much larger than ID 10449. We augment the feature space of ID 1008 (dashed ellipsoid) and propose improved training strategies, which corrects the classifier bias and learns a better feature representation.

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

[*]Main part of the work is done when Xi was an intern at NEC Laboratories America.

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.

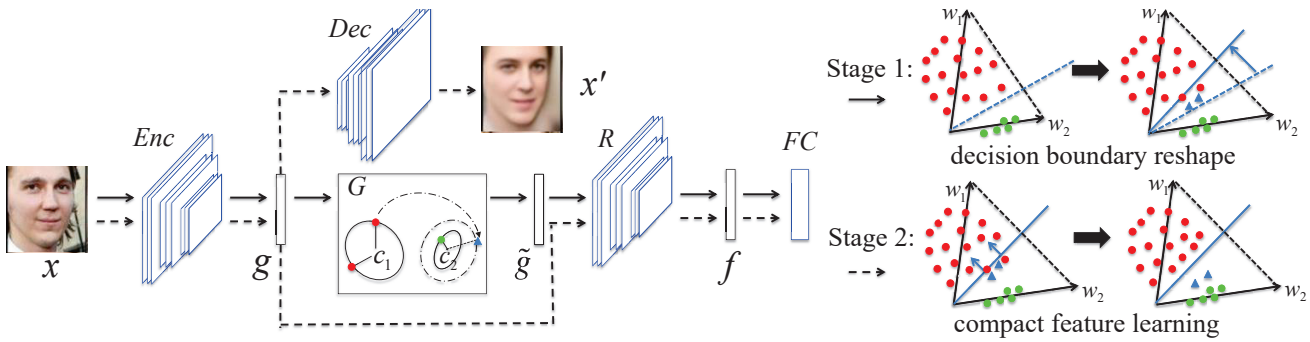


Figure 2. Overview of our proposed FTL framework. It consists of a feature extractor Enc, a decoder Dec, a feature filter R, a fully connected layer as classifier FC, and a feature transfer module G. The network is trained with an alternating bi-stage strategy. At stage 1 (solid arrows), we fix Enc and apply feature transfer G to generate new feature samples (blue triangles) that are more diverse to reshape the decision boundary. In stage 2 (dashed arrows), we fix the rectified classifier FC, and update all the other models. As a result, the samples that are originally on or across the boundary are pushed towards their center (blue arrows in bottom right). Best viewed in color.

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 Enc computes rich features $\mathbf{g} = Enc(\mathbf{x}) \in \mathbb{R}^{320}$ from an input image $\mathbf{x} \in \mathbb{R}^{100\times100}$ and reconstructs the input with a decoder Dec, *i.e.*, $\mathbf{x}' = Dec(\mathbf{g}) = Dec(Enc(\mathbf{x})) \in \mathbb{R}^{100\times100}$. This pathway is trained with the following pixel-wise reconstruction loss:

$$\mathcal{L}_{recon} = \|\mathbf{x}' - \mathbf{x}\|_2^2. \tag{1}$$

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

A filtering network R is applied to generate discriminative identity features $\mathbf{f} = R(\mathbf{g}) \in \mathbb{R}^{320}$ that are fed to a linear classifier layer FC with weight matrix $\mathbf{W} = [\mathbf{w_j}]_{j=1}^{N_c} \in$

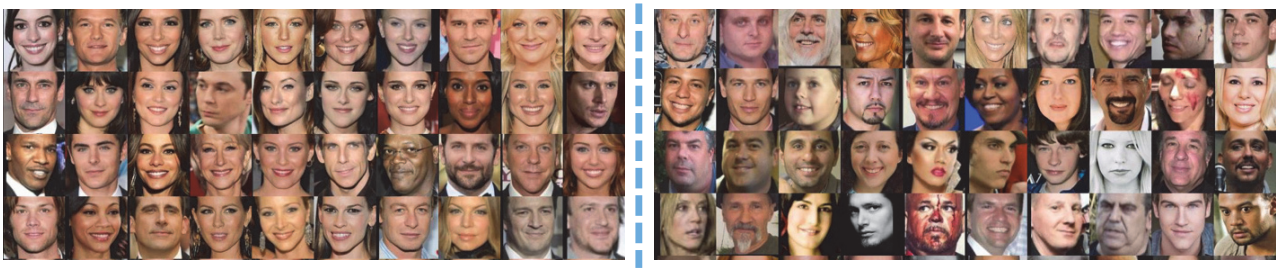


Figure 3. Visualization of samples closest to the feature center of classes with most number of images (left) and classes with least number of images (right). We find that near-frontal close-to-neutral faces are the nearest neighbors of the feature centers of regular classes. However, the nearest neighbors of the feature centers of *UR* classes still contain pose and expression variations. Features are extracted by VGGFace model [28] and samples are from CASIA-WebFace dataset.

$\mathbb{R}^{N_c \times 320}$ where N_c is the total number of classes. This pathway optimizes the softmax loss:

$$\mathcal{L}_{sfmx} = -\log \frac{\exp(\mathbf{w}_{y_i}^T \mathbf{f})}{\sum_j^{N_c} \exp(\mathbf{w}_j^T \mathbf{f})}, \tag{2}$$

where y_i is the ground-truth identity label of $\mathbf{x}$.

Note that softmax loss is *scale-dependent* where the loss can be made arbitrarily small by scaling the norm of the weights $\mathbf{w}_j$ or features $\mathbf{f}$. Typical solutions to prevent this problem are to either regularize the norm of weights[1] or features, or to normalize both of them [40]. However, we argue that these methods are too stringent since they penalize norms of individual weights and features without considering their compatibility. Instead, we propose to regularize the norm of the output of FC as following:

$$\mathcal{L}_{reg} = \|\mathbf{W}^T \mathbf{f}\|_2^2. \tag{3}$$

We term the proposed regularization as metric L_2 or m-L_2 regularization. As will shown in the experiment, joint regularization on weights and features works better than individual regularization.

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:

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

3.3. Feature Transfer for *UR* Classes

Following the Joint Bayesian face model [2], we assume that the rich feature $\mathbf{g}_{ik}$ from class i lies in a Gaussian distribution with a class mean $\mathbf{c}_i$ and a covariance matrix Σ_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.

[1] http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression#Weight_Decay

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 $\bar{\mathbf{g}}_{ik} \in \mathbb{R}^{320}$ denoting the rich feature extracted from the flipped image, the feature center is estimated as follows:

$$\mathbf{c}_i = \frac{1}{2|\Omega_i|} \sum_{k \in \Omega_i} (\mathbf{g}_{ik} + \bar{\mathbf{g}}_{ik}),\ \Omega_i = \{k \mid |p_{ik}| + |\bar{p}_{ik}| \leq \tau\}, \tag{5}$$

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., $\mathbf{\Sigma}_i = \mathbf{\Sigma}$. In theory, one can draw feature samples of *UR* classes by adding a noise vector $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ to its center $\mathbf{c}_i$. 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 $\mathbf{V}$ via:

$$\mathbf{V} = \sum_{i=1}^{N_c} \sum_{k=1}^{m_i} (\mathbf{g}_{ik} - \mathbf{c}_i)^T (\mathbf{g}_{ik} - \mathbf{c}_i) \tag{6}$$

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

$$\tilde{\mathbf{g}}_{ik} = \mathbf{c}_i + \mathbf{Q}\mathbf{Q}^T (\mathbf{g}_{jk} - \mathbf{c}_j), \tag{7}$$

where $\mathbf{g}_{jk}$ and $\mathbf{c}_j$ are the feature-level sample and the center of a regular class j. $\mathbf{c}_i$ is the feature center of an *UR* class i and $\tilde{\mathbf{g}}_{ik}$ is the transferred features for class i. Here, $\tilde{\mathbf{g}}_{ik}$ preserves the same identity as $\mathbf{c}_i$, with similar intra-class variance as $\mathbf{g}_{jk}$. By sufficiently sampling $\mathbf{g}_{jk}$ across different regular classes, we expect to obtain an enriched distribution of the *UR* class i, which consists of both the original features $\mathbf{g}_{ik}$ and the transferred features $\tilde{\mathbf{g}}_{ik}$.

Stage 1: Decision boundary reshape.
Fixed models: Enc and Dec.
Training models: R and FC, using Eqn. 2 and 3.
Init $[\mathbf{C}, \mathbf{Q}, \mathbf{h}]$ = ***UpdateStats***(), N_{iter} = # iterations.
for $i = 1, \ldots, N_{iter}$ **do**
  Train 1st batch sampled from $\mathbf{h}$ in $\mathbb{D}_{reg}$: $\{\mathbf{x}^r, \mathbf{y}^r\}$.
  Train 2nd batch sampled from $\mathbb{D}_{UR}$: $\{\mathbf{x}^u, \mathbf{y}^u\}$.
  Feature transfer: $\tilde{\mathbf{g}}^u$ = ***Transfer***$(\mathbf{x}^r, \mathbf{y}^r, \mathbf{y}^u)$.
  Train 3rd batch: $\{\tilde{\mathbf{g}}^u, \mathbf{y}^u\}$.
Stage 2: Compact feature learning.
Fixed models: FC.
Training models: Enc, Dec, and R, using Eqn. 4.
for $i = 1, \ldots, N_{iter}$ **do**
  train batch sampled from $\mathbb{D}$: $\{\mathbf{x}, \mathbf{y}\}$.
Alternate stage 1 and 2 every N_{iter} until convergence.

Function $[\mathbf{C}, \mathbf{Q}, \mathbf{h}]$ = ***UpdateStats***()
Init $\mathbf{C} = []$, $\mathbf{V} = []$, $\mathbf{h} = []$, 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**
    $\mathbf{g}_{ij} = Enc(\mathbf{x}_{ij})$, $\bar{\mathbf{g}}_{ij} = Enc(\bar{\mathbf{x}}_{ij})$
  $\mathbf{c}_i = \frac{1}{2|\Omega_i|} \sum_{k \in \Omega_i} (\mathbf{g}_{ik} + \bar{\mathbf{g}}_{ik})$
  $\mathbf{C}$.append($\mathbf{c}_i$)
  if i *in* $\mathbb{D}_{reg}$ **then**
    $d_i = \frac{1}{m_i} \sum_k ||\mathbf{g}_{ik} - \mathbf{c}_i||_2$
    for $j = 1, \ldots, m_i$ **do**
      $\mathbf{V}$ += $(\mathbf{g}_{ij} - \mathbf{c}_i)^T (\mathbf{g}_{ij} - \mathbf{c}_i)$
      if $||\mathbf{g}_{ij} - \mathbf{c}_i||_2 > d_i$ **then**
        $\mathbf{h}$.append([i,j])
$\mathbf{Q} = \text{PCA}(\mathbf{V})$
Function $\tilde{\mathbf{g}}^u$ = ***Transfer***$(\mathbf{x}^r, \mathbf{y}^r, \mathbf{y}^u)$
$\mathbf{g}^r = Enc(\mathbf{x}^r)$
for $k = 1, \ldots, N_s$ **do**
  $\mathbf{c}_j = \mathbf{C}(\mathbf{y}^r_k, :)$, $\mathbf{c}_i = \mathbf{C}(\mathbf{y}^u_k, :)$
  $\tilde{\mathbf{g}}^u_k = \mathbf{c}_i + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}^r_k - \mathbf{c}_j)$

Algorithm 1: Two-stage alternating training strategy.

3.4. Alternating Training Strategy

Given a training set of both regular and *UR* classes $\mathbb{D} = \{\mathbb{D}_{reg}, \mathbb{D}_{UR}\}$, we first pre-train all modules $\mathbb{M} = \{Enc, Dec, R, FC\}$ using Eqn. 4 without feature transfer. Then, we alternate between the training 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 $\mathbf{C}$, PCA basis $\mathbf{Q}$ and an index list $\mathbf{h}$ of hard samples whose distances to the feature centers exceeding the average distance. The PCA basis $\mathbf{Q}$ is achieved by decomposing the covariance matrix $\mathbf{V}$ computed with the samples from all regular classes $\mathbb{D}_{reg}$. Three batches are applied for training in each iteration: (1) a regular batch sampled from hard index list $\mathbf{h}$: $\{\mathbf{g}^r, \mathbf{y}^r\}$, to guarantee no degradation in the performance; (2) a *UR* batch sampled from *UR* classes $\{\mathbf{g}^u, \mathbf{y}^u\}$, to conduct the updating similar to class-balanced sampling; (3) a transferred batch $\{\tilde{\mathbf{g}}^u, \mathbf{y}^u\}$ 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 $\{\mathbf{x}, \mathbf{y}\}$ 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.

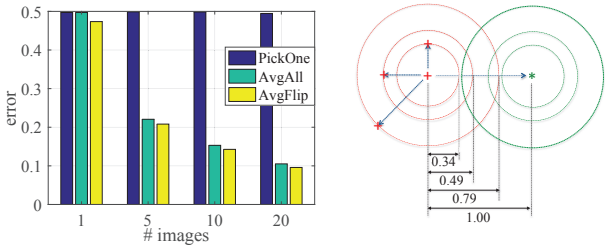
4. Experiments

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 key point 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 deconvolution layers, two convolution layers and another linear layer to obtain $\mathbf{f} \in \mathbb{R}^{320}$. 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.

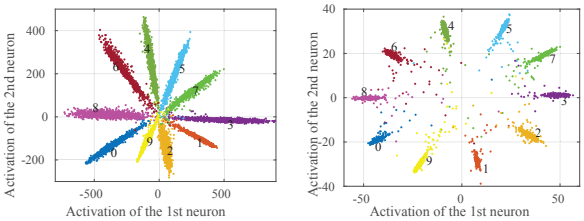
4.1. Feature Center Estimation

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



(a) (b)

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.



(a) (b)

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.

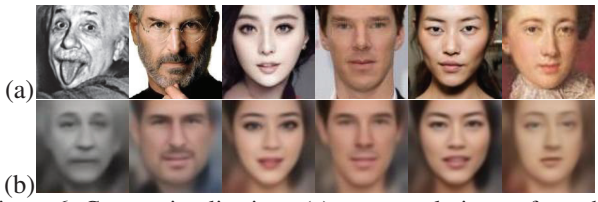
the full set (ground truth) and the subset (estimated), and is normalized by the inter-class variance.

Results in Figure 4 show that our "AvgFlip" achieves a smaller error. When compared to the intra-class variance, the error is fairly small, which suggests that our center estimation is accurate to support the feature transfer.

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.



(a) (b)

Figure 6. Center visualization. (a) one sample image from the selected class; (b) the decoded image from the feature center.

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 $\mathbf{g}$ is less discriminative than the feature space $\mathbf{f}$, which validates our intuition that $\mathbf{g}$ is rich in intra-class variance for feature transfer while $\mathbf{f}$ 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

Test →		LFW		IJB-A: Verif.		IJB-A: Identif.		MS1M: NN	
Train↓	Method↓	**g**	**f**	FAR@.01	@.001	Rank-1	Rank-5	Reg.	*UR*
10K0K	sfmx	97.15	97.45	69.39	33.04	81.63	90.35	87.17	82.47
	sfmx+m-L_2	97.00	97.88	73.00	44.78	83.77	91.49	90.21	84.68
10K10K	sfmx	–	97.85	72.96	49.22	82.38	90.46	85.87	85.25
	sfmx+m-L_2	97.08	97.85	74.07	46.27	83.70	91.74	89.48	84.10
	FTL (Ours)*	96.72	98.33	80.25	54.95	85.88	92.83	92.27	88.16
10K30K	sfmx	–	97.80	74.03	47.93	83.04	91.25	86.14	85.47
	sfmx+m-L_2	97.13	98.08	76.92	47.17	84.81	91.93	90.60	86.40
	FTL (Ours)*	96.87	98.42	81.80	61.04	86.08	92.62	91.76	88.72
10K50K	sfmx	–	97.93	72.87	49.04	82.40	91.15	85.28	84.21
	sfmx+m-L_2	97.32	98.10	78.52	53.44	84.95	92.17	90.24	87.11
	FTL (Ours)*	96.95	98.48	82.60	62.60	86.53	93.08	92.08	89.36
60K0K	sfmx	97.52	98.30	82.75	62.33	87.11	93.78	90.43	89.54
	sfmx+m-$L2$	**97.90**	**98.85**	**86.38**	**74.44**	**89.34**	**94.65**	**93.68**	**93.46**

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

Method	Ext	#Models	Base	Novel
MCSM [45]	YES	3	–	61.0
Cheng et al. [5]	YES	4	99.74	**100**
Choe et al. [6]	NO	1	$\geq$ 95.00	11.17
UP [11]	NO	1	99.80	77.48
Hybrid [44]	NO	2	99.58	**92.64**
DM [33]	NO	1	–	73.86
FTL (Ours)	NO	1	99.21	92.60

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".

Method	Acc	Method	Acc
L-Softmax [25]	98.71	ArcFace [8]	99.53
VGG Face [28]	98.95	FaceNet [32]	**99.63**
DeepID2 [36]	99.15	CosFace [41]	**99.73**
NormFace [40]	99.19	sfmx	98.60
CenterLoss [42]	99.28	sfmx + L_2	98.53
SphereFace [24]	99.42	sfmx + m-L_2 (Ours)	99.18
RangeLoss [50]	99.53	FTL (Ours)	99.55

Table 3. Performance comparisons on LFW. Methods of sfmx, sfmx+L_2, sfmx+m-L_2 are our implementations.

classes while monitoring the performance on base classes.

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-L_2 regularization already provides good results denoted as sfmx+m-L_2. 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].

Test →	Verification		Identification		
Method ↓	0.01	0.001	1	5	10
PAMs [26]	82.6	65.2	84.0	92.5	94.6
DR-GAN [37]	83.1	69.9	90.1	95.3	–
FF-GAN [48]	85.2	66.3	90.2	95.4	–
TA [7]	93.9	–	92.8	–	98.6
TPE [31]	90.0	81.3	86.3	93.2	97.7
NAN [46]	94.1	88.1	95.8	98.0	98.6
sfmx	91.5	77.4	92.4	96.4	97.3
sfmx + m-L_2 (Ours)	92.5	80.2	93.9	97.2	97.9
FTL (Ours)	93.5	82.9	94.8	97.8	98.3
FTL + MP (Ours)	94.3	85.1	95.1	97.8	98.4
FTL + MP + TA (Ours)	**95.3**	**91.2**	**96.0**	**98.3**	**98.7**

Table 4. Face recognition results on IJB-A. "MP" and "TA" represent media pooling and template adaptation. Verification and identification results are reported at different FARs and ranks.

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

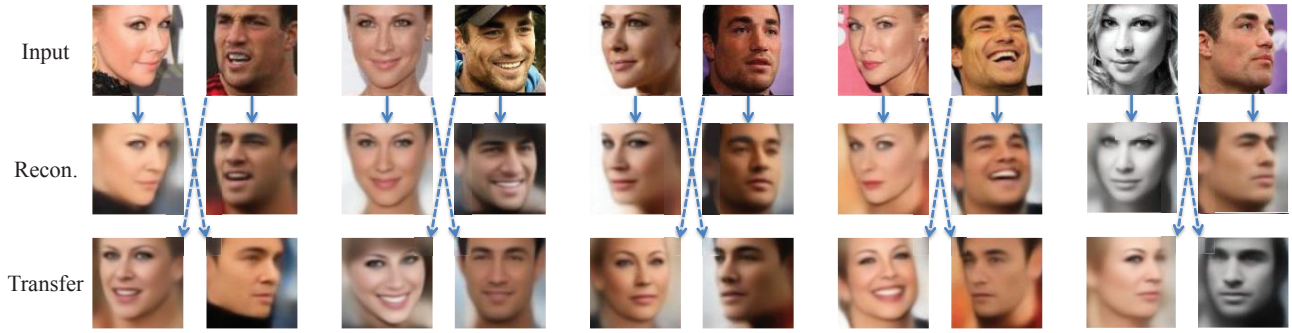


Figure 7. Feature transfer visualization between two classes for every two columns. The first row are the input, in which odd column denotes class 1: $\mathbf{x}_1$ and the even column denotes class 2: $\mathbf{x}_2$. The second row are the reconstructed images $\mathbf{x}'_1$ and $\mathbf{x}'_2$. In 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*.

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 $\mathbf{x}_{1,2}$, $\mathbf{x}'_{1,2}$, $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ 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: $\mathbf{g}_{12} = \mathbf{c}_2 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_1 - \mathbf{c}_1)$, 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 $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ denote the encoded features and the centers of two classes. Previous work generates a new representation as $\mathbf{g} = \mathbf{g}_1 + \alpha(\mathbf{g}_2 - \mathbf{g}_1)$ where identity and non-identity changes are mixed together. In our work, we can generate transitions of non-identity change as $\mathbf{g} = \mathbf{c}_1 + \alpha\mathbf{Q}\mathbf{Q}^T(\mathbf{g}_2 - \mathbf{c}_2)$ and identity change as $\mathbf{g} = \mathbf{g}_1 + \alpha(\mathbf{c}_2 - \mathbf{c}_1)$. 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

[1] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *JAIR*, 2002.

[2] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*, 2012.

[3] J. Chen and X. Liu. Transfer learning with one-class data. *Pattern Recognition Letters*, 37:32–49, February 2014.

[4] J. Chen, X. Liu, P. Tu, and A. Aragones. Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters*, 34(15):1964–1970, November 2013.

[5] Y. Cheng, J. Zhao, Z. Wang, Y. Xu, K. Jayashree, S. Shen, and J. Feng. Know you at one glance: A compact vector representation for low-shot learning. In *ICCV workshop*, 2017.

[6] J. Choe, S. Park, K. Kim, J. Hyun Park, D. Kim, and H. Shim. Face generation for low-shot learning using generative adversarial networks. In *ICCV workshop*, 2017.

[7] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. In *FG*, 2017.

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[9] M. Dixit, R. Kwitt, M. Niethammer, and N. Vasconcelos. AGA: Attribute-guided augmentation. In *CVPR*, 2017.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[11] Y. Guo and L. Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, 2017.

[12] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[13] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014.

[14] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017.

[15] H. He and E. A. Garcia. Learning from imbalanced data. In *TKDE*, 2009.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] G. V. Horn, O. M. Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist challenge 2017 dataset. In *CVPR Workshop*, 2017.

[18] G. V. Horn and P. Perona. The devial is in the tails: Fine-grained classification in the wild. In *arXiv:1709.01450*, 2017.

[19] C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

[20] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[21] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *CVPR*, 2015.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] Y. LeCun, C. Cortes, and C. J.C. Burges. The MNIST database of handwritten digits. Technical report, 1998.

[24] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. SphereFace: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[25] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[26] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *CVPR*, 2016.

[27] S. J. Pan and Q. Yang. A survey on transfer learning. In *TKDE*, 2009.

[28] O. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[29] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[31] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *BTAS*, 2016.

[32] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[33] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, and G. Lavrentyeva. Doppelganger mining for face representation learning. In *ICCV workshop*, 2017.

[34] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016.

[35] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *ICCV*, 2017.

[36] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[37] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017.

[38] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[39] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *CoRR*, 2016.

[40] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l_2 hypersphere embedding for face verification. *arXiv preprint arXiv:1704.06369*, 2017.

[41] H. Wang, Y. Wang, Z. Zhou, X. Ji, and W. Liu. CosFace: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[42] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[43] X. Wu, R. He, Z. Sun, and T. Tan. A light CNN for deep face representation with noisy labels. *arXiv preprint arXiv:1511.02683*, 2015.

[44] Y. Wu, H. Liu, and Y. Fu. Low-shot face recognition with hybrid classifiers. In *ICCV workshop*, 2017.

[45] Y. Xu, Y. Cheng, J. Zhao, Z. Wang, L. Xiong, K. Jayashree, H. Tamura, T. Kagaya, S. Pranata, S. Shen, et al. High performance large scale face recognition with multi-cognition softmax and feature retrieval. In *ICCV workshop*, 2017.

[46] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.

[47] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint:1411.7923*, 2014.

[48] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017.

[49] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *ECCV*, 2016.

[50] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, 2017.