Co-Mining: Deep Face Recognition with Noisy Labels

Xiaobo Wang*, Shuo Wang*, Jun Wang, Hailin Shi, Tao Mei
JD AI Research, Beijing, China
{wangxiaobo8,wangjun492,shihailin,tmei}@jd.com, cudaconvnet@gmail.com

Abstract

Face recognition has achieved significant progress with the growing scale of collected datasets, which empowers us to train strong convolutional neural networks (CNNs). While a variety of CNN architectures and loss functions have been devised recently, we still have a limited understanding of how to train the CNN models with the label noise inherent in existing face recognition datasets. To address this issue, this paper develops a novel co-mining strategy to effectively train on the datasets with noisy labels. Specifically, we simultaneously use the loss values as the cue to detect noisy labels, exchange the high-confidence clean faces to alleviate the errors accumulated issue caused by the sample-selection bias, and re-weight the predicted clean faces to make them dominate the discriminative model training in a mini-batch fashion. Extensive experiments by training on three popular datasets (i.e., CASIA-WebFace, MS-Celeb-1M and VggFace2) and testing on several benchmarks, including LFW, CALFW, CPLFW, AgeDB, CFP, RFW, and MegaFace, have demonstrated the effectiveness of our new approach over the state-of-the-art alternatives. Our code is available at `http://www.cbsr.ia.ac.cn/users/xiaobowang/`.

1. Introduction

Datasets are of crucial to the development of face recognition. From the early CASIA-WebFace [45] to the more recent VggFace [27], MS-Celeb-1M [11], VggFace2 [5] and IMDb [36], face recognition datasets play a main role in driving the development of new techniques. Not only face recognition datasets become more diverse, but also the scale of data is growing tremendously. For instance, MS-Celeb-1M [11] contains about 10M images of 100K identities, far exceeding CASIA-WebFace [45] that only has 0.5M images from 10,575 individuals. Large-scale datasets together with the emergence of deep convolutional neural networks technique have led to the immense success of face recognition in recent years. However, these public large-scale datasets probably contain noisy faces because most of them are automatically collected via image search engines or from movies. Figure 1 gives an example of noisy faces in MS-Celeb-1M [11]. Detailedly, we refer to the samples whose identities are mislabeled/incorrectly annotated as noisy faces and denote their labels as noisy labels. Such noisy labels can fall into two types, "Label Flips (closed-set)" and "Outliers (open-set)". More specifically, a label flip (close-set) noisy label occurs when a noisy face possesses a true class that is contained within the set of known classes in the training data (*e.g.*, face in the blue box of Figure 1). In contrast, an outlier (open-set) noisy label occurs when a noisy face possesses a true class that is not contained within the set of known classes in the training data (*e.g.*, faces in the red box of Figure 1).

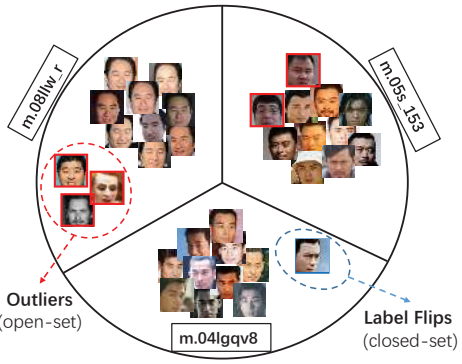


Figure 1. An illustration of deep face recognition with noisy labels in MS-Celeb-1M [11]. "Label Flips" means that the faces have erroneously been given the label of another class within the dataset. "Outlier" means that faces do not belong to any of the classes under consideration, but mistakenly have one of their labels.

It is well-known that noisy labels inevitably degenerate the robustness of learned models, especially for deep CNNs. Wu *et al.* [43] adopt a semantic bootstrapping rule to select the clean samples via the prediction consistency. Deng *et al.* [8] resort to the feature dis-similarities to drop the noisy faces and further manually check the unreliable ones. The trillion-pairs consortium [1] has published a relatively clean version of MS-Celeb-1M. These methods try to clean

*These authors contributed equally to this work.

a noisy face dataset into a well-annotated one. However, the process is prohibitively expensive and time-consuming. Taking CASIA-WebFace [45] as an example, until now, the data clean processing is still on the agenda [2]. This motivates researchers to shift their attention to resort to cheap but imperfect alternatives. Miyato *et al.* [22] add both explicit and implicit regularizations to overcome the noisy labels issue, but the permanent regularization bias make the learned classier barely reaches the optimal performance. Patrini *et al.* [29] try to estimate the label transition matrix, but it is difficult to estimate accurately when the number of classes is large. Wang *et al.* [42] use the Local Outlier Factor (LOF) algorithm [4] to detect the noise samples. But the process is slow on large-scale dataset. Jiang *et al.* [16] design a self-pace learning strategy, which is similar to co-training, thus it may suffer from the errors accumulated issue. Malach *et al.* [21] trains two networks simultaneously, but it does not explicitly address noisy labels. Recently, Han *et al.* [12] and Yu *et al.* [46] develop a co-teaching strategy to directly handle the noisy labels for training models.

Although the above approaches have achieved promising results on noisy label problems, they mainly have three shortcomings: 1) Many works [42, 22, 29] can not detect the noisy labels effectively and accurately, especially for large-scale face recognition problem. 2) Most of works [16, 21, 42] are not aware of the errors accumulated issue caused by the sample-selection bias. 3) Existing works [12, 46] simply try to distinguish clean samples from the noise ones, without considering the importance of clean samples for learning discriminative features.

To overcome the aforementioned shortcomings, this paper proposes a novel Co-Mining strategy, which identifies the training samples into three parts, noisy faces, high-confidence clean faces and clean faces. Specifically, it uses the loss values as the cue to effectively and accurately detect the noisy faces, exchanges the high-confidence clean faces to alleviate the errors accumulated issue, and re-weights the clean faces to make them more important to train the discriminative CNN models. To sum up, the main contributions of this paper can be summarized as follows:

- We identify the samples in noisy face recognition dataset as three parts, *i.e.*, the noisy faces, the high-confidence clean faces and the clean faces.
- We propose a novel co-mining framework, which employs two peer networks to detect the noisy faces, exchanges the high-confidence clean faces and re-weights the clean faces in a mini-batch fashion.
- We emphasize the open-set evaluation for face recognition and conduct extensive experiments on both synthetic and real-world benchmarks, which have verified the effectiveness of our new approach over the state-of-the-art alternatives.

2. Related Work

2.1. Deep Face Recognition

Existing deep face recognition mainly comes from three aspects: *i.e.*, large-scale datasets, effective architectures and loss functions. For large-scale datasets, from the early CASIA-WebFace [45] to the recent MS-Celeb-1M [11] and VggFace2 [5] , the diverse and the scale of face recognition datasets increase gradually and play a main role in boosting the development of new techniques. With these datasets, the effective and representative architectures like VGGNet [31], GoogleNet [34], ResNet [13], AttentionNet [38] and MobileFaceNet [7] have been introduced or devised for deep face recognition. For loss functions, the metric learning loss functions like the contrastive loss [33, 44] and the triplet loss [34] may be good candidates. But they usually suffer from high-computational cost and slow convergence. Recently, researchers start to shift their attention to the classical softmax loss, and several margin-based softmax losses [20, 37, 41, 8, 40] have been exploited. Among them, ArcSoftmax loss [8] is probably the most prevalent one in the current stage. And the success of it depends on the well-cleaned datasets. However, large-scale datasets inevitably contain noisy labels, especially when they are automatically collected from Internet. On this account, while with a variety of architectures and loss functions, we still have a limited understanding of the source and consequence of label noise inherent in existing face recognition datasets.

2.2. Training with Noisy Labels

Learning with noisy labels has drawn much attention recently in deep learning because it is a data-driven approach and accurate label annotation is quite expensive. Mnih and Hinton [23] propose two robust loss functions for noisy label aerial images. However, it is only applicable for binary classification. Sukhbaatar *et al.* [32] consider multi-class classification for modeling class dependent noise distribution. Wu *et al.* [43] propose a semantic bootstrap strategy, which re-labels the samples by the predictions, and then does back propagation. Wang *et al.* [42] detect the noisy labels by using the discrimiantive features and design a iterative learning framework for training with open-set noisy labels. Jiang *at al.* [16] resort to an extra pre-trained teacher network to filter out noisy instances for its student network. Malach *at al.* [21] propose a method to update the parameters only using the samples which have different prediction from two classifiers. Han *et al.* [12] develop a co-teaching strategy to robustly train the deep neural networks. Although these strategies have been studied for noisy label problem, most of them are not designed for deep face recognition with large number of classes. It is still an open issue for massive noisy labels in face recognition.

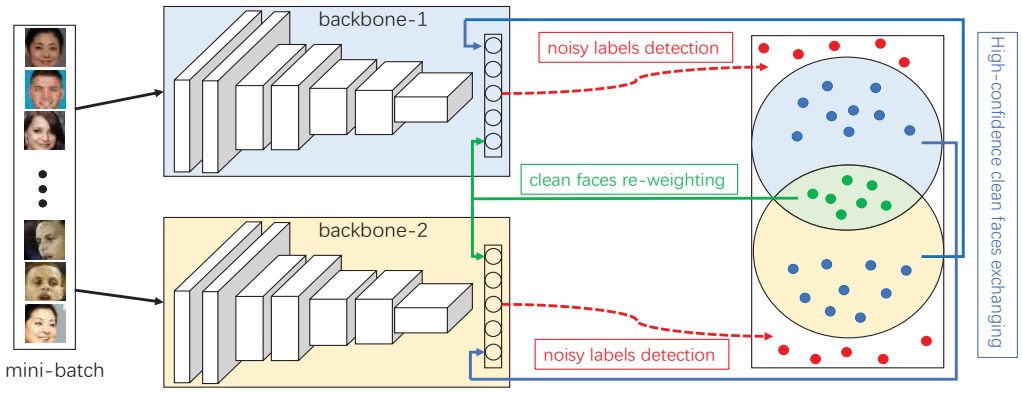


Figure 2. The framework of the proposed Co-Mining strategy. For each peer network, it uses loss values as the cue to detect noisy labels. In consequence, according to the sorting of loss values, the mini-batch samples can be identified as three parts, noise faces (red dots), high-confidence clean faces (blue dots) and clean faces (green dots). For the noise faces, they will be dropped because they may deteriorate the performance heavily. For the high-confidence clean faces, we exchange them to alleviate the accumulated errors caused by the potential sample-selection bias. For the clean faces, we re-emphasize them to learn discriminative CNN features for face recognition.

3. Propose Method

Our goal is to learn discriminative CNN features from a dataset with noisy labels, wherein the noise type and noise rate are usually unknown in advance. To achieve this goal, we propose a novel Co-Mining strategy to simultaneously remove the bad influences of noisy labels, alleviate the issue of accumulated errors, and emphasize the gains of clean instances. As illustrated in Figure 2, our framework consists of three major modules: 1) Using loss values as the cue to detect the noisy labels; 2) Exchanging the high-confidence clean faces to prevent the potential errors accumulated issue; 3) Re-weighting the clean faces to make them dominant to learn discriminative CNN features.

3.1. Noisy Labels Detection

To detect noisy labels, current methods resort to estimate the noise transition matrix or use the density-based outlier detection algorithm. For example, on top of the softmax layer, Goldberger *et al.* [10] add an additional softmax layer to model the noise transition matrix. Patrini *et al.* [29] leverage a two-step solution to estimating the noise transition matrix heuristically. Wang *et al.* [42] adopt the density-based outlier detection LOF algorithm [4] to iteratively detect the noisy labels. However, neither the noise transition matrix nor the density-based outlier detection is hard to filter out noisy labels accurately, especially when the number of classes is large. Fortunately, recent studies [12, 46] on the memorization effects of deep neural networks show that they would first memorize training data of clean labels and then those of noisy labels. This motivates us to use loss values as the cue to detect noisy labels. Intuitively, when labels are correct, small-loss instances are more likely to be the ones which are correctly labeled. Therefore, if we train our classifier only using small-loss instances in each mini-batch data, it should be resistant to noisy labels.

Assume that we have estimated the noise rate r of a face recognition dataset. M is the mini-batch size. Similar to the works [21, 14, 49], our method also maintains two networks simultaneously. That being said, in each mini-batch of data, each peer network views its small-loss instances as the useful knowledge and drops about $[r * M]$ numbers of big-loss instances as the distractors, leaving the rest of samples into two parts, intersected faces and non-intersected faces of these two peer networks. For the intersected faces, since two peer networks predict them as clean faces, we have reason to believe that they are clean enough for deep face recognition. For the non-intersected ones, they have high confidence to be clean faces. But they may also be noisy faces, especially when the noise rate r is estimated inaccurately[1]. To sum up, we have identified the faces in each mini-batch as three parts, *i.e.*, noise faces, high-confidence faces and clean faces.

For the loss function, several margin-based softmax loss functions [20, 41, 37, 8] have been proposed in recent years. Without loss of generality, we adopt the Arc-Softmax loss [8] as the baseline. Specifically, for each normalized feature $\boldsymbol{x}$, the Arc-Softmax loss is defined as follows:

$$\mathcal{L}_{Arc} = -\log \frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)} + \sum_{k\neq y}^{K} e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}, \quad (1)$$

where $w_k, k \in \{1, \ldots, K\}$ is the k-th normalized classifier. y is the corresponding label. m is the margin parameter to learn discriminative features and s is a preset scale parameter. For more details, please refer to [8].

[1]Even though the noise rate r of a dataset can be estimated accurately, in each mini-batch, the noise rate r_t is hard to predict.

3.2. High-Confidence Clean Faces Exchanging

According to the above discussion, we donate the sampled instances of each peer network as $\mathcal{S}_n^1$ and $\mathcal{S}_n^2$, respectively. Since different networks can generate different decision boundaries and then have different learning abilities. Thus, when training on noisy labels, they have different abilities to filter out the noisy labels. In other words, the sampled instances of each peer network, $\mathcal{S}_n^1$ and $\mathcal{S}_n^2$, are different. As a consequence, we can further divide the sampled instances into the intersected faces $\mathcal{S}_n^1 \cap \mathcal{S}_n^2$ and the respective ones, $\mathcal{S}_n^1 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)$ and $\mathcal{S}_n^2 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)$.

In this part, we discuss the respective faces of each peer network. We identify these faces as high-confidence clean faces because they may also be noisy labels due to inaccurately estimated noise rate in each mini-batch. If we directly feed back to itself in the second mini-batch of data, the errors should be increasingly accumulated. To alleviate this issue, we expect to exchange them. That is, to update parameters of $\mathbf{\Theta}^1$ (resp. $\mathbf{\Theta}^2$) using the high-confidence clean faces $\mathcal{S}_n^2 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)$ (resp. $\mathcal{S}_n^1 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)$) selected from its peer network $\mathbf{\Theta}^2$ (resp. $\mathbf{\Theta}^1$). This process is derived from co-training [6], and these two networks will adaptively correct the training errors by its peer network. Take "peer-review" as a supportive example, when students check their own exam papers, it is hard for them to find any errors because they have some personal bias for the answers. Luckily, they can ask peer classmates review their papers. Then, it becomes much easier to find their potential faults. To sum up, as the errors from one network will not be directly transferred back itself, we can expect that exchanging the high-confidence clean faces can deal with the errors accumulated issue compared with the self-evolving one.

3.3. Clean Faces Re-weighting

To the intersected faces $\mathcal{S}_n^1 \cap \mathcal{S}_n^2$, since two peer networks agree that they are clean faces, we have reason to believe that they are labeled correctly. Thus we should concentrate on them when training the models. Actually, in face recognition with noisy labels, the center task is to find those clean faces and mainly using them to learn discriminative features. In this paper, we adopt two peer networks to collaboratively find clean faces and adaptively re-weight them. Thus we call it Co-Mining strategy. To emphasize the contributions of clean faces, we introduce a novel re-weighting module. Specifically, we reduce the baseline probability:

$$p_{Arc} = \frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)} + \sum_{k\neq y}^{K} e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}} \tag{2}$$

to our re-weighting one:

$$p_{Our} = \frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)} + \sum_{k\neq y}^{K} g(\mu) * e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}, \tag{3}$$

Algorithm 1: Co-Mining Algorithm

Input: The training set $\mathcal{S}$, model parameters $\mathbf{\Theta}^1$ and $\mathbf{\Theta}^2$, learning rate λ, fixed noise rate r, epoch T_k and T, iteration N in each epoch.

for $t = 1, 2, \ldots, T$ **do**
  Shuffle the training set $\mathcal{S}$
  for $n = 1, 2, \ldots, N$ **do**
    1. Fetch mini-batch $\mathcal{S}_n$ from the training set $\mathcal{S}$;
    2. Sample (1-r_t)% of small-loss faces: $\mathcal{S}_n^1 = \text{argmin}_{\mathcal{S}_n':|\mathcal{S}_n'|\geq r_t|\mathcal{S}_n|} \mathcal{L}_{Arc}^1(\mathcal{S}_n')$ and $\mathcal{S}_n^2 = \text{argmin}_{\mathcal{S}_n':|\mathcal{S}_n'|\geq r_t|\mathcal{S}_n|} \mathcal{L}_{Arc}^2(\mathcal{S}_n')$;
    3. Exchange the high-confidence clean faces and compute the loss $\mathcal{L}_{Arc}^1(\mathcal{S}_n^2 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2))$ and $\mathcal{L}_{Arc}^2(\mathcal{S}_n^1 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2))$ by Eq. (1);
    4. Compute the clean faces loss $\mathcal{L}_{Our}(\mathcal{S}_n^1 \cap \mathcal{S}_n^2)$ by Eq. (5);
    5. Update the parameters: $\mathbf{\Theta}^1 := \mathbf{\Theta}^1 - \lambda\nabla[\mathcal{L}_{Arc}^1(\mathcal{S}_n^2 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)) + \mathcal{L}_{Our}(\mathcal{S}_n^1 \cap \mathcal{S}_n^2)]$ and $\mathbf{\Theta}^2 := \mathbf{\Theta}^2 - \lambda\nabla[\mathcal{L}_{Arc}^2(\mathcal{S}_n^1 \setminus (\mathcal{S}_n^1 \cap \mathcal{S}_n^2)) + \mathcal{L}_{Our}(\mathcal{S}_n^1 \cap \mathcal{S}_n^2)]$;
  end
  Update the $r_t = \min\{\frac{t}{T_k}r,\ r\}$;
end

Output: Model parameters $\mathbf{\Theta}^1$ and $\mathbf{\Theta}^2$.

where $\boldsymbol{x} \in \mathcal{S}_n^1 \cap \mathcal{S}_n^2$. $g(\mu) \geq 1$ is a re-weighting function, which is defined as follows:

$$g(\mu) = e^{s\mu(\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})+1)}, \tag{4}$$

where μ is a non-negative value. Obviously, when $\mu = 0$ (*i.e.*, $g(\mu) = 1$), our re-weighting probability (3) becomes identical to the baseline Arc-Softmax (2). Because the cross-entropy loss $-\log(p)$ is a monotonically decreasing function, reducing the baseline probability (*i.e.*, $p_{Our} \leq p_{Arc}$) will increase the importance of clean samples. In that way, to the clean faces, their loss function will be changed into:

$$\mathcal{L}_{Our} = -\log \frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}+m)} + \sum_{k\neq y}^{K} g(\mu) * e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}. \tag{5}$$

Meanwhile, according to the memory mechanism [3], the deep models usually tend to memorize the easy instances first and gradually adapt to hard instances when training epochs become large. To rectify the problem of overfitting on noisy labels eventually, we keep more instances in the mini-batch at the beginning of training. Then, we gradually increase the drop rate, so that we can keep clean instances and drop those noisy ones before the networks memorize them. Specifically, we adaptively set the noise rate $r_t = \min\{\frac{t}{T_k}r, r\}$, where T_k is predefined. For clarity, the whole scheme of our framework is summarized in Algorithm 1.

4. Experiments

4.1. Datasets

Training Data. This paper involves three popular training datasets, including CASIA-WebFace [45], MS-Celeb-1M [11] and VggFace2 [5]. The original of these three datasets consist of noisy labels with different unknown noise rate. To the synthetic noise experiments, we use a clean version of CASIA-WebFace, *i.e.*, CASIA-WebFace-Clean [2], to train with different synthetic noises.

Test Data. We use seven benchmarks, including LFW [15], CALFW [51], CPLFW [50], AgeDB [24], CFP [30], RFW[39] and MegaFace [17, 25] as the test data. LFW contains 13,233 web-collected images from 5,749 different identities, with large variations in pose, expression and illuminations. CALFW [51] was collected by crowdsourcing efforts to seek the pictures of people in LFW with age gap as large as possible on the Internet. CPLFW [50] is similar to CALFW, but from the perspective of pose difference. AgeDB [24] contains images annotated with accurate to the year, noise-free labels. CFP [30] consists of collected images of celebrities in frontal and profile views. RFW [39] is a benchmark for measuring racial bias, which consists of four test subsets, namely Caucasian, Asian, Indian and African. MegaFace [17] aims at evaluating the face recognition performance at the million scale of distractors, which includes gallery set and probe set. In this study, we use the Facescrub [26] as the probe set.

Dataset Overlap Removal. In face recognition, it is very important to perform open-set evaluation, *i.e.*, there should be no overlapping identities between training set and test set. To this end, we need to carefully remove the overlapped identities between the employed training datasets (*i.e.*, CASIA-WebFace [45], MS-Celeb-1M [11] and VggFace2 [5]) and the test datasets (including LFW [15], CALFW [51], CPLFW [50], AgeDB [24], CFP [30], RFW [39] and MegaFace [17]. For the overlap identities removal tool, we use the publicly available script provided by [37] to check whether if two names are of the same person. As a consequence, we remove 696 identities from the training set CASIA-WebFace, 14,718 identities from MS-Celeb-1M, and 1,514 peoples from VggFace2. For clarity, we donate the refined training datasets as CASIA-R, CASIA-Clean-R, MsCeleb-R and VggFace2-R, respectively. Important statistics of the datasets are summarized in Table 1. To be rigorous, all the experiments in this paper are based on the refined training datasets. To encourage more researchers to abide by the open-set protocol, the overlapping lists and the refined datasets are publicly available.

	Datasets	Identities	Images
Training	CASIA-R [45]	9,879	0.43M
	CASIA-Clean-R [2]	9,879	0.38M
	MsCeleb-R [11]	85,173	7.03M
	VggFace2-R [5]	7,617	2.71M
Test	LFW [15]	5,749	13,233
	CALFW [51]	5,749	12,174
	CPLFW [50]	5,749	11,652
	AgeDB [24]	568	16,488
	CFP [30]	500	7,000
	RFW [39]	11,430	40,607
	MegaFace [17]	530(P)	1M(G)

Table 1. Face datasets for training and testing. "(P)" and "(G)" refer to the probe and gallery set, respectively.

4.2. Experimental Settings

Data Processing. We detect the faces by adopting the FaceBoxes detector [48] and localize five landmarks (two eyes, nose tip and two mouth corners) through a simple 6-layer CNN [9]. The detected faces are cropped and resized to 120×120, and each pixel (ranged between [0,255]) in RGB images is normalized by subtracting 127.5 and divided by 128. For all the training faces, they are horizontally flipped with probability 0.5 for data augmentation.

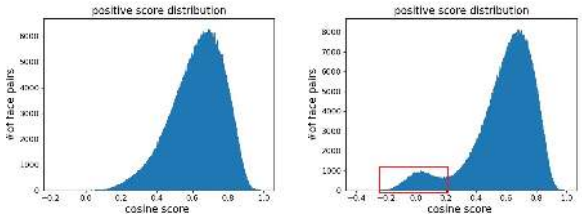
CNN Architecture & Loss Function. In face recognition, there are many kinds of network architectures [20, 7, 36] and several loss functions [20, 37, 8]. To be fair, the CNN architecture and the employed loss function should be the same to test different methods with noisy labels. Without loss of generality, we use the MobileFaceNet [7] and the Arc-Softmax loss [8] as the baseline. To the margin m and scale s, we set 0.5 and 32, respectively.

Training. All the CNN models are trained with stochastic gradient descent (SGD) and trained from scratch, with the batch size of 128 on 4 P40 GPUs parallelly, total batch size 512. All experiments in this paper are implemented by PyTorch [28]. The weight decay is set to 0.0005 and the momentum is 0.9. The learning rate is initially 0.1 and divided by 10 at the 6, 12, 17 epochs, and we finish the training process at 20 epochs. T_k is empirically set to 10.

Test. At test stage, only original image features are employed (512-dimension). We use the backbone-1 of two peer networks to extract face features. For the evaluation metric, cosine similarity is utilized. We follow the unrestricted with labelled outside data protocol [15] to report the performance on LFW [15], CALFW [51], CPLFW [50], AgeDB [24], CFP [30] and RFW [39]. Moreover, we also report the BLUFR protocol [18] on LFW [15]. On Megaface [17] challenge, face identification and verification are conducted by ranking and thresholding the scores. Specifically, for face identification, the Cumulative Match Characteristics (CMC) curves are adopted to evaluate the Rank-1 accuracy. For face verification, the Receiver Operating Characteristic (ROC) curves are adopted.

	Strategy			LFW	LFW @1e-3	LFW @1e-4	CALFW	CPLFW	AgeDB	CFP	Average
	NLD	HCCFE	CFR								
CASIA-R				98.71	96.96	91.79	88.21	79.18	91.46	91.10	91.05
CASIA-Clean-R				**98.80**	**97.59**	**93.05**	**88.85**	**80.31**	**92.21**	**91.30**	**91.73**
CASIA-Clean-R (symmetric=0.1)				98.01	96.69	92.48	87.48	79.43	91.15	91.08	90.90
	✓			97.98	96.80	92.67	88.06	79.50	91.48	91.32	91.11
	✓	✓		98.33	97.33	92.87	88.28	79.60	91.70	91.68	91.39
	✓	✓	✓	**98.91**	**97.91**	**94.42**	**89.48**	**80.03**	**92.68**	**91.75**	**92.16**
CASIA-Clean-R (symmetric=0.2)				88.56	66.30	27.14	76.65	68.56	80.53	80.22	69.70
	✓			96.90	78.31	37.43	84.81	72.58	87.55	81.75	77.04
	✓	✓		97.78	81.84	38.45	85.35	74.11	88.33	**84.25**	78.58
	✓	✓	✓	**98.23**	**92.85**	**61.41**	**87.76**	**76.85**	**90.95**	79.61	**83.95**
CASIA-Clean-R (symmetric=0.3)				81.89	0.05	0.04	50.76	53.31	71.05	72.64	58.81
	✓			95.53	50.84	11.21	83.35	69.03	85.41	75.34	67.24
	✓	✓		96.66	66.74	27.40	83.76	71.08	86.61	78.20	72.92
	✓	✓	✓	**97.66**	**87.67**	**55.68**	**86.63**	**73.50**	**88.58**	**79.54**	**81.32**

Table 2. Verification results (%) with different strategies. NLD refers to the Noisy Labels Detection. HCCFE is the High-Confidence Clean Faces Exchanging. CFR means the Clean Faces Re-weighting. The bold number in each column of sub-boxes represents the best result.



(a) CASIA-Clean-R (b) CASIA-R

Figure 3. **From left to right**: Cosine similarity distributions of all positive pairs from CASIA-Clean-R and CASIA-R, respectively. The face pairs in the red box are possibly with noisy labels.

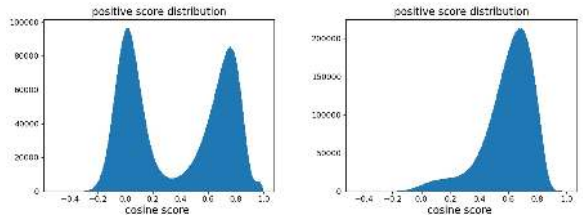
4.3. Experiments on Synthetic Data

To begin with, we use MobileFaceNet [7] and ArcSoftamx loss [8] as the baseline to separately train on the noisy CASIA-R [45] and its clean version CASIA-Clean-R [2]. From the first two lines of Table 2, we can observe that the model trained on CASIA-Clean-R performs better than trained on CASIA-R, even though CASIA-Clean-R is with smaller size of training images. So it can be concluded that noisy faces are evil for training models. Further, we use the SEResNet100-IR model [8] pre-trained on MS1M-deepglint [1] to show the cosine similarity distribution of all positive face pairs from these two datasets. Figure 3 displays the histograms and we can verify that CASIA-R indeed contains noisy faces. Next, we use the CASIA-Clean-R with different synthetic noises to show the robustness and effectiveness of our method.

The effectiveness of Noisy Labels Detection (NLD). We do the experiments with/without noisy labels detection under different synthetic noise rate. We use the symmetry flipping [35] to simulate the noisy labels, where labelers may make mistakes only within very similar classes. From the first two lines of each sub-boxes in Table 2, we can observe that with the noisy labels detection (NLD), the average performance of LFW, CALFW, CPLFW, AgeDB, CFP has been improved from 90.90 to 91.11 under 0.1 noise rate, 69.70 to 77.04 under 0.2 noise rate, and 58.81 to 67.24 under 0.3 noise rate. Eventually, the experiments have validated the effectiveness of our noisy labels detection.

The effectiveness of High-Confidence Clean Faces Exchanging (HCCFE). We further add the HCCFE strategy to validate whether it can alleviate the errors accumulated issue caused by sample-selection bias. From the third row of each sub-boxes in Table 2, we can see that HCCFE can further boost the performance, about 0.2% improvement under 0.1 noise rate, 1.5% improvement under 0.2 noise rate, and 5% improvement under 0.3 noise rate. These average accuracy improvements, compared to those without this strategy, can be interpreted as the contribution of the HCCFE module. Particularly, with large noise rate, the average accuracy increases significantly, which indicates that the module can alleviate the errors accumulated issue effectively.

The effectiveness of Clean Faces Re-weighting (CFR). We finally add the re-weighting strategy on the clean faces to make them dominant to train the models. From the last row of each sub-boxes in Table 2, we can see that under different noise rate, the CFR strategy is helpful for boosting the performance. Thus we can conclude that the clean faces are more important when training with noisy faces, especially with large noise rate, *e.g.*, in the case of symmetric=0.3, the importance of clean faces are even more obvious. After tuning the importance parameter μ with several values (*i.e.*, from 0 to 0.4, with stepvalue 0.1) on CASIA-Clean-R, we set $\mu = 0.1$ in the subsequent experiments.



(a) MsCeleb-R (b) VggFace2-R

Figure 4. **From left to right**: Cosine similarity distributions of all positive pairs from MsCeleb-R and VggFace2-R, respectively.

4.4. Experiments on Real-World Data

Similar to CASIA-R, we show the cosine similarity distributions of all positive face pairs from MsCeleb-R [11] and VggFace2-R [5] in Figure 4. From the sub-figures, we notice that MsCeleb-R is with much noise than VggFace2-R. The rest of experiments are conducted based on these two real-world training datasets. According to the statistics in Figure 4, we empirically use the threshold 0.3 to indicate the noisy labels. As a consequence, we set the noise rates of MsCeleb-R and VggFace2-R as 0.4 and 0.05, respectively. For the noise rate r of a specific dataset, one can also infer it by using validation sets, as previous works [47, 19] do.

4.4.1 Compared Method

We compare our method with the recently proposed six state-of-the-art competitors, including:

MentorNet (MN) [16]. An auxiliary teacher network is pre-trained and used to drop noisy faces for its student network. Then, student network is used for face recognition.

De-Coupling (DC) [21]. This method updates the parameters only using the samples which have different predictions from two classifiers.

DualNet (DN) [14]. Two parallel neural networks are coordinated to learn complementary features. The iterative training strategy is adopted to make the two networks cooperate with each other.

Mutual-Learning (ML) [49]. An ensemble of networks learn collaboratively in a mutual learning strategy and teach each other throughout the training process.

Co-Teaching (CT) [12]. The method trains two neural networks simultaneously, and let them teach each other in each mini-batch. Each network back propagates the selected data without distinction by its peer network and updates itself.

Co-Teaching+ (CT+) [46]. This method is similar to Co-Teaching [12], but updates the parameters only using the disagreement of the two predictions.

For all these competitors, the backbone architecture is the MobileFaceNet [7] and is equipped with Arc-Softmax loss [8].

Method	CALFW	CPLFW	AgeDB	CFP	Ave.
Baseline	91.95	81.13	92.76	89.74	88.89
MN[16]	92.94	83.49	94.66	92.58	90.91
DC[21]	90.28	84.01	93.80	92.81	90.22
DN[14]	89.79	76.38	92.26	84.90	85.83
ML[49]	91.89	82.44	93.70	91.54	89.89
CT[12]	92.08	84.83	94.36	92.35	90.90
CT+[46]	92.14	84.26	94.80	92.34	90.88
Our	**93.28**	**85.70**	**95.80**	**93.32**	**92.02**

Table 3. Results (%) of different methods training on MsCeleb-R.

Method	CALFW	CPLFW	AgeDB	CFP	Ave.
Baseline	90.11	86.30	92.81	95.50	91.18
MN[16]	90.14	85.41	92.70	95.20	90.86
DC[21]	90.23	86.14	93.90	95.85	91.53
DN[14]	90.26	85.01	93.33	95.05	90.91
ML[49]	90.08	86.00	93.35	95.51	91.23
CT[12]	89.90	85.05	92.05	95.05	90.62
CT+[46]	89.43	85.23	92.50	95.41	90.64
Our	**91.06**	**87.31**	**94.05**	**95.87**	**92.07**

Table 4. Results (%) of different methods training on VggFace2-R.

4.4.2 Results on CALFW, CPLFW, AgeDB and CFP

We use two training sets, MsCeleb-R and VggFace2-R, to separately train the deep CNN models. Tables 3-4 provide the quantitative results of the baseline, the competitors and our method on the test sets CALFW [51], CPLFW [50], AgeDB [24] and CFP [30], respectively. From the numbers, we observe that most of the competitors are better than the baseline when training on the MsCeleb-R dataset. It is because that the MsCeleb-R is very noisy. Most of the competitors can filter out the noisy labels effectively and thus result in higher performance. To our method, we not only detect the noisy labels effectively, but also avoid the errors accumulated issue and absorb the gains of clean faces. Therefore, our method can reach higher performance than both the baseline and the competitors. Specifically, we achieve about 2% average improvement over the baseline and 1% average improvement over the best competitor Co-Teaching algorithm [12]. While training on the VggFace2-R dataset, the competitors are slightly lower than the baseline. The possible reason is that the noise rate of VggFace2-R is small. They may drop some essential clean faces during the training process. In contrast, our method may also discard some important clean faces, but we have re-weighted most of the remaining clean samples, thus we can also keep resulting in more discriminative features and achieve more promising performance. Specifically, we still achieve about 1% average improvement over the baseline, 0.5% average improvement over the best competitor Decoupling algorithm [21].

Method	RFW				Ave.
	Caucasian	Indian	Asian	African	
Baseline	93.83	81.83	86.83	83.16	86.14
MN[16]	95.46	87.50	87.16	83.83	88.48
DC[21]	95.66	84.50	88.33	84.83	88.33
DN[14]	89.83	79.83	80.33	78.16	82.06
ML[49]	91.99	84.33	87.33	83.83	86.87
CT[12]	94.00	86.16	87.66	84.66	88.12
CT+[46]	95.50	86.16	88.33	85.50	88.87
Our	**95.83**	**89.83**	**89.16**	**86.16**	**90.24**

Table 5. Results (%) of different methods training on MsCeleb-R.

Method	RFW				Ave.
	Caucasian	Indian	Asian	African	
Baseline	93.16	85.33	85.83	80.83	86.28
MN[16]	90.83	86.00	86.83	83.16	86.70
DC[21]	93.33	85.00	88.00	83.83	87.50
DN[14]	90.33	86.33	82.83	83.83	85.83
ML[49]	93.66	86.49	**90.00**	83.00	88.28
CT[12]	92.66	86.16	86.33	83.50	87.16
CT+[46]	92.33	86.49	86.00	79.33	86.03
Our	**94.83**	**87.83**	88.00	**85.33**	**88.99**

Table 6. Results (%) of different methods training on VggFace2-R.

Data	Method	MegaFace Rank1@1e6	MegaFace TPR@FAR=1e-6
MsCeleb-R	Baseline	84.56	87.72
	MN[16]	86.10	89.07
	DC[21]	86.52	88.90
	DN[14]	78.42	81.23
	ML[49]	83.25	86.46
	CT[12]	86.45	88.53
	CT+[46]	**87.46**	88.77
	Our	87.37	**89.69**

Table 7. Results (%) of different methods on MegaFace Challenge.

Data	Method	MegaFace Rank1@1e6	MegaFace TPR@FAR=1e-6
VggFace2-R	Baseline	78.04	83.06
	MN[16]	76.79	81.91
	DC[21]	78.95	82.45
	DN[14]	74.38	79.63
	ML[49]	78.90	82.48
	CT[12]	75.43	80.79
	CT+[46]	73.19	78.15
	Our	**81.51**	**86.07**

Table 8. Results (%) of different methods on MegaFace Challenge.

4.4.3 Results on RFW

Tables 5-6 display the performance comparison of all the methods on RFW test set. From the values, we can conclude that the results exhibit the same trends that emerged on previous test datasets. Concretely, when training on the MsCeleb-R dataset, most of the competitors are better than the baseline because they can filter out the noisy labels effectively and thus can reduce their influences. When training on the VggFace2-R dataset, the results are not much different. For our method, which simultaneously detects the noisy labels, exchanges the high-confidence clean faces and re-weights the clean ones, has show its superiority over the baseline and the state-of-the-art alternatives. Specifically, our method achieves 90.24 average accuracy when training on the MsCeleb-R dataset, and 88.99 average accuracy when training on the VggFace2-R datset, at least 1% average improvement over the second best one.

4.4.4 Results on MegaFace Challenge

Tables 7-8 show the identification and verification results on MegaFace dataset. In particular, compared with the baseline, the competitors have shown their strong abilities to filter out noisy labels and usually achieve better performance. For our method, we achieve about 2% improvement at both the Rank-1@1e6 identification rate and the verification TPR@FAR=1e-6 rate over the baseline. Compared with the competitors, the improvement of our method is not quite large, but is still better than them. Specifically, our method beats the competitor MentorNet [16] about 1.0% at Rank-1 identification rate and 0.7% verification rate when training on the MsCeleb-R, achieves about 2.5% at Rank-1 identification rate and 3.5% verification rate higher than the competitor Decoupling [21] when training on the VggFace2-R. To sum up, our co-mining strategy, which effectively detects the noisy labels, exchanges the high-confidence faces and adaptively concentrates on the clean ones, is inherently better than the state-of-the-arts.

5. Conclusion

In this paper, we have proposed a novel co-mining strategy to train the CNN models on large-scale face recognition datasets with noisy labels. Specifically, we identify the mini-batch samples into three parts, the noisy labels, the high-confidence clean faces and the clean faces. Next, to each part, we develop different strategies. For the noisy labels, we drop them to prevent the model degeneration problem caused by them. For the high-confidence clean faces, we exchange them to alleviate the errors accumulated issue. For the clean faces, we re-weight them and make them dominant to learn discriminative features. Extensive experiments on both the synthetic and real-world benchmarks have demonstrated the advantages of our new approach over the state-of-the-art alternatives.

References

[1] http://trillionpairs.deepglint.com/overview.

[2] https://github.com/ZhaoJ9014/face.evoLVe.PyTorch.

[3] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, 2017.

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, 2000.

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018.

[6] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *TNNLS*, 2009.

[7] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *CCBR*, 2018.

[8] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv:1801.07698*, 2018.

[9] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. *arXiv:1711.06753*, 2017.

[10] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

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

[12] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[13] 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.

[14] Saihui Hou, Xu Liu, and Zilei Wang. Dualnet: Learn complementary features for image recognition. In *ICCV*, 2017.

[15] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[16] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv:1712.05055*, 2017.

[17] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[18] Shengcai Liao, Zhen Lei, Dong Yi, and Stan Z Li. A benchmark study of large-scale unconstrained face recognition. In *ICB*, 2014.

[19] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *PAMI*, 2016.

[20] 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.

[21] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *NeurIPS*, 2017.

[22] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Virtual adversarial training for semi-supervised text classification. 2016.

[23] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012.

[24] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *CVPRW*, 2017.

[25] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *CVPR*, 2017.

[26] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014.

[27] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, 2015.

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[29] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

[30] 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 *WACV*, 2016.

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

[32] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv:1406.2080*, 2014.

[33] 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.

[34] 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.

[35] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*.

[36] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *ECCV*, 2018.

[37] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *SPL*, 2018.

[38] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. *arXiv:1704.06904*, 2017.

[39] Mei Wang, Weihong Deng, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Racial faces in-the-wild: Reducing racial bias by deep unsupervised domain adaptation. *arXiv:1812.00194*, 2018.

[40] Xiaobo Wang, Shuo Wang, Shifeng Zhang, Tianyu Fu, Hailin Shi, and Tao Mei. Support vector guided softmax loss for face recognition. *arXiv:1812.11317*, 2018.

[41] Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z Li. Ensemble soft-margin softmax loss for image classification. *arXiv:1805.03922*, 2018.

[42] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, 2018.

[43] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *TIFS*, 2018.

[44] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Z Li. Large scale similarity learning using similar pairs for person verification. In *AAAI*, 2016.

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

[46] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement benefit co-teaching? *arXiv:1901.04215*, 2019.

[47] Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *CVPR*, 2018.

[48] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. Faceboxes: A cpu real-time face detector with high accuracy. In *IJCB*, 2017.

[49] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.

[50] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying crosspose face recognition in unconstrained environments. *Tech. Rep*, 2018.

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