

# Meta-Mining Discriminative Samples for Kinship Verification

Wanhua Li<sup>1,2</sup>, Shiwei Wang<sup>3</sup>, Jiwen Lu<sup>1,2,\*</sup>, Jianjiang Feng<sup>1,2</sup>, Jie Zhou<sup>1,2</sup>

<sup>1</sup>Department of Automation, Tsinghua University, China

<sup>2</sup>Beijing National Research Center for Information Science and Technology, China

<sup>3</sup>School of Modern Post, Beijing University of Posts and Telecommunications

li-wh17@mails.tsinghua.edu.cn      2018110842@bupt.edu.cn

{lujiwen, jfeng, jzhou}@tsinghua.edu.cn

## Abstract

Kinship verification aims to find out whether there is a kin relation for a given pair of facial images. Kinship verification databases are born with unbalanced data. For a database with  $N$  positive kinship pairs, we naturally obtain  $N(N - 1)$  negative pairs. How to fully utilize the limited positive pairs and mine discriminative information from sufficient negative samples for kinship verification remains an open issue. To address this problem, we propose a Discriminative Sample Meta-Mining (DSMM) approach in this paper. Unlike existing methods that usually construct a balanced dataset with fixed negative pairs, we propose to utilize all possible pairs and automatically learn discriminative information from data. Specifically, we sample an unbalanced train batch and a balanced meta-train batch for each iteration. Then we learn a meta-miner with the meta-gradient on the balanced meta-train batch. In the end, the samples in the unbalanced train batch are re-weighted by the learned meta-miner to optimize the kinship models. Experimental results on the widely used KinFaceW-I, KinFaceW-II, TSKinFace, and Cornell Kinship datasets demonstrate the effectiveness of the proposed approach.

## 1. Introduction

Facial appearance conveys valuable information, such as identity [7, 23], age [19, 46], gender [17, 33], emotions [16, 44], social relation [18, 54], and so on. Recently, a variety of efforts [15, 21, 25, 32, 45] have been devoted to kinship verification, which predicts the existence of kinship for a given pair of facial images. As an emerging task, it has many potential applications including missing children search [25], intelligent family album organization [15, 29], and social media analysis [49, 58].

Existing kinship databases organize data in terms of pos-

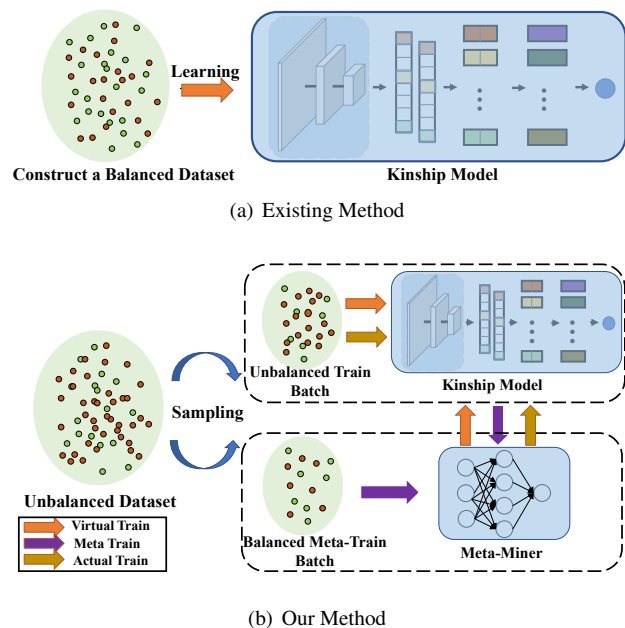


Figure 1. The key idea of our method. (a) The existing method constructs a fixed database by sampling the equal size of negative samples. Then a kinship model is learned with the balanced database. (b) Our method utilizes all possible pairs and does not discard any negative samples. Specifically, we sample an unbalanced train batch and a balanced meta-train batch for each iteration. A meta-miner is introduced to mine the samples in the training batch. Our method alternately optimizes the kinship model and the proposed meta-miner via a meta-learning framework.

itive samples. For example, KinFaceW-I and KinFaceW-II [25] datasets collect parent-child image pairs from the Internet search. Assuming that a kinship database contains  $N$  positive kinship pairs, the negative samples are generated by combining all unrelated parent-child image pairs. Therefore, we obtain  $N(N - 1)$  negative samples, which is far more than the number of positive pairs. Existing methods usually randomly sample fixed  $N$  negative samples to construct a balanced database. However, this strategy simply

\* Corresponding author

ignores the remaining  $N(N - 2)$  negative samples leading to overfitting. Besides, the real decision boundary cannot be well learned with randomly selected  $N$  negative samples. One simple strategy to address this issue is to sample a balanced batch from positive samples and all possible negative pairs separately. Nevertheless, most negative pairs are easy samples and they contribute little to the network training.

In this work, we investigate how to mine discriminative information from limited positive pairs and sufficient negative samples and propose a Discriminative Sample Meta-Mining (DSMM) strategy via a meta-learning framework. Specifically We first randomly sample an unbalanced train batch from all possible pairs with a positive to negative ratio of  $1 : C$ , where  $C > 1$ . Then we aim to mine the discriminative samples by re-weighting these samples in the training batch. Instead of manually selecting fixed weighting functions, we consider automatically learning the weighting functions from data. Encouraged by the recent success of meta-learning [9, 41, 43], we introduce a meta-miner network, which predicts the weight for each sample and is optimized with the meta-gradient. Concretely, another balanced meta-train batch is sampled from all possible pairs. Then we optimize the kinship model and meta-miner network alternately with these two batches. After one-step-forward optimization for the kinship model on the unbalanced train batch, we train the meta-miner network on the balanced meta-train batch with one-step-forward parameters of the kinship model. In the end, we perform the real optimization for the kinship model with the sample weights generated by the updated meta-miner network. Figure 1 shows the key idea of our method. The contributions of this paper are summarized as follows:

- To the best of our knowledge, the proposed discriminative sample meta-mining approach is the first attempt to fully utilize the unbalanced data of kinship databases via a meta-learning framework.
- Our method proposes to simultaneously sample an unbalanced train batch and a balanced meta-train batch, and then perform sample mining on the training batch with the guidance of a balanced meta-train batch.
- Extensive experiments on four widely used kinship databases illustrate that the proposed method achieves state-of-the-art results.

## 2. Related Work

**Kinship Verification:** Over the past decade, a variety of approaches [8, 21, 26, 48] have been proposed for facial kinship verification. We can categorize them into three groups [20]: hand-crafted methods, metric-learning based methods, and deep learning-based methods. In traditional exploration, hand-crafted methods have been widely used, such

as histogram of the gradient [8, 40, 57], Gabor gradient orientation pyramid [58], self-similarity [14], and so on. However, hand-crafted methods usually suffer from the limited performance. Distance metric based approach has become the most popular and successful method due to the superior performance in kinship verification [24, 26]. For example, Liang *et al.* [21] presented a weighted graph embedding-based metric learning method, which jointly learned multiple metrics by constructing an intrinsic graph. Recent years have witnessed the remarkable success of deep learning in computer vision, such as object detection [3, 28], and face recognition [2, 13]. Some researchers [20, 52] applied deep learning technologies into kinship verification. For example, Zhang *et al.* [52] extracted facial key-points features with CNNs. However, all these methods only use a fixed database, where only a small proportion of the negative samples are used. By contrast, we aim to mine the discriminative samples from all possible pairs.

**Class Imbalance:** Most efforts to address the class imbalance issue can be grouped into two categories [56]: re-sampling and re-weighting. The re-sampling strategy usually constructs a balanced batch by adding samples from minor classes [37] or under-sampling the dominant classes [11]. The re-weighting strategy assigns different weights for samples from different classes in the loss function [42]. Many advanced class imbalance learning methods [5, 22] have been proposed in recent years. Lin *et al.* [22] proposed focal loss to deal with the extreme imbalance between foreground and background classes. The effective number of samples was introduced in [5] and a class-balanced loss was further proposed with a re-weighting scheme. Zhou *et al.* [56] proposed a Bilateral-Branch Network for long-tailed visual recognition, which considers representation learning and classifier learning simultaneously.

**Meta-Learning:** Meta-learning aims to develop a machine learning system that is used to assist in the optimization of other machine learning models [9, 31, 38, 43]. Some researchers have proposed optimization based meta-learning approaches [1, 4, 9, 30]. Andrychowicz *et al.* [1] developed an algorithm with LSTMs to replace the hand-designed optimization algorithms. A variety of metric-based methods [39, 43] have been proposed, especially for the area of one/few shot learning. Sung *et al.* [41] utilized a relation network to learn an embedding and a deep distance metric for comparing two samples. There are also studies [27, 34] focusing on memory-based approaches. Santoro *et al.* [34] proposed a memory-augmented neural network to rapidly encode new data and make accurate predictions with the assimilated data. Encouraged by the success of meta-learning, we design a meta-miner and utilize it to guide the training process of our kinship model by fully exploiting all possible samples in the database. Although meta-learning has been successfully applied to face recognition [10], our

---

**Algorithm 1:** Discriminative Sample Meta-Mining.

---

**Input:** Iteration number  $T$ .**Output:** Model parameters  $\theta^T$  and  $\phi^T$ .

- 1: Initialize model parameters  $\theta^0$  and  $\phi^0$ .
  - 2: **for**  $t = 0, 1, \dots, T - 1$  **do**
  - 3:   Sample an unbalanced train batch  $(\mathcal{S}^{trn,p}, \mathcal{S}^{trn,n})$ .
  - 4:   Sample a balanced batch  $(\mathcal{S}^{meta,p}, \mathcal{S}^{meta,n})$ .
  - 5:   Perform the virtual training by (4) - (5)
  - 6:   Update  $\phi^{t+1}$  on the meta-train batch via (6) - (7).
  - 7:   Normalize sample weights with (8).
  - 8:   Update  $\theta^{t+1}$  by (9) - (11).
  - 9: **end for**
  - 10: **return**  $\theta^T, \phi^T$ .
- 

method is still significantly different from it in terms of motivation, framework, optimization strategy, *etc.*

### 3. Proposed Approach

In this section, we first introduce the unbalanced data issue for kinship verification. Then, we detail how the proposed discriminative sample meta-mining approach exploits limited positive samples and sufficient negative samples. The meta-learning framework is further presented. Lastly, we introduce the network design of our kinship relation model and the proposed meta-miner.

#### 3.1. Unbalanced Kinship Data

Kinship verification organizes each sample as a facial image pair  $(\mathbf{x}, \mathbf{y})$ , where  $\mathbf{x}$  denotes the parent image and  $\mathbf{y}$  is the child image. Existing kinship verification databases are usually constructed by collecting positive kinship pairs since the negative pairs can be generated by shuffling the positive pairs. Assuming that there are a total of  $N$  positive pairs in a dataset, then we naturally obtain  $N(N - 1)$  negative samples. Therefore, the size of the negative samples is much larger than that of positive samples.

How to learn a kinship model with the unbalanced data remains an open issue. Most existing methods randomly select  $N$  negative samples to form a balanced dataset and discard the remaining  $N(N - 2)$  negative samples. Formally, we denote the constructed positive data and negative data by  $\mathcal{D}^{trn,p}$  and  $\mathcal{D}^{trn,n}$ , respectively. Let  $f(\mathbf{x}, \mathbf{y}; \theta)$  denote the kinship model which is parameterized by  $\theta$ . Then the model is trained with binary cross entropy loss on the balanced dataset:

$$\begin{aligned} \mathcal{L} = & -\frac{1}{2N} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}^{trn,p}} \log(f(\mathbf{x}, \mathbf{y}; \theta)) \\ & -\frac{1}{2N} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}^{trn,n}} \log(1 - f(\mathbf{x}, \mathbf{y}; \theta)). \end{aligned} \quad (1)$$

However, these methods cannot fully utilize the negative

data leading to inaccurate decision boundaries. One strategy to address this issue is to sample a balanced mini-batch for each iteration from all possible pairs. Formally, a positive batch  $\mathcal{B}^{trn,p}$  with  $m$  examples and a negative batch  $\mathcal{B}^{trn,n}$  with  $m$  pairs are sampled. Then we train the kinship model as follows:

$$\begin{aligned} \mathcal{L} = & -\frac{1}{2m} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{B}^{trn,p}} \log(f(\mathbf{x}, \mathbf{y}; \theta)) \\ & -\frac{1}{2m} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{B}^{trn,n}} \log(1 - f(\mathbf{x}, \mathbf{y}; \theta)). \end{aligned} \quad (2)$$

This strategy can use all possible pairs with a balanced batch. Nevertheless, most negative pairs are easy examples, which contribute little to the network training. In this paper, we aim to mine the discriminative information from the limited positive samples and sufficient negative facial image pairs simultaneously.

#### 3.2. Discriminative Sample Meta-Mining

To address the above issue, we propose a discriminative sample meta-mining approach with a meta-learning framework. To fully exploit the potential of kinship data, we perform sampling from all possible pairs to construct a batch. While the kinship data is unbalanced and most negative samples are easy ones, we manually construct an unbalanced train batch with a positive to negative ratio of  $1 : C (C > 1)$ . We then mine the discriminative samples from the unbalanced batch. We employ a sample re-weighting strategy to mining the discriminative information, which re-weights the loss value for each sample. One can assign a constant value or choose a hand-designed weighting function to re-weight the samples in the unbalanced train batch. Instead, we aim to learn a weighting function directly from data with a meta-learning framework. Specifically, we introduce a meta-miner to generate the weights. We implement the meta-miner with a neural network which is denoted as  $g(\phi)$  parameterized by  $\phi$ . The positive set and negative set of the sampled train batch are denoted as  $\mathcal{S}^{trn,p}$  and  $\mathcal{S}^{trn,n}$ , respectively. Then the training objective for a given fixed meta-miner network  $g(\phi)$  is defined as:

$$\begin{aligned} \mathcal{L} = & \frac{-1}{m(1+C)} \left( \sum_{\mathbf{s} \in \mathcal{S}^{trn,p}} g(\mathbf{s}; \phi) \log(f(\mathbf{s}; \theta)) \right. \\ & \left. + \sum_{\mathbf{s} \in \mathcal{S}^{trn,n}} g(\mathbf{s}; \phi) \log(1 - f(\mathbf{s}; \theta)) \right), \end{aligned} \quad (3)$$

where  $m$  is the size of positive samples in the training batch and  $\mathbf{s}$  represents a pair of positive images.

We then optimize the parameters  $\theta$  and  $\phi$  with an online strategy. The framework is depicted in Figure 1(b).

**Virtual Training the Kinship Model:** Assuming that we have obtained the parameters  $\theta^t$  and  $\phi^t$  at  $t$ -th time step,

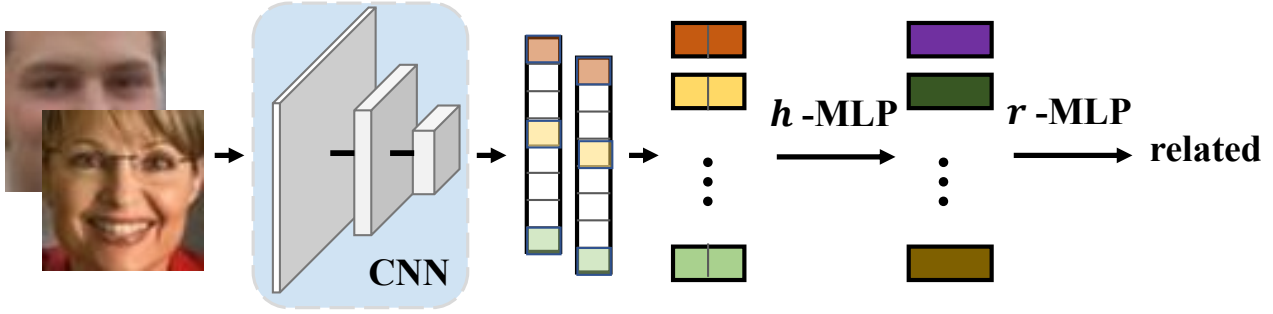


Figure 2. The overall framework of our kinship relation model. For paired facial images, we use a CNN to extract deep features for them. Then we explicitly model the relations of these two features with a relation module.

now we perform a virtual update for  $\theta^t$  to obtain  $\hat{\theta}^{t+1}$ . The unbalanced train batch is used for updating. Therefore, the loss function for this virtual training step is expressed as:

$$\mathcal{L}_{v.trn}(\phi^t) = \frac{-1}{m(1+C)} \left( \sum_{\mathbf{s} \in \mathcal{S}^{trn,p}} g(\mathbf{s}; \phi^t) \log(f(\mathbf{s}; \theta^t)) + \sum_{\mathbf{s} \in \mathcal{S}^{trn,n}} g(\mathbf{s}; \phi^t) \log(1 - f(\mathbf{s}; \theta^t)) \right). \quad (4)$$

Since different  $\phi^t$  leads to different virtual train loss values, the loss in (4) is essentially a function of  $\phi^t$ . With the defined loss, we can update the kinship model with SGD:

$$\hat{\theta}^{t+1}(\phi^t) = \theta^t - \alpha \nabla_{\theta^t} \mathcal{L}_{v.trn}(\phi^t), \quad (5)$$

where  $\alpha$  is the step size. It is worth noting that the virtual training step does not actually update the parameter  $\theta^t$  of the kinship model, which is used to calculate the second-order gradients in the next step.

**Training the Meta-Miner Network:** Having obtained the  $\hat{\theta}^t$ , we then train the meta-miner network to obtain  $\phi^{t+1}$ . Specifically, we optimize the parameter of the meta-miner network using the meta-learning idea [1, 9]. Another balanced meta train batch is sampled, which is used to guide the training process of the meta-miner network. The motivation of this step is that we expect that the obtained parameter  $\hat{\theta}^{t+1}$  in the last step can achieve better performance on the balanced meta-train batch. Formally, we define the positive set and the negative set of the balanced meta-train batch as  $\mathcal{S}^{meta,p}$  and  $\mathcal{S}^{meta,n}$ , respectively. Then the meta-loss is given by:

$$\mathcal{L}_{meta}(\phi^t) = -\frac{1}{2m} \sum_{\mathbf{s} \in \mathcal{S}^{meta,p}} \log(f(\mathbf{s}; \hat{\theta}^{t+1}(\phi^t))) - \frac{1}{2m} \sum_{\mathbf{s} \in \mathcal{S}^{meta,n}} \log(1 - f(\mathbf{s}; \hat{\theta}^{t+1}(\phi^t))), \quad (6)$$

where we use  $m$  positive samples and  $m$  negative samples to form a balanced batch. Then we update the parameter  $\phi^t$

via gradient descent method with a learning rate of  $\beta$ :

$$\phi^{t+1} = \phi^t - \beta \nabla_{\phi^t} \mathcal{L}_{meta}(\phi^t). \quad (7)$$

With the updated the parameter  $\phi^{t+1}$ , we can generate the real weights for the samples in the unbalanced train batch. For each sample  $\mathbf{s} \in \mathcal{S}^{trn}$ , we obtain the weight  $g(\mathbf{s}; \phi^{t+1})$ . Before they are used in the next step, we first normalize them within the train batch:

$$\tilde{w}_{\mathbf{s}} = \frac{g(\mathbf{s}; \phi^{t+1})}{\sum_{\mathbf{q} \in \mathcal{S}^{trn,p}} g(\mathbf{q}; \phi^{t+1}) + \sum_{\mathbf{q} \in \mathcal{S}^{trn,n}} g(\mathbf{q}; \phi^{t+1})}. \quad (8)$$

**Actual Training the Kinship Model:** With the obtained weights in the second step, we can define the weighed loss function as follows:

$$\mathcal{L}_{trn}(\theta^t) = \frac{-1}{m(1+C)} \left( \sum_{\mathbf{s} \in \mathcal{S}^{trn,p}} \tilde{w}_{\mathbf{s}} \log(f(\mathbf{s}; \theta^t)) + \sum_{\mathbf{s} \in \mathcal{S}^{trn,n}} \tilde{w}_{\mathbf{s}} \log(1 - f(\mathbf{s}; \theta^t)) \right). \quad (9)$$

Then we perform the actual network optimization for kinship model with gradient descent:

$$\theta^{t+1} = \theta^t - \gamma \nabla_{\theta^t} \mathcal{L}_{trn}(\theta^t), \quad (10)$$

where  $\gamma$  is the learning rate.

In this way, the parameters  $\theta$  and  $\phi$  are alternately optimized via meta-learning. We show the above framework in Algorithm 1.

### 3.3. Implementation Details

The proposed DSMM method involves two networks: kinship model  $f(\theta)$  and meta-miner network  $g(\phi)$ . Here we detail the design of these two models.

For the kinship model, we first use a ResNet-18 to extract image features for each facial image. Formally, for a paired sample  $(\mathbf{x}, \mathbf{y})$ , we obtain the deep embeddings

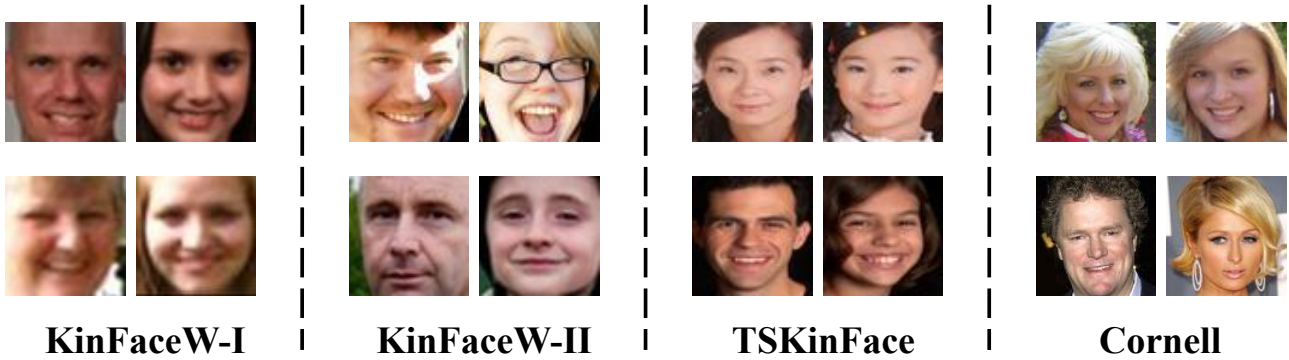


Figure 3. Some examples of four widely used databases. Each row shows a paired sample, where the first image represents the parent and the second image belongs to the child.

$(e_x, e_y) \in (\mathbb{R}^D, \mathbb{R}^D)$ , where  $D$  represents the feature dimension. Inspired by the recent success of relation networks [35], we consider modeling the relations of these two features to infer their kinship. We explicitly model the per-dimension relations of two extracted features and formulate the kinship relation model as follows:

$$f(\theta) = r\left(\left\| \begin{array}{c} D \\ h(e_x^i, e_y^i) \end{array} \right\| \right), \quad (11)$$

where  $\|$  represents concatenation,  $e_x^i$  and  $e_y^i$  denote the  $i$ -th element in the embeddings  $e_x$  and  $e_y$  respectively. Both  $r(\cdot)$  and  $h(\cdot)$  are implemented with MLPs. We depict the kinship relation model in Figure 2.

For the meta-miner network  $g(\phi)$ , we implement it with a three-layer MLP. Instead of taking the whole samples as the input, we only use the loss information to generate the sample weights. Specifically, the meta-miner network takes sample labels, predictions of our kinship model, and corresponding train losses as inputs. Therefore, the input dimension of  $g(\phi)$  is 3. Besides, we set the neural number of the hidden layer as 512, which achieves the best performance.

## 4. Experiments

To evaluate the effectiveness of the proposed discriminative sample meta-mining approach, we conducted extensive experiments on four widely used kinship datasets

### 4.1. Datasets

**KinFaceW-I Dataset [26]:** This dataset contains 533 pairs of facial images of persons with a kin relation. Four different kin relations are considered in the dataset: father and daughter (F-D) with 134 pairs, father and son (F-S) with 156 pairs, mother and daughter (M-D) with 127 pairs, mother and son (M-S) with 116 pairs. Each sample is composed of one parent face image and one child face image. The images are marked with the positions of eyes and are aligned to a size of  $64 \times 64$  based on the landmarks. We follow

Table 1. Mean accuracy (%) of different fusion methods. EF means early fusion, LF represents late fusion.

| Dataset     | CNNP | EF   | LF   | Relation | DSMM        |
|-------------|------|------|------|----------|-------------|
| KinFaceW-I  | 77.5 | 75.7 | 72.3 | 78.1     | <b>82.4</b> |
| KinFaceW-II | 88.4 | 87.9 | 81.8 | 90.4     | <b>93.0</b> |

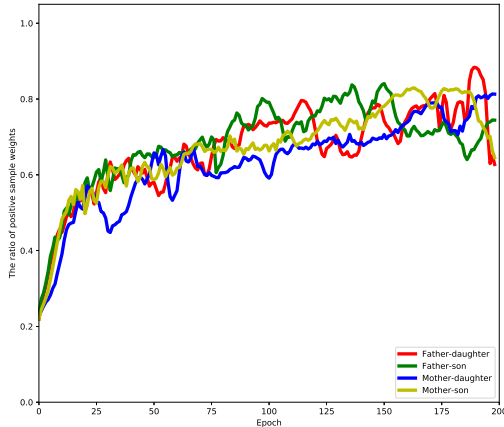
the standard protocol [26] in this database which adopts the five-fold cross-validation.

**KinFaceW-II Dataset [26]:** This dataset consists of 1000 pairs of facial images of individuals with a kin relation. This database also considers four common kin relations: father and daughter (F-D), father and son (F-S), mother and daughter (M-D), mother and son (M-S). Different from the KinFaceW-I database, the positive pairs in this dataset are taken from the same photo. Following the standard protocol [26], we perform five-fold cross-validation.

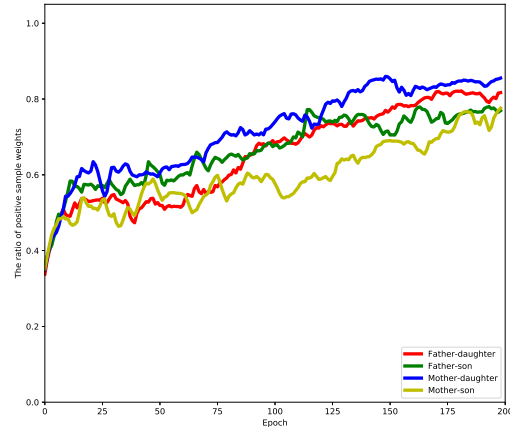
**TSKinFace Dataset [47]:** TSKinFace dataset has 2589 face images in total. Unlike the traditional kinship database which investigates bi-subject relations, it is constructed to model tri-subject relations. Two kin relations are considered in this database: Father-Mother-Son (FM-S) with 285 groups and Father-Mother-Daughter (FM-D) with 274 groups. Following [24], we re-organize the data into the four kinds of bi-subject relations as other datasets. Then we obtain 502 pairs of father and daughter (F-D) relations, 513 pairs of father and son (F-S) relations, 502 pairs of mother and daughter (M-D) relations, 513 pairs of mother and son (M-S) relations. Five-fold cross-validation is adopted.

**Cornell KinFace Dataset [8]:** Cornell KinFace database is a small kinship dataset with only 150 pairs of face images. All images are collected from the internet including four family relations. There are 22% father and daughter (F-D) relations, 40% father and son (F-S) relations, 25% mother and daughter (M-D) relations, and 13% mother and son (M-S) relations in this dataset. We follow the standard protocol [8] with five-fold cross-validation.

Some examples of these datasets are shown in Figure 3.



(a) KinFaceW-I



(b) KinFaceW-II

Figure 4. The ratio curves of positive sample weights on the unbalanced train batch changing with epochs on the KinFaceW-I database and KinFaceW-II database.

## 4.2. Experimental Settings

For all datasets, we performed data augmentation to alleviate the overfitting issue. First, we resized the image size to  $73 \times 73$ , and then randomly cropped it to a size of  $64 \times 64$ . Finally, we performed a random horizontal flipping with a probability of 50%. The positive sample batch size on the KinFaceW-II and TSKinFace datasets was set 16. The positive sample batch size on the KinFaceW-I and Cornell KinFace datasets was set to 8 since they have smaller data sizes. For the kinship relation model, we used a ResNet-18 network to extract image features, which was trained with binary cross-entropy loss. We optimized our network with Adam optimizer. We set the learning rates  $\alpha = 0.001$ ,  $\beta = 0.0001$ , and  $\gamma = 0.001$ . We decayed the learning rate  $\gamma$  by 0.1 at 100 and 150 epochs. We trained our model for 200 epochs. The meta-miner network was implemented with an MLP. The output dimension of  $g(\phi)$  was set to 1, which corresponds to the generated sample weight. A sigmoid function was employed to normalize the output.

## 4.3. Results and Analysis

In this subsection, we present the experimental results on four kinship databases and give a detailed analysis.

**Kinship Relation Model:** Note that the proposed kinship relation model as defined in (11) is novel to kinship verification. Since kinship verification deals with two facial images, one key design is how to fuse these two images. The closest method is CNNP [52], which uses a self-designed backbone and early fusion. We consider several different strategies and show the results in Table 1. Compared with CNNP, the EF (early fusion) solution only replaces the feature extractor model with ResNet-18, which

Table 2. Ablation study on the KinFaceW-I dataset.

| Method            | FD          | FS          | MD          | MS          | Mean        |
|-------------------|-------------|-------------|-------------|-------------|-------------|
| Balance Batch     | 76.2        | 79.5        | 87.9        | 79.8        | 80.9        |
| Unbalance Batch   | 75.8        | 80.1        | 88.2        | 79.3        | 80.9        |
| Balance + Focal   | 69.8        | 72.1        | 83.5        | 72.8        | 74.6        |
| Unbalance + Focal | 71.3        | 73.7        | 83.0        | 75.4        | 75.8        |
| <b>DSMM</b>       | <b>76.7</b> | <b>81.7</b> | <b>89.0</b> | <b>82.3</b> | <b>82.4</b> |

Table 3. Ablation study on the KinFaceW-II dataset.

| Method            | FD          | FS          | MD          | MS          | Mean        |
|-------------------|-------------|-------------|-------------|-------------|-------------|
| Balance Batch     | 89.2        | 92.0        | 94.2        | 92.0        | 91.8        |
| Unbalance Batch   | 89.2        | 92.0        | 94.8        | <b>94.0</b> | 92.5        |
| Balance + Focal   | 78.6        | 79.0        | 83.0        | 81.4        | 80.5        |
| Unbalance + Focal | 79.4        | 81.6        | 86.4        | 84.4        | 83.0        |
| <b>DSMM</b>       | <b>89.8</b> | <b>92.6</b> | <b>95.8</b> | 93.6        | <b>93.0</b> |

achieves slightly worse performance due to a simpler network. For simplicity, the ResNet-18 is adopted as the backbone for all other strategies. The LF (late fusion) solution fuses two images in the last convolution layer instead of the input layer, leading to terrible results. As a late fusion strategy, our proposed relation module greatly improves performance, which shows the effectiveness of our kinship relation model. With the proposed kinship relation model, our DSMM further improves the accuracy by a large margin.

**Ablation Study:** To validate the effectiveness of our method, we consider several different strategies to handle the unbalanced data issue. The first strategy is named Balance Batch, which is a re-sampling strategy. It randomly selects a balanced batch from all possible pairs and uses the

Table 4. Results with different  $C$  on the KinFaceW-I dataset.

| $C$ | FD          | FS          | MD          | MS          | Mean        |
|-----|-------------|-------------|-------------|-------------|-------------|
| 1   | 76.6        | 80.1        | 88.6        | 81.9        | 81.8        |
| 2   | 75.3        | 81.1        | 86.9        | 78.9        | 80.6        |
| 4   | <b>76.7</b> | <b>81.7</b> | <b>89.0</b> | <b>82.3</b> | <b>82.4</b> |
| 8   | 75.5        | 77.9        | 87.8        | 78.0        | 79.8        |

Table 5. Results with different  $C$  on the KinFaceW-II dataset.

| $C$ | FD          | FS          | MD          | MS          | Mean        |
|-----|-------------|-------------|-------------|-------------|-------------|
| 1   | 89.0        | 91.8        | <b>96.0</b> | 93.4        | 92.5        |
| 2   | <b>89.8</b> | <b>92.6</b> | 95.8        | <b>93.6</b> | <b>93.0</b> |
| 3   | 87.6        | 91.4        | 94.0        | 92.0        | 91.3        |
| 4   | 87.0        | 91.0        | 92.8        | 91.2        | 90.5        |

sampled balance batch to train our kinship relation model. The second method sample the same unbalanced train batch as our method does. To address the unbalanced issue, we directly set the sample weight for all negative samples as  $1/C$ , which is a re-weighting strategy. We also consider applying focal loss [22] for the sampled balanced batches or unbalanced batches to mine discriminative samples.

We conducted experiments with these different strategies on the KinFaceW-I and KinFaceW-II datasets. The results are shown in Table 2 and Table 3, respectively. We observe that Unbalance Batch usually gives better performance. The reason is that an unbalanced batch contains more negative samples so that they can be mined later. Since negative samples usually are the easy ones, sampling more negative samples are more useful. Both Balance Batch and Unbalance Batch can improve performance compared with the kinship relation model baseline as shown in Table 1. However, the Focal loss significantly hurts performance. Due to the limited data size, we observed significant over-fitting (the accuracy on the training set is close to 100%). The use of Focus Loss makes only a few hard samples dominate the gradient, resulting in a highly biased model with poor generalization. By contrast, our method introduces the balanced meta-train batch to guarantee the generalization ability of our model. Besides, our method can dynamically fit a wide range of weighting functions rather than just a certain family of functions. In the end, we see that our method achieves the best performance on both datasets, which demonstrates the effectiveness of our proposed approach.

To better understand the learning process of our meta-miner, we show the ratio of positive sample weights on the unbalanced train batch, which is defined as the total of positive weights divided by the total of all sample weights per batch. The curves are depicted in Figure 4. We observe that the ratio of positive sample weights increases with the number of epochs, which shows that the contribution of positive samples for training is increasing during training. Such observation verifies the necessity of sampling an unbalanced

Table 6. Comparison of verification accuracy of DSMM with state-of-the-art on the KinFaceW-I dataset.

| Method      | FD          | FS          | MD          | MS          | Mean        |
|-------------|-------------|-------------|-------------|-------------|-------------|
| IML [50]    | 67.5        | 70.5        | 72.0        | 65.5        | 68.9        |
| MNRML [26]  | 66.5        | 72.5        | 72.0        | 66.2        | 69.3        |
| MPDFL [51]  | 73.5        | 67.5        | 66.1        | 73.1        | 70.1        |
| DMML [50]   | 69.5        | 74.5        | 75.5        | 69.5        | 72.3        |
| GA [6]      | 76.4        | 72.5        | 71.9        | 77.3        | 74.5        |
| WGEML [21]  | <b>78.5</b> | 73.9        | 80.6        | 81.9        | 78.7        |
| WLDA [36]   | 76.6        | 71.2        | 77.7        | 76.4        | 75.5        |
| MMFA [36]   | 77.9        | 72.0        | 77.2        | 75.2        | 75.6        |
| CNNP [52]   | 71.8        | 76.1        | 84.1        | 78.0        | 77.5        |
| <b>DSMM</b> | 76.7        | <b>81.7</b> | <b>89.0</b> | <b>82.3</b> | <b>82.4</b> |

Table 7. Comparison of verification accuracy of DSMM with state-of-the-art on the KinFaceW-II dataset.

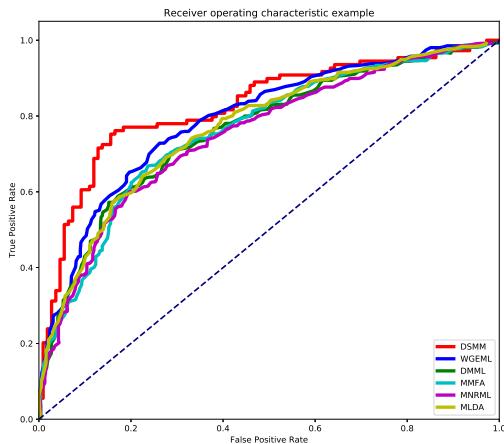
| Method                 | FD          | FS          | MD          | MS          | Mean        |
|------------------------|-------------|-------------|-------------|-------------|-------------|
| IML [50]               | 74.0        | 74.5        | 78.5        | 76.5        | 75.9        |
| MNRML [26]             | 74.3        | 76.9        | 77.6        | 77.4        | 76.6        |
| MPDFL [51]             | 77.3        | 74.7        | 77.8        | 78.0        | 77.0        |
| DMML [50]              | 76.5        | 78.5        | 79.5        | 78.5        | 78.3        |
| LM <sup>3</sup> L [12] | 82.4        | 74.2        | 79.6        | 78.7        | 78.7        |
| GA( [6])               | 83.9        | 76.7        | 83.4        | 84.8        | 82.2        |
| WGEML [21]             | 88.6        | 77.4        | 83.4        | 81.6        | 82.8        |
| WLDA [36]              | 86.6        | 74.4        | 81.0        | 78.8        | 80.2        |
| MMFA [36]              | 85.6        | 73.2        | 80.4        | 77.2        | 79.1        |
| CNNP [52]              | 81.9        | 89.4        | 92.4        | 89.9        | 88.4        |
| <b>DSMM</b>            | <b>89.8</b> | <b>92.6</b> | <b>95.8</b> | <b>93.6</b> | <b>93.0</b> |

train batch and the effectiveness of our meta-miner.

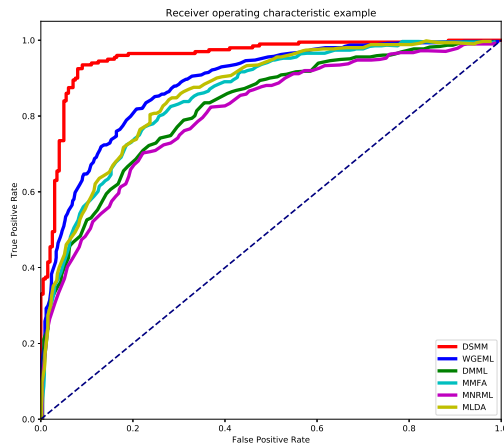
**Parameters Analysis:** We conducted experiments on the KinFaceW-I and KinFaceW-II datasets to choose the best value for parameter  $C$ . The results are shown in Table 4 and Table 5, respectively. On the KinFaceW-I data set, we set the values of  $C$  as 1, 2, 4, and 8, respectively. The experimental results show that our method achieves the best performance when  $C = 4$ . On the KinFaceW-II data set, we set the values of  $C$  as 1, 2, 3, and 4, respectively. The experimental results show that our method achieves the best performance when  $C = 2$ . The above setting is used for our following experiments on the corresponding datasets.

**Comparisons with State-of-the-Art:** We compared the proposed DSMM with existing kinship verification methods on the KinFaceW-I database, the KinFaceW-II database, the TSKinFace database, and the Cornell Kinship datasets.

Table 6 and Table 7 show the experimental results on the KinFaceW-I and KinFaceW-II databases respectively. Bolded numbers represent the best results. On the KinFaceW-I data set, the average verification rate of our proposed method is 82.4%, which outperforms the previous best method WGEML [21] by 3.7%. The proposed



(a) KinFaceW-I



(b) KinFaceW-II

Figure 5. The comparisons of ROC curves with other methods on the KinFaceW-I dataset and the KinFaceW-II dataset.

Table 8. Comparison of verification accuracy of DSMM with state-of-the-art on the TSKinFace dataset.

| Method      | FD          | FS          | MD          | MS          | Mean        |
|-------------|-------------|-------------|-------------|-------------|-------------|
| BSIF [47]   | 81.4        | 81.5        | 79.9        | 82.0        | 81.2        |
| DDMML [24]  | 86.6        | 82.5        | 83.2        | 84.3        | 84.2        |
| GMP [53]    | 88.5        | 87.0        | 87.9        | 87.8        | 87.8        |
| MKSM [55]   | 82.0        | 81.4        | 82.3        | 81.9        | 81.9        |
| <b>DSMM</b> | <b>91.4</b> | <b>92.4</b> | <b>93.9</b> | <b>93.2</b> | <b>92.7</b> |

Table 9. Comparison of verification accuracy of DSMM with state-of-the-art on the Cornell dataset.

| Method      | FD          | FS          | MD          | MS          | Mean        |
|-------------|-------------|-------------|-------------|-------------|-------------|
| PSM [8]     | 72.9        | 54.6        | 73.8        | 61.3        | 70.7        |
| MNRML [25]  | 74.5        | 68.8        | 77.2        | 65.8        | 71.6        |
| MPDFL [51]  | 74.8        | 69.1        | 77.5        | 66.1        | 71.9        |
| DMML [50]   | 76.0        | 70.5        | 77.5        | 71.0        | 73.5        |
| MKSM [55]   | 80.5        | 80.6        | 79.5        | 86.2        | 81.7        |
| <b>DSMM</b> | <b>88.0</b> | <b>87.0</b> | <b>94.0</b> | <b>93.0</b> | <b>90.5</b> |

method achieves an average verification rate of 93.0% on the KinFaceW-II data set, which outperforms the previous best approach CNNP [52] by 4.6%.

We also present the comparisons of our DSMM method and other methods on the KinFaceW-I and KinFaceW-II datasets in terms of ROC curves. As can be seen in Figure 5, the curve of the DSMM method is higher than other methods, which demonstrates the superiority of the proposed discriminative sample meta-mining approach. Note that the KVRL [15] achieves higher scores due to the multi-patches input and outside data training. For a fair comparison, we do not include it in Table 6 and Table 7.

Table 8 shows the experimental results on the TSKin-

Face database while Table 9 lists the experimental results on the Cornell KinFace database. We observe that our method attains an average verification rate of 92.7% on the TSKin-Face dataset, outperforming the GMP [53] method by 4.9%. The proposed discriminative sample meta-mining method obtains an average verification rate of 90.5% on the Cornell dataset, which outperforms the previous best-performing MKSM [55] method by 8.8%.

## 5. Conclusion

In this paper, we have presented the discriminative sample meta-mining approach for kinship verification. Unlike the existing methods that usually construct a balanced database with fixed negative samples, our method utilizes all possible pairs and aims to exploit the discriminative information from the limited positive samples and sufficient negative pairs. For each iteration, an unbalanced train batch and a balanced meta-train batch are sampled. The balanced meta-train batch is used to guide the training process of our meta-miner via a meta-learning framework. Then our kinship model is trained with the discriminative samples on the training batch, which are weighted by the meta-miner. Extensive experiment results on four widely used kinship databases demonstrate that our method significantly outperforms the state-of-the-art methods.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61822603, Grant U1813218, Grant U1713214, in part by Beijing Academy of Artificial Intelligence (BAAI), and in part by a grant from the Institute for Guo Qiang, Tsinghua University.



## References

- [1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, pages 3981–3989, 2016. [2](#), [4](#)
- [2] Dong Cao, Xiangyu Zhu, Xingyu Huang, Jianzhu Guo, and Zhen Lei. Domain balancing: Face recognition on long-tailed domains. In *CVPR*, pages 5671–5679, 2020. [2](#)
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. [2](#)
- [4] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando Freitas. Learning to learn without gradient descent by gradient descent. In *ICML*, pages 748–756, 2017. [2](#)
- [5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019. [2](#)
- [6] Afshin Dehghan, Enrique G Ortiz, Ruben Villegas, and Mubarak Shah. Who do i look like? determining parent-offspring resemblance via gated autoencoders. In *CVPR*, pages 1757–1764, 2014. [7](#)
- [7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. [1](#)
- [8] Ruogu Fang, Kevin D Tang, Noah Snavely, and Tsuhan Chen. Towards computational models of kinship verification. In *ICIP*, pages 1577–1580, 2010. [2](#), [5](#), [8](#)
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. [2](#), [4](#)
- [10] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *CVPR*, pages 6163–6172, 2020. [2](#)
- [11] Haibo He and Eduardo A Garcia. Learning from imbalanced data. *TKDE*, 21(9):1263–1284, 2009. [2](#)
- [12] Junlin Hu, Jiwen Lu, Junsong Yuan, and Yap-Peng Tan. Large margin multi-metric learning for face and kinship verification in the wild. In *ACCV*, pages 252–267, 2014. [7](#)
- [13] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *CVPR*, pages 5901–5910, 2020. [2](#)
- [14] Naman Kohli, Richa Singh, and Mayank Vatsa. Self-similarity representation of weber faces for kinship classification. In *BTAS*, pages 245–250, 2012. [2](#)
- [15] Naman Kohli, Mayank Vatsa, Richa Singh, Afzel Noore, and Angshul Majumdar. Hierarchical representation learning for kinship verification. *TIP*, 26(1):289–302, 2016. [1](#), [8](#)
- [16] Jean Kossaifi, Antoine Toisoul, Adrian Bulat, Yannis Panagakis, Timothy M Hospedales, and Maja Pantic. Factorized higher-order cnns with an application to spatio-temporal emotion estimation. In *CVPR*, pages 6060–6069, 2020. [1](#)
- [17] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *CVPRW*, pages 34–42, 2015. [1](#)
- [18] Wanhua Li, Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Graph-based social relation reasoning. In *ECCV*, pages 18–34, 2020. [1](#)
- [19] Wanhua Li, Jiwen Lu, Jianjiang Feng, Chunjing Xu, Jie Zhou, and Qi Tian. Bridgenet: A continuity-aware probabilistic network for age estimation. In *CVPR*, pages 1145–1154, 2019. [1](#)
- [20] Wanhua Li, Yingqiang Zhang, Kangchen Lv, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Graph-based kinship reasoning network. In *ICME*, pages 1–6, 2020. [2](#)
- [21] Jianqing Liang, Qinghua Hu, Chuangyin Dang, and Wangmeng Zuo. Weighted graph embedding-based metric learning for kinship verification. *TIP*, 28(3):1149–1162, 2018. [1](#), [2](#), [7](#)
- [22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. [2](#), [7](#)
- [23] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Spherefacer: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017. [1](#)
- [24] Jiwen Lu, Junlin Hu, and Yap-Peng Tan. Discriminative deep metric learning for face and kinship verification. *TIP*, 26(9):4269–4282, 2017. [2](#), [5](#), [8](#)
- [25] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *TPAMI*, 36(2):331–345, 2013. [1](#), [8](#)
- [26] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *TPAMI*, 36(2):331–345, 2014. [2](#), [5](#), [7](#)
- [27] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICML*, 2018. [2](#)
- [28] Wanli Peng, Hao Pan, He Liu, and Yi Sun. Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving. In *CVPR*, pages 13015–13024, 2020. [2](#)
- [29] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. Tri-subject kinship verification: Understanding the core of a family. *TMM*, 17(10):1855–1867, 2015. [1](#)
- [30] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. [2](#)
- [31] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018. [2](#)
- [32] Joseph P Robinson, Ming Shao, Yue Wu, Hongfu Liu, Timothy Gillis, and Yun Fu. Visual kinship recognition of families in the wild. *TPAMI*, 40(11):2624–2637, 2018. [1](#)
- [33] Pau Rodríguez, Guillem Cucurull, Josep M Gonfaus, F Xavier Roca, and Jordi González. Age and gender recognition in the wild with deep attention. *PR*, 72:563–571, 2017. [1](#)
- [34] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016. [2](#)
- [35] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, pages 4967–4976, 2017. [5](#)
- [36] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *CVPR*, pages 2160–2167, 2012. [7](#)

- [37] Li Shen, Zhouchen Lin, and Qingming Huang. Relay back-propagation for effective learning of deep convolutional neural networks. In *ECCV*, pages 467–482, 2016. 2
- [38] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NIPS*, pages 1919–1930, 2019. 2
- [39] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017. 2
- [40] Gowri Somanath and Chandra Kambhampettu. Can faces verify blood-relations? In *BTAS*, pages 105–112, 2012. 2
- [41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018. 2
- [42] Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *ICML*, 2000. 2
- [43] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016. 2
- [44] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. Suppressing uncertainties for large-scale facial expression recognition. In *CVPR*, pages 6897–6906, 2020. 1
- [45] Wei Wang, Shaodi You, Sezer Karaoglu, and Theo Gevers. Kinship identification through joint learning using kinship verification ensemble. In *ECCV*, 2020. 1
- [46] Xin Wen, Biying Li, Haiyun Guo, Zhiwei Liu, Guosheng Hu, Ming Tang, and Jinqiao Wang. Adaptive variance based label distribution learning for facial age estimation. In *ECCV*, 2020. 1
- [47] Xiaoting Wu, Elhocine Boutellaa, Miguel Bordallo López, Xiaoyi Feng, and Abdenour Hadid. On the usefulness of color for kinship verification from face images. In *EIFS*, pages 1–6, 2016. 5, 8
- [48] Siyu Xia, Ming Shao, Jiebo Luo, and Yun Fu. Understanding kin relationships in a photo. *TMM*, 14(4):1046–1056, 2012. 2
- [49] Zhe Xu, Ya Zhang, and Longbing Cao. Social image analysis from a non-iid perspective. *TMM*, 16(7):1986–1998, 2014. 1
- [50] Haibin Yan, Jiwen Lu, Weihong Deng, and Xiuzhuang Zhou. Discriminative multimetric learning for kinship verification. *TIFS*, 9(7):1169–1178, 2014. 7, 8
- [51] Haibin Yan, Jiwen Lu, and Xiuzhuang Zhou. Prototype-based discriminative feature learning for kinship verification. *TCYB*, 45(11):2535–2545, 2014. 7, 8
- [52] Kaihao Zhang, Yongzhen Huang, Chunfeng Song, Hong Wu, and Liang Wang. Kinship verification with deep convolutional neural networks. In *BMVC*, 2015. 2, 6, 7, 8
- [53] Ziming Zhang, Yuting Chen, and Venkatesh Saligrama. Group membership prediction. In *ICCV*, pages 3916–3924, 2015. 8
- [54] Zhanpeng Zhang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Learning social relation traits from face images. In *ICCV*, pages 3631–3639, 2015. 1
- [55] Yan-Guo Zhao, Zhan Song, Feng Zheng, and Ling Shao. Learning a multiple kernel similarity metric for kinship verification. *IS*, 430:247–260, 2018. 8
- [56] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, pages 9719–9728, 2020. 2
- [57] Xiuzhuang Zhou, Junlin Hu, Jiwen Lu, Yuanyuan Shang, and Yong Guan. Kinship verification from facial images under uncontrolled conditions. In *ACM MM*, pages 953–956, 2011. 2
- [58] Xiuzhuang Zhou, Jiwen Lu, Junlin Hu, and Yuanyuan Shang. Gabor-based gradient orientation pyramid for kinship verification under uncontrolled environments. In *ACM MM*, pages 725–728, 2012. 1, 2