KinNet: Fine-to-Coarse Deep Metric Learning for Kinship Verification

Yong Li[1], Jiabei Zeng[1], Jie Zhang[4], Anbo Dai[4], Meina Kan[1], Shiguang Shan[1,2,3,4], Xilin Chen[1]

[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology
[4]Seeta Tech. Inc., Beijing, 100190, China
{yong.li,jiabei.zeng,meina.kan}@vipl.ict.ac.cn
{sgshan,xlchen}@ict.ac.cn
{jie.zhang,anbo.dai}@seetatech.com

ABSTRACT

Automatic kinship verification has attracted increasing attentions as it holds promise to an abundance of applications. However, existing kinship verification methods suffer from the lack of large scale real-world data. Without enough training data, it is difficult to learn proper features that are discriminant for blood-related peoples. In this work, we propose KinNet, a fine-to-coarse deep metric learning framework for kinship verification. In the framework, we transfer knowledge from the large-scale-data-driven face recognition task, which is a fine-grained version of kinship recognition, by pre-training the network with massive data for face recognition. Then, the network is fine-tuned to find a metric space where kin-related peoples are discriminant. The metric space is learned by minimizing a soft triplet loss on the augmented kinship dataset. An augmented strategy is proposed to balance the amount of images per family member. Finally, we ensemble four networks to further boost the performance. The experimental results on the 1st Large-Scale Kinship Recognition Data Challenge (Track 1) demonstrate that our KinNet achieves the state-of-the-art performance in kinship verification.

KEYWORDS

kinship verification; deep metric learning; soft triplet loss; fine-to-coarse; data augmentation

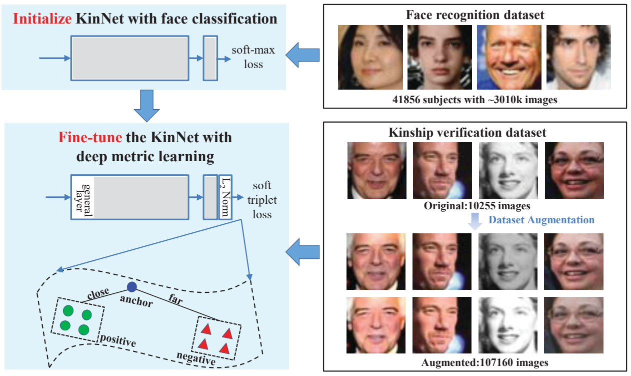

Figure 1: Main idea of the proposed KinNet under a fine-to-coarse manner. To generalize well on real-world kinship verification, the KinNet is firstly initialized with a large-scale data driven task of face classification. Then it is fine-tuned for the coarser-grained task by finding a kinship-specific metric space with soft triplet loss.

1 INTRODUCTION

Kinship verification aims to determine whether there is a specified kinship relation within a pair of facial images. Recently, kinship verification has received increasing attention from computer vision research community due to its wide potential applications, e.g., real-time paternity testing, family album organization, social media analysis, and missing children/parents search. However, it is a challenge to determine the biological relationship given only two facial images. To meet the challenge, researchers make efforts in discovering appearance resemblance between blood-related people by various learning methods [2, 3, 6, 10, 14–17, 21–28].

Although some encouraging results are obtained during the last few years, automatic kinship verification performs poorly in daily life applications due to the lack of large-scale real-world datasets. Existing datasets (Family101[2], Cornell KinFace[3], UB KinFace[17], KinFaceW-I[14], and KinFaceW-II[14]) contain very few examples and fail to reflect the true distributions of kinship relationships. For example, the prevalent KinFaceW-I dataset only has 156, 134, 116, and 127 pairs of images for father-son, father-daughter, mother-son, and mother-daughter kinship verification respectively. Classifiers trained from the limited-scale dataset fail to generalize well on real-world facial images with large diversity of appearances, poses, ethnicity, ages, and so on. Additionally, most image pairs of blood-related persons are cropped from the same photographs. The single source of images makes the verification system sensitive to various conditions of illumination, resolutions, and camera settings.

RFIW'17, October 27, 2017, Mountain View, CA, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5511-7/17/10…$15.00
https://doi.org/10.1145/3134421.3134425

In this work, we study the automatic kinship verification problem by adopting Families in the Wild (FIW) dataset, which is the largest and most comprehensive image dataset for kinship verification, provided by the 1st Large-Scale Kinship Recognition Data Competition. This competition supplies 300 families with totally 10255 images for training. Each family has 3 to 24 members and each member has 1 to 50 image samples. Relationship between every two family members is also provided. Figure 2 shows example images of some family members and their relationships in FIW dataset. Facial images in FIW dataset is collected from different search engines as well as social media outlets, improving the variations in occlusion, pose, lighting, ethnicity, and expression.

To improve the generalization ability of the kinship verification system, we propose a fine-to-coarse metric learning network, dubbed as KinNet, where the coarse-grained kinship verification model can benefit from the massive amount of existing face images that are for the fine-grained face recognition task. Figure 1 demonstrates the main idea of the proposed method. We initialize the KinNet by pre-training it on a face classification problem with millions of images in MS-Celeb-1M [4] dataset. Then, to make KinNet adapted to kinship verification task, we fine-tune the parameters on the augmented FIW dataset with parameters in the most general layer frozen. The FIW dataset is augmented by adding different illuminations. During the fine-tuning, we adopt a soft triplet loss to learn a non-linear metric space where the KINs (related pairs) distribute closely and the NON-KINs (unrelated pairs) distribute remotely. Below, we summarize the contributions of this work:

(1) We propose an end-to-end deep metric learning approach with a fine-to-coarse manner for kinship verification. To achieve a high generalization ability on real-world data, the network is initialized by face classification task with millions of facial images. Then the network is fine-tuned by a deep metric learning with kinship dataset.
(2) We introduce an augmentation schema to enlarge the kinship dataset and balance the amount of images for each family member. Experimental results show that the augmentation process boosts the verification performance.
(3) We adopt a soft triplet loss rather than a conventional triplet loss [18] to learn a kinship-specific metric space. Training with the soft triplet loss avoids enumerating all the possible triplets and achieves better performance.
(4) Intensive experimental study of the proposed scheme for robust kinship verification is conducted and our approach won the champion in the 1st Large-Scale Kinship Recognition Data Challenge (Track 1).

The rest of this paper is organized as follows: Section 2 reviews the related work about kinship verification. Section 3 details the proposed method and section 4 presents the experimental results. Section 5 concludes this work and discusses the further research.

2 RELATED WORK

In this section, we review the existing kinship verification methods that make efforts on designing or learning discriminant features. We classify these methods into three categories: handcrafted features based, shallow metric learning based, deep learning based.

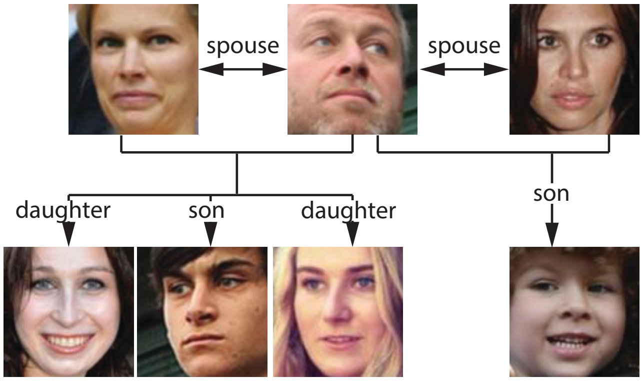

Figure 2: Example images within a family in FIW dataset.

Handcrafted features based: Early kinship verification methods usually extract handcrafted descriptors from facial images, and then learn classifiers based on them.

For example, Fang et al. [3] selected up to 22 facial features including eye color, skin color, and eye-to-nose distance for kinship verification. Zhou et al. [27] proposed a spatial pyramid learning-based feature descriptor that utilized both local and global information. Liu et al. [13] applied inheritable transformation on the Fisher vector derived for each image to enhance kinship verification accuracy. Kohli et al. [9] proposed to encode kinship similarity through self-similarity descriptor and formalized kinship verification as a two classification problem.

However, these low-level features cannot well represent the underlying visual resemblance among kinship-related people.

Shallow metric learning based: To learn discriminant features for kinship verification, researchers use metric learning methods to learn a Mahalanobis distance based upon the handcrafted features, to make the similarity score of a kinship-related pair higher than that of non kinship-related pairs. Lu et al. [14] adopted multiview neighborhood repulsed metric learning (MNRML) method to pull intraclass samples close and push interclass samples far. Hu et al. [6] proposed a large margin multi-metric learning (LM^{3}L) method which jointly learns multiple distance metrics to maximize distance of negative pairs and minimize distance of positive pairs. Yan et al. [25] utilized correlation similarity measure and exploited most discriminative encoded by negative samples for kinship verification. To avoid learning a full matrix from low-dimensional data that will bring about redundant computation and risk of over fitting, zhou et al. [23, 28] enforced sparsity on desired distance matrix to achieve better generality.

Deep learning based: Motivated by the impressive success of deep learning approaches in various image representation and classification[5, 11, 19, 20], more and more works adopt deep learning framework to learn discriminant features for kinship verification[1, 10, 12, 26]. Zhang et al. [26] adopted convolution neural network trained with concentrated image pairs for kinship verification, Kohli et al. [10] utilized filtered contractive deep belief networks to encode compact representation of facial images of kin. Dehghan et al. [1] introduced fusing the features and metrics via gated autoencoders to learn optimal features reflecting parent-offspring resemblance.

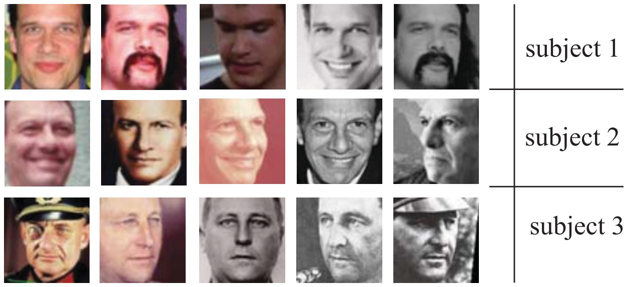

Figure 3: Three example subjects in the MS-Celeb-1M dataset, where images for each subject are with large diversity of appearances, poses, illuminations and so on.

3 PROPOSED METHOD

This section details the proposed fine-to-coarse deep metric learning method for robust kinship verification.

3.1 Method overview

Insufficient images in existing kinship datasets result in the trained model performing poorly on real-world kinship verification. To address this issue, we propose an end-to-end deep metric learning framework.

Figure 1 illustrates the main idea of the proposed method that integrates three aspects: transferring knowledge from face recognition task, augmenting kinship data, and learning kinship-specific metric space. First, We pre-train the kinship verification network with millions of face identities. Then augment the kinship dataset and balance the amount of images for each family member. Finally, we fine-tune the pre-trained network with augmented image data by a deep metric learning to adapt the model to kinship verification task. The following parts detail the three aspects respectively. In the end of this section, we discuss the ensemble of different models, which further improves the verification accuracy.

3.2 Knowledge Transferring from Face Recognition Task

Face recognition is highly related to the kinship verification task. Specifically speaking, it is a fine-grained version of kinship verification. Compared with the kinship datasets, datasets for face recognition are adequate in quantity and diversity. We use 3,095,536 images of 41,856 subjects from MS-Celeb-1M dataset during the training process. Figure 3 shows some examples in MS-Celeb-1M dataset. As can be seen, images for each subject are with various poses, illuminations, and image styles. The proposed KinNet aims to get knowledge from the large-scale data driven task by being pre-trained with millions of facial images for face recognition. The pre-trained network shows robustness to various illumination, poses, occlusions, expression, and ethnicity in the real-world applications.

We adopt different variations of residual network[5] as the basic architecture of KinNet because residual network converges fast and can be trained with hundreds or thousands of layers without degraded accuracy. Figure 4 illustrates the network architectures of 80-layer, 101-layer, 152-layer, and 269-layers residual network. As can be seen, networks of different depth differ in repeat times of basic residual blocks. For instance, ResNet-80 (80-layer residual network) is composed of three residual-block[0], eight residual-block[1], twelve residual-block[2], three residual-block[3] respectively.

The network is pre-trained on subset of MS-Celeb-1M dataset, with the output dimension of fc-layer as 41,856. A soft-max loss is placed after the fc-layer to classify the 41,856 subjects. During the fine-tune phase, the fc-layer is replaced with another one with 1024-dimensional outputs followed by a L_2-norm layer that normalizes the features into unit length. In the end, a soft triplet loss is configured to force the KINs close and NON-KINs far away from each other. Since the first layer learns the most general representations as some basic filters, we freeze the first 7×7 convolution layer and update other layers to make the model adapted to kinship verification.

3.3 Augmentation of Kinship Data

In the kinship dataset, despite being diverse, the image examples for the family members are imbalance in amount. The provided training data from FIW dataset contains 300 families, including 1828 family members and 10255 images in total. Figure 5 demonstrates the histogram of image examples for family member. X-axis denotes the number of images per member. Y-axis denotes the number of members. As can be seen, the number of images per member differs a lot. Majority of the members has less than 10 image examples, more than 250 people has only 1~2 image example, and a few ones have more than 20 examples.

To increase the amount of training images as well as balance the number of images per member, we propose an augmentation strategy in Algorithm 1 based on various methods proposed in[11, 19]. This algorithm make all the members in 300 families have the same amount of images. For members that have less images, Algorithm 1 augments their images by randomly choosing one or several of the following options:

- Gamma correction, which obtain output gamma corrected image by the following operation:

  $$O = I\hat{}(1/G)$$

  where I and O denote the input and output image respectively. $G < 1$ will shift the image towards darker and vice versa.
- Decreasing the resolution by a down-sample and up-sample process.
- Adding some gaussian noise to the input image.

3.4 Learning Kinship-Specific Metric Space

As mentioned before, the KinNet is fine-tuned on the kinship data. In kinship verification task, facial expression, scale, illumination, occlusion and some uncontrolled factors may dramatically change the appearance variations of faces. To address this issue, we learns a non-linear mapping function by a deep neural network that maps the input image to a low-dimensional feature space, yielding some visual constraints among facial tracks. This is technically realized by minimizing a soft triplet loss [7] function, which requires the distance between positive pairs to be less than that of negative

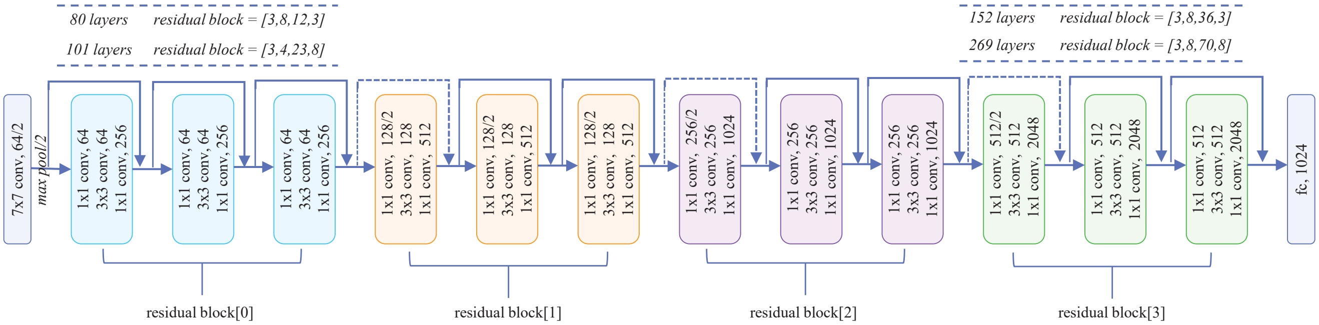

Figure 4: The architecture of the four deep residual neural network adopted for kinship verification. bottom: basic residual blocks that constitute deep residual network, top: repeat times of the four residual blocks included in network of different depth.

Algorithm 1: Algorithm for augmenting kinship image data

```
Input:
    Total amount of family members, $total\_num$
    A set of functions designed for augmenting images, $func\_array$
    length of function set, $func\_array\_len$
Output:
    Augmented kinship images, amount of images for each family member is balanced
    /* firstly, determine maximum images per family member contains */
    set $max\_num \Leftarrow -1$
    set $final\_num \Leftarrow 0$
    for $i = 1; i < total\_num; i++$ do
        $current\_num_i = count(member[i])$
        if $max\_num < current\_num_i$ then
            $max\_num \Leftarrow current\_num_i$
        end if
    end for
    /* secondly, determine desired number of images each family member will contain after augmentation */
    set $final\_num \Leftarrow 1.2 * max\_num$
    /* finally, augment images for every family member */
    for $i = 1; i < total\_num; i++$ do
        $aug\_num_i \Leftarrow final\_num - current\_num_i$
        while $aug\_num_i \neq 0$ do
            $func\_index = rand()\%func\_array\_len$
            $status = func\_array(func\_index)$
            if $status$ then
                $aug\_num_i \Leftarrow aug\_num_i - 1$
            end if
        end while
    end for
```

pairs. In the following part, we present the technique details of soft triplet loss, including the formulation, triplet settings, hard sample selection, and gradient computation.

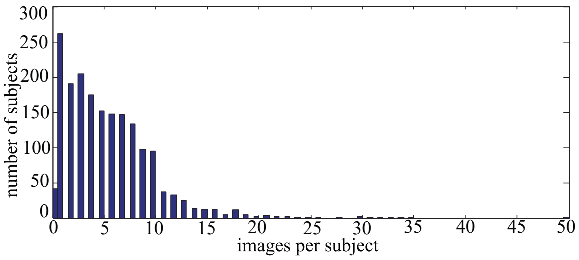

Figure 5: Distribution of image numbers per family member in the original dataset.

Formulation: Suppose $\mathcal{X}$ is a training set with class labels for every sample. $x_i \sim x_j$ denotes x_i and x_j are of the same category; $x_i \nsim x_k$ denote they are of different categories. The empirical error on all triplet $(x_i, x_j, x_k), x_i \sim x_j, x_i \nsim x_k$ constrains from $\mathcal{X}$ is defined by

$$\begin{aligned} \ell(\mathcal{X}) &= \Pr(d_{ij} > d_{ik} | x_j \sim x_i, x_k \nsim x_i) \\ &= \mathbb{E}_{x_i \sim x_j, x_i \nsim x_k} \mathbf{1}_{d_{ij} > d_{ik} - c} \end{aligned} \tag{1}$$

where $d_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)$ is the distance between x_i and x_j, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the features of x_i and x_j respectively. c is the margin, which requires that the distance between positive pair should be less than those of negative pair by c, or a penalty would be added in the loss.

To get a smooth and convex loss, we replace $\mathbf{1}(.)$ with the exponential-based logit surrogate function $\psi_\beta(e^z) = \ln(1 + \beta e^z)/\ln(1 + \beta)$ and obtain an upper bound of (1)

$$\ell(\mathcal{X}) \leq \mathbb{E}_{x_i \sim x_j, x_i \nsim x_k} \psi_\beta(e^{d_{ij} - d_{ik} + c}) \tag{2}$$

However, (2) is complex in computation because enumerating all the triplets (x_i, x_j, x_k) could be $O(N^3)$. By using the concavity of $\psi_\beta(.)$, (2) can be further relaxed to a soft version:

$$\begin{aligned} \mathbb{E}_{x_i \sim x_j, x_i \nsim x_k} \psi_\beta(e^{d_{ij} - d_{ik} + c}) &\leq \mathbb{E}_{x_i} \psi_\beta \left(\mathbb{E}_{x_j \sim x_i, x_k \nsim x_i} e^{d_{ij} - d_{ik} + c} \right) \\ &= \mathbb{E}_{x_i} \psi_\beta \left(\mathbb{E}_{x_j \sim x_i} e^{d_{ij}} \cdot \mathbb{E}_{x_k \nsim x_i} e^{-d_{ik}} \cdot e^c \right) \\ &= \mathbb{E}_{x_i} \psi_\beta \left(\phi_i^+ \cdot \phi_i^- \cdot e^c \right) = \ell, \end{aligned} \tag{3}$$

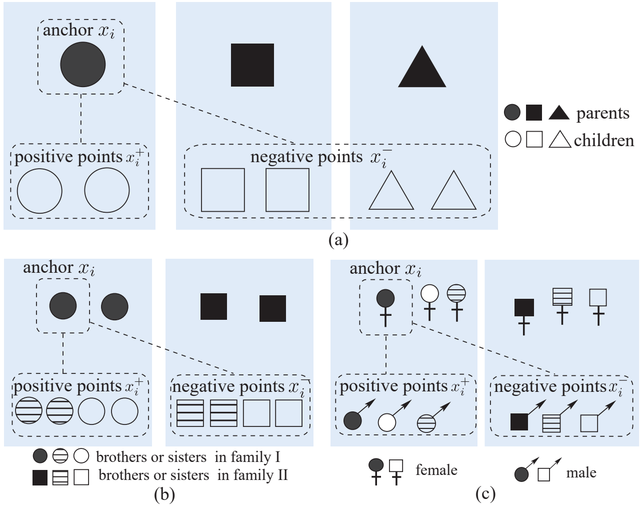

Figure 6: Soft triplet selection strategies in (a) parent-children relationship, (b) brothers or sisters relationship (same gender), (c) siblings relationship (different genders). Different shapes denote different families. Different textures denote different members.

where

$$\phi_i^+ = \mathbb{E}_{x_j \sim x_i} e^{d_{ij}} = \frac{1}{N_i^+} \sum_{x_j \sim x_i} e^{d_{ij}} \tag{4}$$

$$\phi_i^- = \mathbb{E}_{x_k \nsim x_i} e^{-d_{ik}} = \frac{1}{N_i^-} \sum_{x_k \nsim x_i} e^{-d_{ik}} \tag{5}$$

In (3), the loss is computed on soft triplet (x_i, x_i^+, x_i^-), where $x_i^+ = \{x_j | x_j \sim x_i\}$ and $x_i^- = \{x_k | x_k \nsim x_i\}$. A soft triplet contains one anchor point, a set of positive points, and a set of negative points.

Soft-Triplet Settings: We train a kinship-specific non-linear metric space for each type of kinship. Batch-based optimization method is applied to minimize the soft triplet loss. To avoid the absence of positive or negative pairs in a mini-batch, we shuffle the input images by preserving a subgroup of family unseparated. Below, we discuss how we we define a soft triple for different kinship types.

- For parent-child relationship, i.e., father-daughter (FD), father-son (FS), mother-daughter (MD), and mother-son (MS), we input images in several families within a mini-batch. For each family, we include N_p images for the parent and N_c images for the child. Figure 6 (a) illustrates the soft triplets settings for parent-child relationship. Same shapes represent images in the same family. Solid shapes are parents and empty shapes are the children. In the illustrated example, three families are included with $N_p = 1$ and $N_c = 2$. To determine a soft triplet, we select an image of the parent as anchor $\mathbf{x}$. Then the positive points $\mathbf{x}^+$ are the images of the anchor's child, the negative points $\mathbf{x}^-$ are the images of others' children.
- For brothers (BB) or sisters (SS) relationship, we input the brothers or sisters as a family within a mini-batch. Figure 6 (b) illustrates the soft triplets settings for brothers or sisters. Same shapes denote members in the same family. Different textures denote different members. There are three brothers in each family and two example images are included for each member. To determine a soft triplet, we randomly select an image as anchor $\mathbf{x}$. Then, images of the anchor's brothers/sisters are positive points $\mathbf{x}^+$ and those of members in other families are negative points $\mathbf{x}^-$.
- For siblings relationship (SIBS), we input the siblings as a family within a mini-batch. Figure 6 (c) illustrates the soft triplets settings for siblings. In a soft triplet, an image is randomly selected as the anchor $\mathbf{x}$. Then, images of the anchor's siblings with the opposite gender are selected as positive points $\mathbf{x}^+$. Images of members in other families are negative points $\mathbf{x}^-$.

Hard Sample Selection: Positive or negative pairs are not equally important during the training process. To select the most informative points, we design an online exemplars mining strategy in a mini-batch. Since pairs with higher loss have more impact to the model, we select the hard samples as positive and negative points to construct soft triplets. Specifically, we sort the positive points by its distance to the anchor, and then choose the K_p-st farthest ones as $\mathbf{x}^+$ in the soft triplet. Similarly, we sort the negative points and choose the K_n-st nearest ones as $\mathbf{x}^-$.

Gradient Computation: During the back propagation, for every general triplet, we compute the gradient with respect to the feature of anchor point $\mathbf{x}_i$, feature of each positive point $\mathbf{x}_j$, and feature of each negative point $\mathbf{x}_k$.

For anchor point $\mathbf{x}_i$,

$$\begin{aligned}\frac{\partial \ell}{\partial \mathbf{x}_i} &= \frac{\partial \ell}{\partial \phi_i^+ \phi_i^-} \frac{\partial \phi_i^+ \phi_i^-}{\partial \mathbf{x}_i} \\ &= \frac{\beta e^c}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)} \left(\frac{\partial \phi_i^+}{\partial \mathbf{x}_i} \phi_i^- + \phi_i^+ \frac{\partial \phi_i^-}{\partial \mathbf{x}_i} \right) \\ &= \frac{2\beta e^c \left(\frac{\phi_i^-}{N_i^+} \sum_{x_j \sim x_i} e^{d_{ij}} (\mathbf{x}_i - \mathbf{x}_j) - \frac{\phi_i^+}{N_i^-} \sum_{x_k \nsim x_i} e^{-d_{ik}} (\mathbf{x}_i - \mathbf{x}_k) \right)}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)}\end{aligned} \tag{6}$$

For each positive point $\mathbf{x}_j$,

$$\begin{aligned}\frac{\partial \ell}{\partial \mathbf{x}_j} &= \frac{\partial \ell}{\partial \phi_i^+ \phi_i^-} \frac{\partial \phi_i^+ \phi_i^-}{\partial \mathbf{x}_j} \\ &= \frac{\beta e^c}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)} \left(\frac{\partial \phi_i^+}{\partial \mathbf{x}_j} \phi_i^- \right) \\ &= \frac{2\beta e^c \phi_i^-}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)} \frac{e^{d_{ij}}}{N_i^+} (\mathbf{x}_j - \mathbf{x}_i)\end{aligned} \tag{7}$$

For each negative point $\mathbf{x}_k$,

$$\begin{aligned}\frac{\partial \ell}{\partial \mathbf{x}_k} &= \frac{\partial \ell}{\partial \phi_i^+ \phi_i^-} \frac{\partial \phi_i^+ \phi_i^-}{\partial \mathbf{x}_k} \\ &= \frac{\beta e^c}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)} \left(\frac{\partial \phi_i^-}{\partial \mathbf{x}_k} \phi_i^+ \right) \\ &= \frac{2\beta e^c \phi_i^+}{N \ln(1+\beta)(1+\beta\phi_i^+\phi_i^- e^c)} \frac{e^{-d_{ik}}}{N_i^-} (\mathbf{x}_i - \mathbf{x}_k)\end{aligned} \tag{8}$$

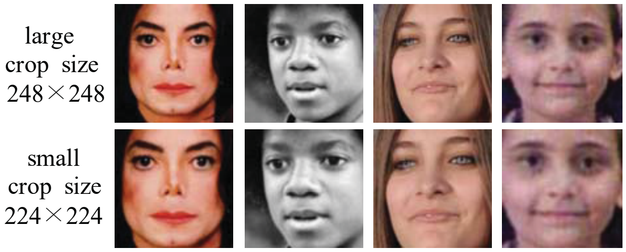

Figure 7: Examples of cropped faces with different crop size.

Table 1: Crop size settings of different network

method	ResNet−80	ResNet−101	ResNet−152	ResNet−269
crop size	248 × 248	248 × 248	248 × 248	224 × 224

3.5 Ensemble Learning

The final kinship verification result is obtained from the fusion of four models with different network depth and input images' crop sizes. We adopt the 80-layer, 101-layer, 152-layers and 269-layer residual networks as the single model. In the rest part of this paper, we call them ResNet-80, ResNet-101, ResNet-152, and ResNet 269, respectively. Figure 4 illustrates the detailed architectures of the four networks. The input images of each network are randomly cropped with two crop sizes, so as to become robust against random perturbation. Figure 7 displays examples of the two kinds of crop faces. The ones with larger crop size keep more context and background information. The ones with smaller crop size are compact and only includes internal facial organs. The images are randomly cropped during training and centered cropped during test. Table 1 shows the crop sizes set in the four models. ResNet-80, ResNet-101, and ResNet-152 share a same crop size of 248 × 248, while ResNet-269 adopts 224 × 224.

Given a pair of facial images, we predict the presence of a specified kinship by comparing the Cosine similarity with a learned threshold. The similarity is computed by regarding the outputs of fc-layer as features. We fuse multiple models by computing the final similarity as a weighted sum of the four similarities from the four models. Mathematically speaking, the final decision value $s = \alpha_1 s_1 + \alpha_2 s_2 + \alpha_3 s_3 + \alpha_4 s_4$, where s_i is the similarity from model i, α_i is the fusion coefficient for model i and yields $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. We determine the fusion coefficients in two ways as follows:

- Average weights: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$;
- Adjusted weights: Getting the optimized weights by brute-force searching.

4 EXPERIMENTS

In this section, we present the experimental evaluations of the proposed scheme. First, we briefly review the first Large-Scale Kinship Recognition Data Challenge (Track 1). Then we describe our implementation details. Finally, we present a comprehensive study of the proposed method.

Table 2: Number of pairs in training/validation/test

kinship type	train (pair)	validation (pair)	test (pair)
father-daughter	42458	11460	23506
father-son	53974	13696	45988
mother-daughter	34828	10698	20674
mother-son	38312	9816	47954
brothers	52483	17342	19946
sisters	19286	6218	6524
siblings	40846	7434	15076

4.1 The First Large-Scale Kinship Recognition Data Challenge (Track 1)

The first Large-Scale Kinship Recognition Data Challenge (Track 1) aims to determine whether a pair of facial images are blood relatives of one of seven types, i.e., father-daughter, father-son, mother-daughter, mother-son, brother-brother, sister-sister, siblings. This is a classical Boolean problem with system responses being either KIN (related) or NON-KIN (unrelated). The competition releases the largest and most comprehensive image database for automatic kinship recognition, called Families in the Wild (FIW), which has around 644,000 face pairs for kinship verification from 300 families.

The challenge is composed of development phase and the final test phase. In the development phase, the model is trained on the training set and evaluation is conducted on the validation set. In the final test phase, both the training set and validation set can be deployed for model training. Table 2 presents the number of KIN pairs in training/validation/test part for each kinship type. The performance is measured by the accuracy of binary classification for each kinship type.

4.2 Implementation Details

We implemented the proposed method using Caffe deep learning framework[8]. A batch-based stochastic gradient descent method is applied to optimize the model. The base learning rate was set as 0.01 and was reduced by polynomial policy with gamma of 0.1. The momentum was set as 0.9 and the weight decay was set as 0.0005. The training of models was completed on a Titan-X GPU with 12GB memory. During the pre-training stage, we set the batch size as 128 and the maximum iterations as 300K. It took 4.5 days to get a well pre-trained model. During the fine-tuning stage, we set the batch size as 80 and the maximum iterations as 2K. It took about 43 minutes to fine-tune the model for a single kinship type. The maximum iterations for pre-training and fine-tuning were different because they used different amounts of images. Around 3010K images were used during pre-training. For finetuning, the pre-augmented image amount was merely 10K. Although the augmetation for kinship dataset remedies the imbalance among different members, it improves little in class-wise diversity. Thus, according to the scale of original dataset, the number of epoches were similar in pretraining and finetuning, which were 12.7 and 15.6 respectively.

When learning the metric space with soft triplet loss, we included images from four families within a batch. For each family, we included 10 images as anchors and 10 non-anchor images that are related to the anchors with a specific kinship. Thus, during each

Table 3: Accuracy on validation set with different settings

method	FD	FS	MD	MS	BB	SS	SIBS	*avg.*
w/o pre-train	0.5149	0.5240	0.5652	0.5270	0.5087	0.5412	0.5184	0.5285
pre-train+fine-tune (triplet loss)	0.6802	0.6812	0.6756	0.7235	0.7261	0.7321	0.6993	0.7026
pre-train+fine-tune (soft triplet loss)	0.6902	0.7085	0.7149	0.7229	0.7298	0.7806	0.6770	0.7177
pre-train+fine-tune (soft triplet loss)+aug.	0.7041	0.7216	0.7314	0.7493	0.7530	0.7978	0.6911	0.7355

Table 4: Accuracy on validation set with single and ensemble models

Experiment	FD	FS	MD	MS	BB	SS	SIBS	*avg.*
ResNet-80	0.7041	0.7216	0.7314	0.7493	0.7530	0.7978	0.6911	0.7355
ResNet-101	0.6603	0.7252	0.7353	0.6781	0.8235	0.8067	0.6936	07318
ResNet-152	0.6886	0.7351	0.7274	0.6845	0.8287	0.7974	0.7168	0.7397
ResNet-269	0.6761	0.7310	0.7379	0.6906	0.8508	0.8036	0.7012	0.7416
All fusion(average weight)	0.7277	0.7699	0.7582	0.7564	0.8419	0.7935	0.7272	0.7678
All fusion(adjusted weight)	0.7277	0.7729	0.7582	0.7629	0.8419	0.8165	0.7272	07724

iteration, we updated the parameters by learning with around 40 soft triplets. In each soft triplet, we selected 5 non-anchor images in the anchor's family as positive images and 6 non-anchor images in other families as negative images, by using hard sample selection strategy discussed in Section 3.4. The margin in (3) was tuned for best performance.

4.3 Experimental Results

In this section, we evaluated the effectiveness of the proposed method on the FIW dataset through comprehensive experiments. Table 3 and Table 4 show the accuracy on validation set with different methods. Below, we discuss the effectiveness of different configuration according to the experimental result.

Knowledge transferring by pre-training: The effectiveness of the knowledge transferring setting was validated by comparing *w/o pre-train* and *pre-train+fine-tune (soft triplet loss)*. In *w/o pre-train*, a Resnet-80 was trained for each kinship type from scratch on the training data by minimizing the soft triplet loss. In *pre-train+fine-tune (soft triplet loss)*, the Resnet-80 was pre-trained on MS-Celeb-1M before fine-tuned on the kinship data by minimizing the soft triplet loss. As can be seen from Table 3, the accuracy for every kinship type is dramatically improved by the pre-training process. It indicates that the features that discriminate different individuals also capture some information about the appearance resemblance between blood-related people. Thus, it can be transferred from the large-scale data driven face recognition task to kinship verification task by the pre-training strategy.

Augmentation of kinship data: We compared *pre-train+fine-tune (soft triplet loss)* with *pre-train+fine-tune (soft triplet loss)+aug* to investigate the effectiveness of data augmentation. In *pre-train+fine-tune (soft triplet loss)+aug*, the pre-trained Resnet-80 was fine-tuned with augmented kinship data instead of the original one. The augmentation method are detailed in Section 3.3. As can be seen in Table 3, adding the augmentation process achieves 1.31%~2.64% gains for every kinship type, and an overall 1.78% gains on the average accuracy. This is because data augmentation enlarges the dataset size and eliminates the member-wise imbalance exist in original dataset.

Soft triplet loss V.S. triplet loss: We compared soft triplet loss and triplet loss by *pre-train+fine-tune (soft triplet loss)* and *pre-train+fine-tune (triplet loss)*, the difference between the two methods is that *pre-train+fine-tune (triplet loss)* fine-tuned the model with a conventional triplet loss [18]. The triplets used in *pre-train+fine-tune (triplet loss)* are extracted from the positive and negative pairs provided by the competition. As can be seen in Table 3, soft triplet loss obtains higher accuracy on FD, FS, MD, BB, SS, and the overall average result. The results with soft triplet loss were further improved by augmenting the original dataset. We did not conduct experiments on augmented data with triplet loss because the number of conventional triplets increase exponentially after the augmentation. The enormous number of triplets make it hard to select the informative ones for training. Compared to the conventional triplet loss, soft triplet loss is much more flexible when the scale of dataset increases.

Ensemble of multiple models: We fused four residual-network based models with 80, 101, 152, and 269 layers. All the four models are pre-trained on MS-Celeb-1M dataset. The detailed fusion methods are presented in Section 3.5. Table 4 reports the accuracy on validation set with single and the fused model. As can be seen, on either fusion setting, the fused models significantly improve the performance of any single models. By adjusting the fusion weights of the single models, we further improve the accuracy by assigning better-performed model with higher weight.

Comparison with other participants: Our method achieved the best performance on the public leaderboard of the first large-scale kinship recognition data challenge (Track 1). Table 5 show the result on test set posted in public leaderboard. As can be seen, we are in the 1st place for every kinship type and the overall performance. We are higher than the 2nd place with 5.3%, 7.6%, 8.9%, 10.1%, 0.4%, 6.2%, 3.9% for MD, MS, SS, BB, SIBS, FS, FD respectively, and 6% for the overall average accuracy.

Table 5: Accuracy on test set on public leaderboard

Team	MD	MS	SS	BB	SIBS	FS	FD	*avg.*
Ours	**0.778708**	**0.786170**	**0.799050**	**0.747719**	**0.705691**	**0.714621**	**0.707904**	**0.748552**
Unknown	0.723131	0.710914	0.710914	0.646646	0.701114	0.652844	0.668851	0.688616
Unknown	0.706588	0.721030	0.721030	0.635867	0.665097	0.633839	0.645963	0.665802
Unknown	0.622956	0.638259	0.638259	0.605986	0.616211	0.619792	0.613078	0.613307
Unknown	0.638048	0.635806	0.635806	0.627043	0.645728	0.632556	0.613078	0.632174
Unknown	0.650285	0.645923	0.645923	0.617016	0.643075	0.621532	0.607675	0.630971
Unknown	0.600029	0.612048	0.612048	0.596912	0.580194	0.605941	0.561601	0.582052
Unknown	0.548902	0.579552	0.579552	0.533841	0.522486	0.559320	0.549221	0.548094

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a fine-to-coarse deep metric learning framework approach to learn robust facial features that are related to biological resemblance between kinship pairs. we transfer knowledge from the large-scale-data-driven face recognition task to reduce the risk of over-fitting, then fine-tune the pre-trained network using soft-triplet loss to force the KINs close and NON-KINs far away from each other in the metric space. Both the data augmentation and soft-triplet loss improve verification performance.

For future work, we will study how to jointly learn local and global resemblance between kinship pairs in a unified learning framework.

ACKNOWLEDGMENTS

This work was partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61650202, 61402443, 61272321, and the Strategic Priority Research Program of the CAS (Grant XDB02070004).

REFERENCES

[1] Afshin Dehghan, Enrique G Ortiz, Ruben Villegas, and Mubarak Shah. 2014. Who Do I Look Like? Determining Parent-Offspring Resemblance via Gated Autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1757–1764.

[2] Ruogu Fang, Andrew C Gallagher, Tsuhan Chen, and Alexander Loui. 2013. Kinship classification by modeling facial feature heredity. In *International Conference on Image Processing*. IEEE, 2983–2987.

[3] Ruogu Fang, Kevin D. Tang, Noah Snavely, and Tsuhan Chen. 2010. Towards computational models of kinship verification. In *IEEE International Conference on Image Processing*. 1577–1580.

[4] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. 2016. MS-Celeb-1M: A Dataset and Benchmark for Large Scale Face Recognition. In *European Conference on Computer Vision*. 87–102.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. (2015), 770–778.

[6] Junlin Hu, Jiwen Lu, Junsong Yuan, and Yappeng Tan. 2014. Large Margin Multi-metric Learning for Face and Kinship Verification in the Wild. (2014), 252–267.

[7] Chang Huang, Shenghuo Zhu, and Kai Yu. 2012. Large Scale Strongly Supervised Ensemble Metric Learning, with Applications to Face Verification and Retrieval. *Computer Science* (2012).

[8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[9] N Kohli, R Singh, and M Vatsa. 2012. Self-similarity representation of Weber faces for kinship classification. In *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems*. 245–250.

[10] Naman Kohli, Mayank Vatsa, Richa Singh, Afzel Noore, and Angshul Majumdar. 2016. Hierarchical Representation Learning for Kinship Verification. *IEEE Transactions on Image Processing* 26, 1 (2016), 289–302.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*. 1097–1105.

[12] Lei Li, Xiaoyi Feng, Xiaoting Wu, Zhaoqiang Xia, and Abdenour Hadid. 2016. Kinship Verification from Faces via Similarity Metric Based Convolutional Neural Network. (2016), 539–548.

[13] Qingfeng Liu, A Puthenputhussery, and Chengjun Liu. 2015. Inheritable Fisher vector feature for kinship verification. In *IEEE International Conference on Biometrics Theory, Applications and Systems*. 1–6.

[14] Jiwen Lu, Xiuzhuang Zhou, Y C Tan, Yuanyuan Shang, and Jie Zhou. 2014. Neighborhood Repulsed Metric Learning for Kinship Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 2 (2014), 331–345.

[15] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. 2015. Tri-Subject Kinship Verification: Understanding the Core of A Family. *IEEE Transactions on Multimedia* 17, 10 (2015), 1855–1867.

[16] Joseph P Robinson, Ming Shao, Yue Wu, and Yun Fu. 2016. Families in the Wild (FIW): Large-Scale Kinship Image Database and Benchmarks. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 242–246.

[17] J. Luo S. Xia, M. Shao and Y. Fu. 2012. Understanding Kin Relationships in a Photo. *IEEE Transactions on Multimedia* 14, 4 (2012), 1046–1056.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition*. 815–823.

[19] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science* (2014).

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, S Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Computer Vision and Pattern Recognition*. 1–9.

[21] Shuyang Wang, Joseph P Robinson, and Yun Fu. 2017. Kinship Verification on Families in the Wild with Marginalized Denoising Metric Learning. In *Automatic Face and Gesture Recognition (FG)*. 216–221.

[22] Siyu Xia, Ming Shao, and Yun Fu. 2011. Kinship verification through transfer learning. *IJCAI*, 2539–2544.

[23] Min Xu and Yuanyuan Shang. 2017. Kinship Measurement on Face Images by Structured Similarity Fusion. *IEEE Access* 4, 99 (2017), 10280–10287.

[24] Haibin Yan, Jiwen Lu, Weihong Deng, and Xiuzhuang Zhou. 2014. Discriminative Multimetric Learning for Kinship Verification. *IEEE Transactions on Information Forensics and Security* 9, 7 (2014), 1169–1178.

[25] Haibin Yan, Jiwen Lu, and Xiuzhuang Zhou. 2015. Prototype-Based Discriminative Feature Learning for Kinship Verification. *IEEE Transactions on Systems, Man, and Cybernetics* 45, 11 (2015), 2535–2545.

[26] Kaihao Zhang, Yongzhen Huang, Chunfeng Song, Hong Wu, and Liang Wang. 2015. Kinship Verification with Deep Convolutional Neural Networks. In *British Machine Vision Conference*.

[27] Xiuzhuang Zhou, Junlin Hu, Jiwen Lu, Yuanyuan Shang, and Yong Guan. 2011. Kinship verification from facial images under uncontrolled conditions. (2011), 953–956.

[28] Xiuzhuang Zhou, Yuanyuan Shang, Haibin Yan, and Guodong Guo. 2016. Ensemble similarity learning for kinship verification from facial images in the wild. *Information Fusion* 32, 32 (2016), 40–48.