2013 IEEE International Conference on Computer Vision

Hybrid Deep Learning for Face Verification

Yi Sun[1] Xiaogang Wang[2,3] Xiaoou Tang[1,3]

[1]Department of Information Engineering, The Chinese University of Hong Kong

[2]Department of Electronic Engineering, The Chinese University of Hong Kong

[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

sy011@ie.cuhk.edu.hk xgwang@ee.cuhk.edu.hk xtang@ie.cuhk.edu.hk

Abstract

This paper proposes a hybrid convolutional network (ConvNet)-Restricted Boltzmann Machine (RBM) model for face verification in wild conditions. A key contribution of this work is to directly learn relational visual features, which indicate identity similarities, from raw pixels of face pairs with a hybrid deep network. The deep ConvNets in our model mimic the primary visual cortex to jointly extract local relational visual features from two face images compared with the learned filter pairs. These relational features are further processed through multiple layers to extract high-level and global features. Multiple groups of ConvNets are constructed in order to achieve robustness and characterize face similarities from different aspects. The top-layer RBM performs inference from complementary high-level features extracted from different ConvNet groups with a two-level average pooling hierarchy. The entire hybrid deep network is jointly fine-tuned to optimize for the task of face verification. Our model achieves competitive face verification performance on the LFW dataset.

1. Introduction

Face recognition has been extensively studied in recent decades [29, 28, 30, 1, 16, 5, 33, 12, 6, 3, 7, 25, 34]. This paper addresses the key challenge of computing the similarity of two face images given their large intra-personal variations in poses, illuminations, expressions, ages, makeups, and occlusions. It becomes more difficult when faces to be compared are acquired in the wild. We focus on the task of face verification, which aims to determine whether two face images belong to the same identity.

Existing methods generally address the problem in two steps: feature extraction and recognition. In the feature extraction stage, a variety of hand-crafted features are used [10, 22, 20, 6]. Although some learning-based feature extraction approaches are proposed, their optimization targets are not directly related to face identity [5, 13]. Therefore, the features extracted encode intra-personal variations. More importantly, existing approaches extract features from each image separately and compare them at later stages [8, 16, 3, 4]. Some important correlations between the two compared images have been lost at the feature extraction stage.

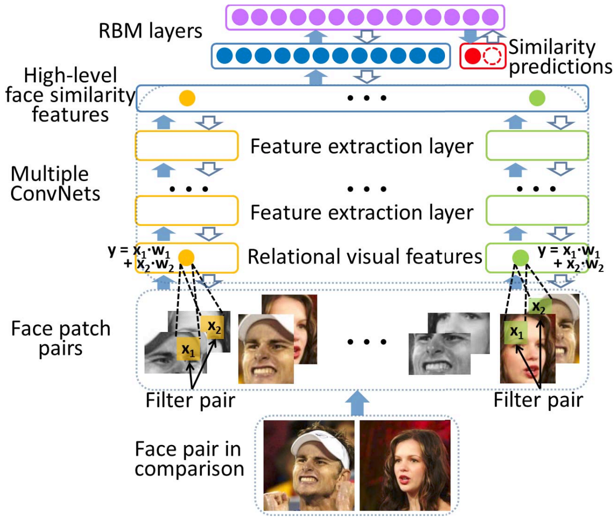

Figure 1: The hybrid ConvNet-RBM model. Solid and hollow arrows show forward and back propagation directions.

At the recognition stage, classifiers such as SVM are used to classify two face images as having the same identity or not [5, 24, 13], or other models are employed to compute the similarities of two face images [10, 22, 12, 6, 7, 25]. The purpose of these models is to separate inter-personal variations and intra-personal variations. However, all of these models have been shown to have shallow structures [2]. To handle large-scale data with complex distributions, large amount of over-completed features may need to be extracted from the face [12, 7, 25]. Moreover, since the feature extraction stage and the recognition stage are separate, they cannot be jointly optimized. Once useful information is lost

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.188

in feature extraction, it cannot be recovered in recognition. On the other hand, without the guidance of recognition, the best way to design feature descriptors to capture identity information is not clear.

All of the issues discussed above motivate us to learn a hybrid deep network to compute face similarities. A high-level illustration of our model is shown in Figure 1. Our model has several unique features, as outlined below.

(1) It directly learns visual features from raw pixels under the supervision of face identities. Instead of extracting features from each face image separately, the model jointly extracts relational visual features from two face images in comparison. In our model, such relational features are first locally extracted with the automatically learned filter pairs (pairs of filters convolving with the two face images respectively as shown in Figure 1), and then further processed through multiple layers of the deep convolutional networks (ConvNets) to extract high-level and global features. The extracted features are effective for computing the identity similarities of face images.

(2) Considering the regular structures of faces, the deep ConvNets in our model locally share weights in higher convolutional layers, such that different mid- or high-level features are extracted from different face regions, which is contrary to conventional ConvNet structures [18], and can greatly improve their fitting and generalization capabilities.

(3) The deep and wide architecture of our hybrid network can handle large-scale face data with complex distributions. The deep ConvNets in our network have four convolutional layers (followed by max-pooling) and two fully-connected layers. In addition, multiple groups of ConvNets are constructed to achieve good robustness and characterize face similarities from different aspects. Predictions from multiple ConvNet groups are pooled hierarchically and then associated by the top-layer RBM for the final inference.

(4) The feature extraction and recognition stages are unified under a single network architecture. The parameters of the entire pipeline (weights and biases in all the layers) are jointly optimized for the target of face verification.

2. Related work

All existing methods for face verification start by extracting features from two faces in comparison separately. A variety of low-level features are commonly used [27, 10, 22, 33, 20, 6], including the hand-crafted features like LBP [23] and its variants [32], SIFT [21], Gabor [31] and the learned LE features [5]. Some methods generated mid-level features [24, 13] with variants of convolutional deep belief networks (CDBN) [19] or ConvNets [18]. They are not learned with the supervision of identity matching. Thus variations other than identity are encoded in the features, such as poses, illumination, and expressions, which constitute the main impediment to face recognition.

Many face recognition models are shallow structures, and need high-dimensional over-completed feature representations to learn the complex mappings from pairs of noisy features to face similarities [12, 7, 25]; otherwise, the models may suffer from inferior performance. Many methods [5, 24, 13] used linear SVM to make the same-or-different verification decisions. Li *et al*. [20] and Chen *et al*. [6, 7] factorized the face images as identity variations plus variations within the same identity, and assumed each factor as a Gaussian distribution for closed form solutions. Huang *et al*. [12] and Simonyan *et al*. [25] learns linear transformations via metric learning.

Some methods further learn high-level features based on low-level hand-crafted features [16, 3, 4]. They are outputs of classifiers that are trained to distinguish faces of different people. All these methods extract features from a single face separately, and the comparison of two face images are deferred in the later recognition stage. Some identity information may have been lost in the feature extraction stage, and it cannot be retrieved in the recognition stage, since the two stages are separated in the existing methods. To avoid the potential information loss and make a reliable decision, a large amount of high-level feature extractors may need to be trained [3, 4].

There are a few methods that also used deep models for face verification [8, 24, 13], but extracted features independently from each face. Thus relations between the two faces are not modeled at their feature extraction stages. In [34], face images under various poses and lighting conditions were transformed to a canonical view with a convolutional neural network. Then features are extracted from the transformed images. In contrast, we deal with face pairs directly by extracting relational visual features from the two compared faces. The top layer RBM in our model is similar to that of the deep belief net (DBN) proposed by Hinton and Osindero [11]. However, we use ConvNets instead of stack of RBMs in the lower layers to take the local correlation in images into consideration. Averaging the results of multiple ConvNets has been shown to be an effective way of improving performance [9, 15], while we will show that our hybrid structure is significantly better than the simple averaging scheme. Moreover, unlike most existing face recognition pipelines, in which each stage is optimized independently, our hybrid ConvNet-RBM model is jointly optimized after pre-training each part separately, which further enhances its performance.

3. The hybrid ConvNet-RBM model

3.1. Architecture overview

We detect the two eye centers and mouth center with the facial point detection method proposed by Sun *et al*. [26]. Faces are aligned by similarity transformation according to

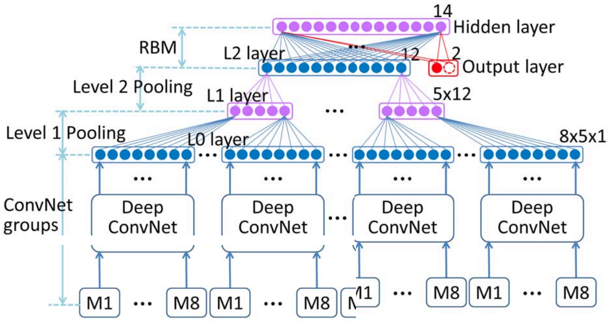

Figure 2: Architecture of the hybrid ConvNet-RBM model. Neuron (or feature) number is marked beside each layer.

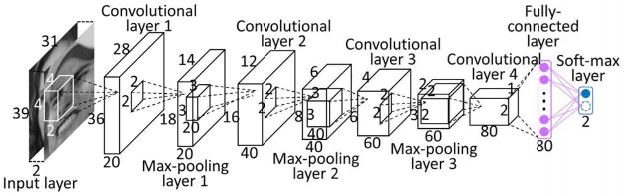

Figure 3: The structure of one ConvNet. The map numbers and dimensions of the input layer and all the convolutional and max-pooling layers are illustrated as the length, width, and height of cuboids. The 3D convolution kernel sizes of the convolutional layers and the pooling region sizes of the max-pooling layers are shown as the small cuboids and squares inside the large cuboids of maps respectively. Neuron numbers of other layers are marked beside each layer.

the three points. Figure 2 is an overview of our hybrid ConvNet-RBM model, which is a cascade of deep ConvNet groups, two levels of average pooling, and Classification RBM.

The lower part of our hybrid model contains 12 groups, each of which contains five ConvNets. Figure 3 shows the structure of one ConvNet. Each ConvNet takes a pair of aligned face regions as input. Its four convolutional layers (followed by max-pooling) extract the relational features hierarchically. Finally, the extracted features pass a fully connected layer and are fully connected to a single neuron in layer L0 (shown in Figure 2), which indicates whether the two regions belong to the same person. The input region pairs for ConvNets in different groups differ in terms of region ranges and color channels (shown in Figure 4) to make their predictions complementary. When the size of the input regions changes in different groups, the map sizes in the following layers of the ConvNets will change accordingly. Although ConvNets in the same group take the same kind of region pair as input, they are different in that they are trained with different bootstraps of the training data (Section 4.1). Each input region pair generates eight modes by exchanging the two regions and horizontally flipping each region (shown in Figure 5). When the eight modes (shown as M1-M8 in Figure 2) are input to the same ConvNet, eight outputs are generated. Layer L0 contains the outputs of all the 5 × 12 ConvNets and therefore has 8 × 5 × 12 neurons. The purpose of bootstrapping and data augmentation is to achieve robustness of predictions.

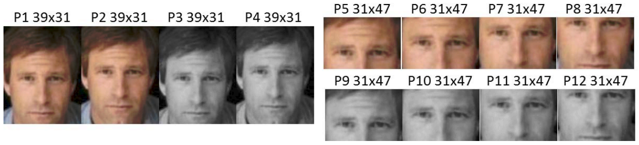

Figure 4: Twelve face regions used in our network. P1 - P4 are global regions covering the whole face, of size 39 × 31. P1 and P2 (P3 and P4) differ slightly in the ranges of regions. P5 - P12 are local regions covering different face parts, of size 31 × 47. P1, P2, and P5 - P8 are in color. P3, P4, and P9 - P12 are in gray values.

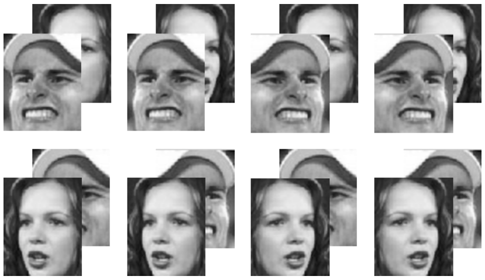

Figure 5: 8 possible modes for a pair of face regions.

The group prediction is given by two levels of average pooling of ConvNet predictions. Layer L1 (with 5 × 12 neurons) is formed by averaging the eight predictions of the same ConvNet from eight different input modes. Layer L2 (with 12 neurons) is formed by averaging the five neurons in L1 associated with the same group. The prediction variance is greatly reduced after average pooling.

The top layer of our model in Figure 2 is a Classification RBM [17]. It merges the 12 group outputs in L2 to give the final prediction. The RBM has two outputs that indicate the probability distribution over the two classes; that is, whether they are the same person. The large number of deep ConvNets means that our model has a high capacity. Directly optimizing the whole network would lead to severe over-fitting. Therefore, we first train each ConvNet separately. Then, by fixing all the ConvNets, the RBM is trained. All the ConvNets and the RBM are trained under supervision with the aim of predicting whether two faces in comparison belong to the same person. These two steps initialize the model to be near a good local minimum. Finally, the whole network is fine-tuned by back-propagating errors from the top-layer RBM to all the lower-layer ConvNets.

3.2. Deep ConvNets

A pair of gray regions forms two input maps of a ConvNet (Figure 5), while a pair of color regions forms six input maps, replacing each gray map with three maps from RGB channels. The input regions are stacked into multiple maps instead of being concatenated to form one map, which enables the ConvNet to model the relations between the two regions from the first convolutional stage.

Our deep ConvNets contain four convolutional layers (followed by max-pooling). The operation in each convolutional layer can be expressed as

$$y_j^r = \max\left(0, b_j^r + \sum_i k_{ij}^r * x_i^r\right), \quad (1)$$

where $*$ denotes convolution, x_i and y_j are the i-th input map and the j-th output map respectively, k_{ij} is the convolution kernel (filter) connecting the i-th input map and the j-th output map, and b_j is the bias for the j-th output map. $\max(0, \cdot)$ is the non-linear activation function, and is operated element-wise. Neurons with such non-linearities are called rectified linear units [15]. Moreover, weights of neurons (including convolution kernels and biases) in the same map in higher convolutional layers are locally shared. r indicates a local region where weights are shared. Since faces are structured objects, locally sharing weights in higher layers allows the network to learn different high-level features at different locations. We find that sharing in this way can significantly improve the fitting and generalization abilities of the network. The idea of locally sharing weights was proposed by Huang *et al.* [13]. However, their model is much shallower than ours and the gained improvement is small.

Since each stage extracts features from all the maps in the previous stage, relations between the two face regions are modeled; see Figure 6 for examples. As the network goes deeper, more global and higher-level relations between the two regions are modeled. These high-level relational features make it possible for the top layer neurons in ConvNets to predict the high-level concept of whether the two input regions come from the same person. The network output is a two-way softmax, $y_i = \frac{\exp(x_i)}{\sum_{j=1}^{2}\exp(x_j)}$ for $i = 1, 2$, where x_i is the total input to an output neuron i, and y_i is its output. It represents a probability distribution over the two classes (being the same person or not). Such a probability distribution makes it valid to directly average multiple ConvNet outputs without scaling. The ConvNets are trained by minimizing $-\log y_t$, where $t \in \{1, 2\}$ denotes the target class. The loss is minimized by stochastic gradient descent, where the gradient is calculated by back-propagation.

3.3. Classification RBM

Classification RBM models the joint distribution between its output neurons y (one out of C classes), input neurons x (binary), and hidden neurons h (binary), as $p(y, x, h) \propto e^{-E(y,x,h)}$, where $E(y, x, h) = -h^\top Wx - h^\top Uy - b^\top x - c^\top h - d^\top y$. Given input x, the conditional probability of its output y can be explicitly expressed as

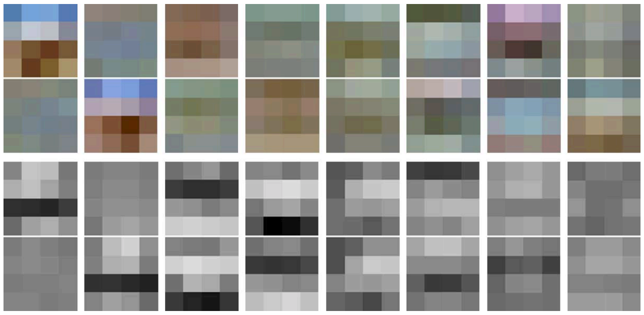

Figure 6: Examples of the learned 4 × 4 filter pairs of the first convolutional layer of ConvNets taking color (line 1) and gray (line 2) input region pairs, respectively. The upper and lower filters in each pair convolve with the two face regions in comparison, respectively, and the results are added. For filter pairs in which one filter varies greatly while the other remains near uniform (column 1, 2), features are extracted from the two input regions separately. For those pairs in which both filters vary greatly, some kind of relations between the two input regions are extracted. Among the latter, some pairs extract simple relations such as addition (column 5) or subtraction (column 6), while others extract more complex relations (column 6, 7). Interestingly, we find that filters in some filter pairs are nearly the same as those in some others, except that the order of the two filters are inversed (columns 1-4). This makes sense since face similarities should be invariant with the order of the two face regions in comparison.

$$p(y_c \mid x) = \frac{e^{d_c}\prod_j\left(1 + e^{c_j + U_{jc} + \sum_k W_{jk}x_k}\right)}{\sum_i e^{d_i}\prod_j\left(1 + e^{c_j + U_{ji} + \sum_k W_{jk}x_k}\right)}, \quad (2)$$

where c indicates the c-th class. We discriminatively train the Classification RBM by minimizing the negative log probability of the target class t given input x; that is, minimizing $-\log p(y_t \mid x)$. The target can be optimized by computing the exact gradient $-\frac{\partial \log p(y_t|x)}{\partial\theta}$, where $\theta \in \{W, U, b, c, d\}$ are RBM parameters to be learned.

3.4. Fine-tuning the entire network

Let N and M be the number of groups and the number of ConvNets in each group, respectively, and $C_m^n(\cdot)$ be the input-output mapping for the m-th ConvNet in the n-th group. Since the two outputs of the ConvNet represent a probability distribution (summed to 1), when one output is known, the other output contains no additional information. So the hybrid model (and the mapping) only keeps the first output from the ConvNet. Let $\{I_k^n\}_{k=1}^K$ be the K possible input modes formed by a pair of face regions of group n.

Then the n-th ConvNet group prediction can be expressed as

$$x_n = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K} C_m^n(I_k^n) , \tag{3}$$

where the inner and outer sums are over different input modes (level 1 pooling) and different ConvNets (level 2 pooling), respectively. Given the N group predictions $\{x_n\}_{n=1}^{N}$, the final prediction by RBM is $\max_{c\in\{1,2\}}\{p(y_c \mid x)\}$, where $p(y_c \mid x)$ is defined in Eq. (2). After separately training each ConvNet and the RBM to derive a good initialization, error is back-propagated from the RBM to all groups of ConvNets and the whole model is fine-tuned. Let $L(x) = -\log p(y_t \mid x)$ be the RBM loss function, and α_m^n be the parameters for the m-th ConvNet in the n-th group. The gradient of the loss w.r.t. α_m^n is

$$\frac{\partial L}{\partial \alpha_m^n} = \frac{\partial L}{\partial x_n}\frac{\partial x_n}{\partial \alpha_m^n} = \frac{1}{MK}\frac{\partial L}{\partial x_n}\sum_{k=1}^{K}\frac{\partial C_m^n(I_k^n)}{\partial \alpha_m^n} . \tag{4}$$

$\frac{\partial L}{\partial x_n}$ can be calculated by the closed form expression of $p(y_t \mid x)$ (Eq. (2)), and $\frac{\partial C_m^n(I_k^n)}{\partial \alpha_m^n}$ can be calculated using the back-propagation algorithm in the ConvNet.

4. Experiments

We evaluate our algorithm on LFW [14], which has been used extensively to evaluate algorithms of face verification in the wild. We conduct evaluation under two different settings: (1) 10-fold cross validation under the unrestricted protocol of LFW without using extra data to train the model, and (2) cross-dataset validation in which external data exclusive to LFW is used for training. The former shows the performance with a limited amount of training data, while the latter shows the generalization ability across different datasets. Section 4.1 explains the experimental settings in detail, section 4.2 validates various aspects of model design, and section 4.3 compares our results with state-of-art results in literature.

4.1. Experiment settings

LFW is divided into 10 folds of mutually exclusive people sets. For the unrestricted setting, performance is evaluated using the 10-fold cross-validation. Each time one fold is used for testing and the other nine for training. Results averaged over the 10 folds are reported. The 600 testing pairs in each fold are predefined by LFW and fixed, whereas training pairs can be generated using the identity information in the other nine folds and the number is not limited. This is referred as the LFW training settings.

For the cross-dataset setting, we use outside data exclusive to LFW for training. PubFig [16] and WDRef [6] are two large datasets other than LFW with faces in the wild. However, PubFig only contains 200 people, thus the identity variation is quite limited, while the images in WDRef are not publicly available. Accordingly, we created a new dataset, called the Celebrity Faces dataset (CelebFaces). It contains $87,628$ face images of $5,436$ celebrities from the web, and was assembled by first collecting the celebrity names that do not exist in LFW to avoid any overlap, then searching for the face images for each name on the web. To conduct cross-dataset testing, the model is trained on CelebFaces and tested on the predefined $6,000$ test pairs in LFW. We will refer to this setting as the CelebFaces training settings.

For both settings, we randomly choose 80% people from the training data to train the deep ConvNets, and use the remaining 20% people to train the top-layer RBM and fine-tune the entire model. The positive training pairs are randomly formed such that on average each face image appears in $k = 6$ (3) positive pairs for LFW (CelebFaces) dataset, unless a person does not have enough training images. Given a fixed number of training images, generating more training pairs provides minimal assistance. Negative training pairs are also randomly generated and their number is the same as the number of positive training pairs. In this way, we generate approximately $40,000$ $(240,000)$ training pairs for the ConvNets and $8,000$ $(50,000)$ training pairs for the RBM and fine-tuning for LFW (CelebFaces) training dataset. This random process for generating training data is repeated for each ConvNet so that multiple different ConvNets are trained in each group.

A separate validation dataset is needed during training to avoid overfitting. After each training epoch[1], we observe the errors on the validation dataset and select the model that provides the lowest validation error. We randomly select 100 people from the training people to generate the validation data. The free parameters in training (the learning rate and its decreasing rate) are selected using view 1 of LFW[2] and are fixed in all the experiments. We report both the average accuracy and the ROC curve. The average accuracy is defined as the percentage of correctly classified face pairs. We assign each face pair to the class with higher probabilities without further learning a threshold for the final classification.

4.2. Investigation on model design

Local weight sharing. Our ConvNets locally share weights in the last two convolutional layers. In the second last convolutional layer, maps are evenly divided into 2×2 regions, and weights are shared among neurons in each region. In the last convolutional layer, weights are independent for each neuron. We compare our ConvNets

[1] One training epoch is a single pass of all the training samples.

[2] View 1 is provided by LFW for algorithm development and parameter selecting without over-fitting the test data. [14].

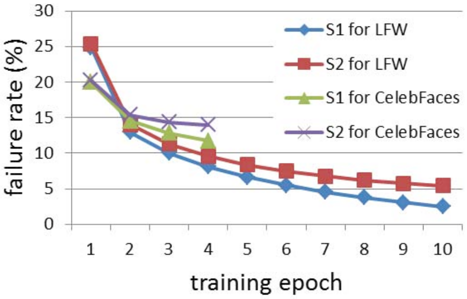

Figure 7: Average training set failure rates with respect to the number of training epochs for ConvNets in group P1 with the local (S1) or global (S2) weight-sharing schemes for the LFW and CelebFaces training settings.

	L0 (%)	L1 (%)	L2 (%)
S1 for LFW	84.78	86.54	88.78
S2 for LFW	83.54	85.28	86.78
S1 for CelebFaces	87.71	88.71	89.60
S2 for CelebFaces	85.65	86.61	87.72

Table 1: Average testing accuracies for ConvNets in group P1 with the local (S1) or global (S2) weight sharing schemes for the LFW and CelebFaces training settings. L0 - L2 refer to the three layers shown in Figure 2. L2 is the final group predictions.

(refer to as S1) with the conventional ConvNets (refer to as S2), where weights in all the convolutional layers are globally shared, on both training errors and test accuracies. Figure 7 and Table 1 show the better fitting and generalization abilities of our ConvNets (S1), where locally sharing weights improved the group P1 (we will refer to each group as the type of regions used (Figure 4)) prediction accuracies by approximately 2% for both the LFW and CelebFaces training settings. The same conclusion holds for ConvNets in other groups.

Two-level average pooling in ConvNet groups. The ConvNet group predictions are derived from two levels of average pooling as described in Section 3.1. Figure 8 shows that the performance is consistently improved after each level of average pooling (from L0 to L2) under the LFW training settings. The accuracy increases over 3% on average after the two levels of pooling (L2 compared to L0). The same conclusion holds for the CelebFaces training settings.

Complementarity of group predictions. We validate that the pooled group predictions are complementary. Given the 12 group predictions (referred as features), we employ a greedy feature selection algorithm. Each time, a feature is added to the feature set, in such a way that the RBM trained on these features provides the highest accuracy on the validation set. The increase of the RBM prediction accuracies would indicate that complementary information is contained in the added features. In this experiment, the ConvNets are pre-trained and their weights are fixed without jointly fine-tuning the whole network. The experiment is repeated five times, with the training samples for the RBM randomly generated each time. The averaged test results are reported. Figure 9 shows that performance is consistently improved when more features are added. So all the group predictions contain additional information.

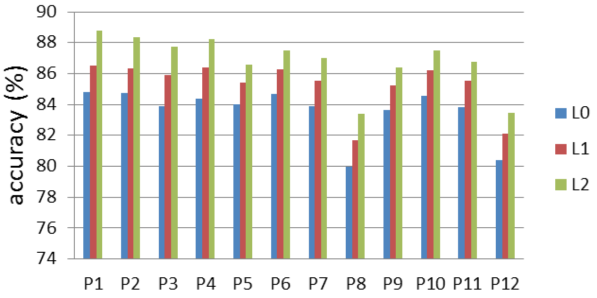

Figure 8: ConvNet prediction accuracies for each group averaged over the 10-fold LFW training settings. L0-L2 refer to the three layers shown in Figure 2.

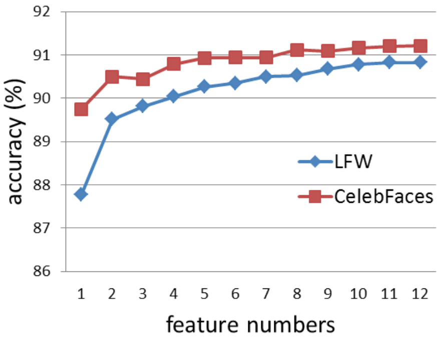

Figure 9: Average RBM prediction accuracies with respect to the number of features selected for the LFW and CelebFaces training settings. The accuracy is consistently improved with the increase of feature numbers.

Top-layer RBM and fine-tuning. Since different groups observe different kinds of regions, each group may be good at judging particular kinds of face pairs differently. Continuing to average group predictions may smooth out the patterns in different group predictions. Instead, we let the top-layer RBM in our model learn such patterns. Then the whole model is fine-tuned to jointly optimize all the parts. Moreover, we find that the performance can be further enhanced by averaging five different hybrid ConvNet-RBM models. This is achieved by first training five RBMs (each with a different set of randomly generated training data) with the weights of ConvNets pre-trained and fixed, and then fine-tuning each of the whole ConvNet-RBM network separately. The results are summarized in Table 2. Interestingly, though directly averaging the 12 group predictions (group averaging) is suboptimal, it

	LFW (%)	CelebFaces (%)
Best single group	88.78	89.70
Group averaging	89.97	90.18
RBM fix	90.93	91.26
Fine-tuning	91.38	92.23
Model averaging	91.75	92.52

Table 2: Accuracies of the best prediction results with a single group (best single group), directly averaging the group predictions (group averaging), training a top layer RBM while fixing the weights of ConvNets (RBM fix), fine-tuning the whole hybrid ConvNet-RBM model (fine-tuning), and averaging the predictions of the five hybrid ConvNet-RBM models (model averaging), for LFW and CelebFaces training settings respectively.

still improves the best prediction results of a single group (best single group). We achieved our best results with the averaging of five hybrid ConvNet-RBM model predictions (model averaging).

4.3. Method comparison

We compare our best results on LFW with the state-of-the-art methods in accuracies (Table 3 and 4) and ROC curves (Figure 10 and 11) respectively. Table 3 and Figure 10 are comparisons of methods that follow the LFW unrestricted protocol without using outside data to train the model. Table 4 and Figure 11 report the results when the training data outside LFW is allowed to use. Methods marked with * are published after the submission of this paper. Our ConvNet-RBM model achieves the third best performance in both settings. Although Tom-vs-Pete [3], high-dim LBP [7], and Fisher vector faces [25] have better accuracy than our method, there are two important factors to be considered. First, all the three methods used stronger alignment than ours: 95 points in [3], 27 points in [7], and 9 points in [25], while we only use three points for alignment. Berg and Belhumeur [3] reported 90.47% accuracy with three point (the eyes and mouth) alignment. Chen et al. [7] reported 6% $\sim$ 7% accuracy drop if use five point alignment and single scale patches. Second, all the three methods used hand-crafted features (SIFT or LBP) as their base features, while we learn features from raw pixels. The base features used in [7] and [25] are densely sampled on landmarks or grids with many different scales and the dimension is particularly high (100K LBP features in [7] and 1.7M SIFT features in [25]).

Method	Accuracy (%)
PLDA [20]	90.07
Joint Bayesian [6]	90.90
Fisher vector faces [25]*	93.03
High-dim LBP [7]*	93.18
ConvNet-RBM	91.75

Table 3: Accuracy comparison of our hybrid ConvNet-RBM model and the state-of-the-art methods under the LFW unrestricted protocol.

Method	Accuracy (%)
Associate-predict [33]	90.57
Joint Bayesian [6]	92.4
Tom-vs-Pete classifiers [3]	93.30
High-dim LBP [7]*	95.17
ConvNet-RBM	92.52

Table 4: Accuracy comparison of our hybrid ConvNet-RBM model and the state-of-the-art methods that rely on outside training data.

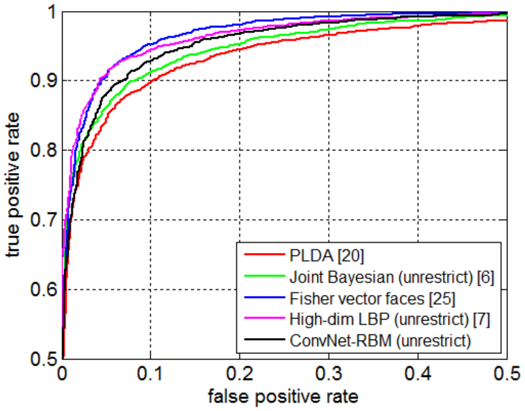

Figure 10: ROC comparison of our hybrid ConvNet-RBM model and the state-of-the-art methods under the LFW unrestricted protocol.

5. Conclusion

This paper has proposed a new hybrid ConvNet-RBM model for face verification. The model learns directly and jointly extracts relational visual features from face pairs under the supervision of face identities. Both feature extraction and recognition stages are unified under a single deep network architecture and all the components are jointly optimized for the target of face verification. It achieved competitive face verification performance on LFW.

6. Acknowledgement

This work is supported by the General Research Fund sponsored by the Research Grants Council of the Kong Kong SAR (Project No. CUHK 416312 and CUHK 416510) and Guangdong Innovative Research Team Program (No.201001D0104648280).

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition.

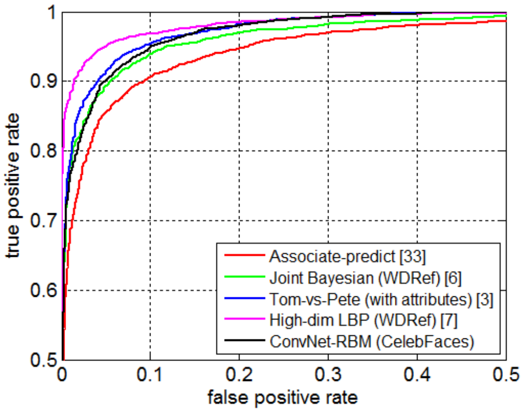

Figure 11: ROC comparison of our hybrid ConvNet-RBM model and the state-of-the-art methods relying on outside training data.

PAMI, 28:2037–2041, 2006. 1

[2] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2:1–127, 2009. 1

[3] T. Berg and P. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *Proc. BMVC*, 2012. 1, 2, 7

[4] T. Berg and P. Belhumeur. Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *Proc. CVPR*, 2013. 1, 2

[5] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Proc. CVPR*, 2010. 1, 2

[6] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Proc. ECCV*, 2012. 1, 2, 5, 7

[7] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proc. CVPR*, 2013. 1, 2, 7

[8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, 2005. 1, 2

[9] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proc. CVPR*, 2012. 2

[10] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Proc. ICCV*, 2009. 1, 2

[11] G. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18:1527–1554, 2006. 2

[12] C. Huang, S. Zhu, and K. Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *NEC Technical Report TR115*, 2011. 1, 2

[13] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *Proc. CVPR*, 2012. 1, 2, 4

[14] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 5

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 2, 4

[16] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proc. ICCV*, 2009. 1, 2, 5

[17] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio. Learning algorithms for the classification restricted boltzmann machine. *JMLR*, 13:643–669, 2012. 3

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2

[19] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proc. ICML*, 2009. 2

[20] P. Li, S. Prince, Y. Fu, U. Mohammed, and J. Elder. Probabilistic models for inference about identity. *PAMI*, 34:144–157, 2012. 1, 2, 7

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 2

[22] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Proc. ACCV*, 2010. 1, 2

[23] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24:971–987, 2002. 2

[24] N. Pinto and D. D. Cox. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In *FG*, 2011. 1, 2

[25] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *Proc. BMVC*, 2013. 1, 2, 7

[26] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013. 2

[27] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proc. BMVC*, 2009. 2

[28] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *Proc. CVPR*, 2004. 1

[29] X. Wang and X. Tang. A unified framework for subspace face recognition. *PAMI*, 26:1222–1228, 2004. 1

[30] X. Wang and X. Tang. Random sampling for subspace face recognition. *IJCV*, 70:91–104, 2006. 1

[31] L. Wiskott, J.-M. Fellous, N. Krger, and C. V. D. Malsburg. Face recognition by elastic bunch graph matching. *PAMI*, 19:775–779, 1997. 2

[32] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop on Faces Real-Life Images at ECCV*, 2008. 2

[33] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *Proc. CVPR*, 2011. 1, 2, 7

[34] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Proc. ICCV*, 2013. 1, 2