Deep Nonlinear Metric Learning with Independent Subspace Analysis for Face Verification

Xinyuan Cai, Chunheng Wang, Baihua Xiao, Xue Chen, Ji Zhou
State Key Laboratory of Management and Control for Complex Systems Institute of Automation
Chinese Academy of Sciences, Beijing, 100190, China
{xinyuan.cai, chunheng.wang, baihua.xiao, xue.chen, ji.zhou}@ia.ac.cn

ABSTRACT

Face verification is the task of determining by analyzing face images, whether a person is who he/she claims to be. It is a very challenge problem, due to large variations in lighting, background, expression, hairstyle and occlusion. The crucial problem is to compute the similarity of two face vectors. Metric learning has provides a viable solution to this problem. Until now, many metric learning algorithms have been proposed, but they are usually limited to learning a linear transformation (i.e. finding a global Mahalanobis metric). In this brief, we propose a nonlinear metric learning method, which learns an explicit mapping from the original space to an optimal subspace, using deep Independent Subspace Analysis network. Compared to kernel methods, which can also learn nonlinear transformations, our method is a deep and local learning architecture, and therefore exhibits more powerful ability to learn the nature of highly variable dataset. We evaluate our method on the LFW benchmark, and results show very comparable performance to the state-of-art methods (achieving 92.28% accuracy), while maintaining simplicity and good generalization ability.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Independent Subspace Analysis, Face verification, Deep Learning Architecture

1. Introduction

There has been a lot of progress made in the area of face recognition and many face recognition systems have been developed. The operation of face recognition systems can be divided into two modes: identification and verification [1]. In the identification mode, the system recognizes the individual by searching the database to find who the individual is or he/she doesn't belong to the database. In the verification mode, the system validates whether the individual is the identity he/she claims to be. In this paper, we concern about verification. In this set-up, pairs of images are given at training time, along with a label indicating whether the pair contains two images of the same person (matched pair), or two images of two different persons (mismatched pair). At test time, a new pair of images is presented, and the task is to assign the ap propriate matched/mismatched label. Unlike other face recogni tion problem formulations, it is not assumed that the person identities in the train and test sets have any overlap, and often the two sets are disjoint.

MM'12, October 29–November 2, 2012, Nara, Japan.

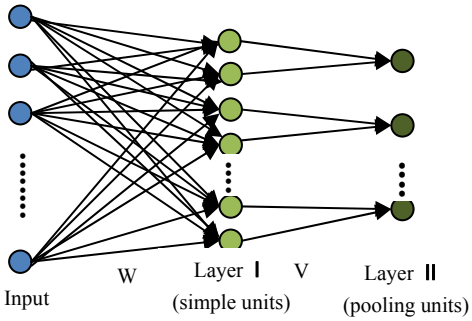

Figure 1: The structure of ISA network.

Face images are highly variable source of multimedia data [2]. Each face image results from the interaction of multiple contributory factors. For instance, one particular face images might be obtained by imaging a certain person (identity), under certain lighting (illumination), from a certain view angle (pose). In face recognition, the only goal is the recognition of identity, regardless of other possible variations. Unfortunately, the variation caused by the changes in illumination, pose, or others could be larger than that caused by identity changes. Traditional methods for frontal face images under some constrained environment may not cover the large variations under unrestricted conditions.

Metric learning has provided a viable solution for unconstrained face verification by comparing the image pairs based on the learned metric. Most metric learning methods attempt to learn an appropriate similarity measure from the labeled side information, which are often available in the form of pairwise constraints, i.e. pairs of similar or dissimilar data points. A common theme in metric learning is to learn a distance metric such that the distance between similar examples should be relatively smaller than that between dissimilar examples [3]. The most commonly used metric is Mahalanobis metric. It is equivalent to first applying a linear transformation, then computing Euclidean distance in the new subspace. But in many situations, a linear transformation often fails to give good performance in high dimensional space, and it is not powerful enough to capture the underlying data manifold. Therefore, we resort to more powerful non-linear transformation. The kernel-based approaches can achieve this goal implicitly, but they behave almost like template-based approaches, so they often have difficulty in handling large datasets.

In this paper, we propose a nonlinear metric learning method using deep Independent Subspace Analysis network (ISA). ISA[4] is a variant of Independent Component Analysis (ICA), and it can be described as a two-layered network (Figure 1), with square and square-root active functions in the first and second layers respectively. In addition, ISA has been used to learn features that achieved state-of-art performance on action recognition tasks[4]. But ISA is an unsupervised learning algorithm, so the learned features may not have enough discriminative power, and may not be suitable for the specific task. We regard the ISA network as a nonlinear function, which transforms the features from the original space to another subspace. And in order to identify good dis-

criminative features, we combine the side information constraints of metric learning with ISA, and stack the ISA networks in deep architecture (Figure 2). We formulate the proposed method as an appropriate optimization problem, and then we employ discriminative pre-training and fine-tuning methods, which are widely used in deep learning, to get the optimal solution. We evaluate the proposed method on the Labeled Faces in the Wild (LFW) [5] benchmark. The result demonstrates superior performance over some state-of-art methods.

The remainders of this paper are structured as follows. In Section 2 we will briefly introduce the traditional metric learning. And then in Section 3, we explain the details of our deep nonlinear metric learning method. Our experiment and result analysis are in Section 4. And the last section has the conclusion.

2. Traditional Supervised Metric Learning

Distance metrics are fundamental concepts in machine learning, and is crucial in real-world application. There has been considerable research on distance metric learning over the past few years [3]. Unlike most supervised learning algorithms where training examples are given class labels, the label information in distance metric learning is usually specified in the form of pairwise constraints on the data [3]: (1) equivalence constraints, which state that the given pairs are semantically-similar and should be close in the learned metric; and (2) inequivalence constraints, which indicate that the given pairs are semantically-dissimilar and should not be near in the learned metric. Most learning algorithms try to find a distance metric that keep all the data pairs in the equivalence constraints close while separating those in the inequivalence constraints. The most representative work is [6], which formulates distance metric learning as a constrained convex programming problem.

Let $\mathrm{C} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{T}\}$ be a collection of data points, where T is the number of samples in the collection. Each $\mathrm{x_i} \in \mathrm{R^n}$ is a data vector where n is the dimension of features. Let the set of equivalence constraints denoted by

$$\mathrm{S} = \{(\mathrm{x_i}, \mathrm{x_j}) | \mathrm{x_i} \text{ and } \mathrm{x_j} \text{ belong to the same class}\}$$

And the set of inequivalence constraints denoted by

$$\mathrm{D} = \{(\mathrm{x_i}, \mathrm{x_j}) | \mathrm{x_i} \text{ and } \mathrm{x_j} \text{ belong to different classes}\}$$

Let the distance metric denoted by matrix $\mathrm{A} \in \mathrm{R^{n \times n}}$, and the distance between any two data points x and y expressed by:

$$\mathrm{d_A^2(x,y)} = ||\mathrm{x-y}||_\mathrm{A}^2 = \mathrm{(x-y)^T A(x-y)} \quad (1)$$

Given the constraints in S and D, [6] formulates the problem of metric learning into the following convex programming problem:

$$\min_{\mathrm{A} \in \mathrm{R^{n \times n}}} \sum_{(\mathrm{x_i}, \mathrm{x_j}) \in \mathrm{S}} \mathrm{d_A^2}(\mathrm{x_i}, \mathrm{x_j}) \quad (2)$$

$$\text{s.t.} \quad \mathrm{A} \geq 0, \ \sum_{(\mathrm{x_i}, \mathrm{x_j}) \in \mathrm{D}} \mathrm{d_A^2}(\mathrm{x_i}, \mathrm{x_j}) \geq 1 \quad (3)$$

The objective term is to make the distance between similar pairs as small as possible. The positive semi-definite constraint $\mathrm{A} \geq 0$, is needed to ensure the negative distance between any two data points and the triangle inequality. The third term is to make the distance between dissimilar pairs at least larger than 1. A is symmetric positive semi-definite, so it can be decomposed as $\mathrm{A} = \mathrm{W^T W}$, and in the learned metric, the distance between any two points can be written as $\mathrm{d_A^2}(x,y) = (x-y)^T A(x-y) = (Wx-Wy)^T(Wx-Wy)$. So the traditional metric learning is equivalent to learn a linear transformation matrix *W*, and then compute Euclidean distance in the transformed subspace.

3. Proposed Framework: Deep Nonlinear Metric Learning with Independent Subspace Analysis

Hinton et al [7] proposed a deep learning architecture called deep belief network (DBN) for auto-encoder neural network. The training of DBN consists of three phases: pretraining, unrolling, and fine-tuning. DBN first trains a sequences of Restricted Boltzman Machines(RBM), and then unrolls the sequence of RBMs to form a deep auto-encoder neural network. Finally, in the fine-tuning phase, the deep auto-encoder neural network is trained using back-propagation algorithms to optimize some task-related objectives [8].

Because of the powerful approximate ability of the deep learning architecture to learn functions or distributions, and the virtues brought by the deep architecture, deep learning methods theoretically exhibit powerful learning ability to discover the nature of the dataset from samples. Deep learning architecture has been successfully employed to enhance the learning ability of existing algorithms [8].

In this work, we also employ the deep learning architecture, but we use ISA network as the basic network. In the following subsections, we will introduce the ISA network, and our discriminative pretraining algorithm—Non-Linear Metric Learning with ISA, and then we stack the pretrained ISA networks for deep metric learning.

3.1 Independent Subspace Analysis

According to [4], ISA is a variant of Independent Component Analysis (ICA), and it can be described as a two layered network (Figure1). The active functions of the first and second layer are square and square root respectively. The connection weight W of the first layer is learned, and the weight V of the second layer is fixed, which represents the subspace structure of the neurons in the first layer. Specifically, each of the second layer hidden units pools over a small neighborhood of adjacent units in the first layer. The first and second layer units are called simple and pooling units respectively.

More precisely, for an input pattern x, the output of each second layer unit is $f_i(x,W,V) = \sqrt{\sum_{j=1}^{k} V_{ij} (\sum_{p=1}^{n} W_{jp} x_p)^2}$. ISA learns the network parameters through finding sparse feature representations in the second layer, by solving:

$$\min_W \sum_{t=1}^{T} \sum_{i=1}^{m} f_i(x^t, W, V) \quad (4)$$

$$s.t. \quad WW^T = I \quad (5)$$

where $\{x^t\}_{t=1}^T$ are input samples. $W \in R^{k \times n}$,$V \in R^{m \times k}$ are the connection weights of the first and second layer. *n,k,m* are the input dimension, number of simple units and pooling units respectively. One property of the ISA pooling units is that they are invariant and thus suitable for recognition task.

3.2 Nonlinear Metric Learning with ISA

We regard the ISA network as an explicit nonlinear transformation function $\mathrm{f(x,W,V)}: \mathrm{R^n} \rightarrow \mathrm{R^m}$, and we use the side information constraints to get the optimal parameters of the ISA network.

Similar to [9], in the transformed subspace, we compute the chi-square distance between data points. And following the idea of [10], we assume a logistic regression model when estimating the probability for any two data points $\mathrm{x_i}$ and $\mathrm{x_j}$ to share the same class or be semantically dissimilar, i.e.,

$$\Pr(\mathrm{l_{i,j}} | \mathrm{x_i}, \mathrm{x_j}) = 1/(1 + \exp(-l_{i,j}(\mathrm{d}(\hat{\mathrm{x}}_\mathrm{i}, \hat{\mathrm{x}}_\mathrm{j}) - \mu))) \quad (6)$$

where $l_{i,j} = \begin{cases} 1 & (\mathrm{x_i}, \mathrm{x_j}) \in \mathrm{S} \\ -1 & (\mathrm{x_i}, \mathrm{x_j}) \in \mathrm{D} \end{cases}$, $\hat{\mathrm{x}}_\mathrm{i} = \mathrm{f}(\mathrm{x_i})$, $\hat{\mathrm{x}}_\mathrm{j} = \mathrm{f}(\mathrm{x_j})$

and $\mathrm{d}(\hat{\mathrm{x}}_\mathrm{i}, \hat{\mathrm{x}}_\mathrm{j}) = \sum_{\mathrm{k=1}}^{\mathrm{m}} (\hat{\mathrm{x}}_\mathrm{i}(\mathrm{k}) - \hat{\mathrm{x}}_\mathrm{j}(\mathrm{k}))^2 / (\hat{\mathrm{x}}_\mathrm{i}(\mathrm{k}) + \hat{\mathrm{x}}_\mathrm{j}(\mathrm{k}))$ (7)

The parameter μ is the threshold. Two data points $\mathrm{x_i}$ and $\mathrm{x_j}$ will have the same class label only when their distance is less than the

threshold μ. Then the overall log likelihood for all the equivalence constraints S and the inequivalence constraints D can be written as

$$\begin{aligned} L_g(W,\mu) &= \log(\Pr(S)) + \log(\Pr(D)) \\ &= -\sum_{(x_i,x_j)\in S} \log\left(1+\exp\left(-d(\hat{x_i},\hat{x_j})+\mu\right)\right) \\ &\quad -\sum_{(x_i,x_j)\in D} \log\left(1+\exp\left(d(\hat{x_i},\hat{x_j})-\mu\right)\right) \end{aligned} \quad (8)$$

Using the maximum likelihood estimation, we will cast the problem of distance metric learning into the following optimization problem:

$$\min_{W,\mu} E = -L_g(W,\mu) + \lambda \sum_{t=1}^{T}\sum_{i=1}^{m} f_i(x^t, W, V) \quad (9)$$

$$s.t. \quad WW^T = I \quad (10)$$

The first term $L_g(W,\mu)$ is the log likelihood of side information constraints, which encourage the margin between positive and negative samples to be large. The second term is the mapping function of ISA, which encourages the sparsity of the transformed features. The hard orthonormality constraints ($WW^T = I$) is used to prevent degenerate solution of *W*. The standard optimization procedure, such as projected gradient descent, can be used to solve the above problem, and W is orthonormalized at each iteration by solving $W := (WW^T)^{-0.5}W$. This symmetric orthonormalization procedure requires Eigen decomposition, which is very challenge and time consuming especially for high dimensional data. However, the side information constraints can also prevent *W* from becoming degenerate. So in order to reduce the computational time, we ignore the hard orthonormality constraints.

In our experiment, we adopt gradient descend method for objective optimization, in which the gradient $\partial E/\partial W$ is computed by back propagation. We write $z_i^{(l)}$, $a_i^{(l)}$ to denote the total weighted sum of inputs and the activation of unit *i* in layer *l* respectively. And we denote $f^{(1)}(x) = x^2$, and $f^{(2)}(x) = \sqrt{x}$ as the active function of the first and second layer of ISA network. Specifically, the computation is given by:

$$z_i^{(1)}(x) = W_i x \qquad a_i^{(1)}(x) = f^{(1)}(z_i^{(1)}(x)) \quad (11)$$

$$z_i^{(2)}(x) = V_i a^{(1)}(x) \qquad a_i^{(2)}(x) = f^{(2)}(z_i^{(2)}(x)) \quad (12)$$

Then the gradient $\partial E/\partial W$ and $\partial E/\partial b$ can be computed as:

$$\begin{aligned} \partial E/\partial W_{jq} = &-\sum_{(x,y)\in S}\left(p_S(x,y)-1\right)\left(\delta_j^{(1)}(x)x_q - \delta_j^{(1)}(y)y_q + \delta_j^{(3)}(x)x_q + \delta_j^{(3)}(y)y_q\right) \\ &-\sum_{(x,y)\in D}\left(p_D(x,y)-1\right)\left(\delta_j^{(1)}(x)x_q - \delta_j^{(1)}(y)y_q + \delta_j^{(3)}(x)x_q + \delta_j^{(3)}(y)y_q\right) \\ &+\lambda\sum_{t=1}^{T}\sum_{i=1}^{m} 1/a_i^{(2)}(x^t)V_{ij}a_j^{(1)}(x^t)x_q^t \end{aligned} \quad (13)$$

$$\partial E/\partial \mu = -\left(\sum_{(x,y)\in S}\left(1-p_S(x,y)\right) + \sum_{(x,y)\in D}(0-p_D(x,y))\right) \quad (14)$$

Where $v_i = a_i^{(1)}(x) + a_i^{(1)}(y)$ $u_i = (a_i^{(1)}(x) - a_i^{(1)}(y))^2$ (15)

$$\delta_j^{(3)}(x) = \sum_{i=1}^{d3} u_i / v_i * z_i^{(2)}(x)^{-0.5} * V_{ij} * z_j^{(1)}(x) \quad (16)$$

$$\delta_j^{(3)}(y) = \sum_{i=1}^{d3} u_i / v_i * z_i^{(2)}(y)^{-0.5} * V_{ij} * z_j^{(1)}(y) \quad (17)$$

$$\delta_i^{(2)}(x) = 1/v_i * 2 * (a_i^{(1)}(x) - a_i^{(1)}(y)) * z_i^{(2)}(x)^{-0.5} \quad (18)$$

$$\delta_i^{(2)}(y) = 1/v_i * 2 * (a_i^{(1)}(x) - a_i^{(1)}(y)) * z_i^{(2)}(y)^{-0.5} \quad (19)$$

$$\delta_j^{(1)}(x) = \sum_{i=1}^{d3} \delta_i^{(2)}(x) * V_{ij} * z_j^{(1)}(x) \quad (20)$$

$$\delta_j^{(1)}(x) = \sum_{i=1}^{d3} \delta_i^{(2)}(y) * V_{ij} * z_j^{(1)}(y) \quad (21)$$

After obtaining the gradient, the parameter *W* and *b* can be updated by (22) until convergence.

$$W = W - \alpha \times (\partial E/\partial W) \quad \mu = \mu - \alpha \times (\partial E/\partial \mu) \quad (22)$$

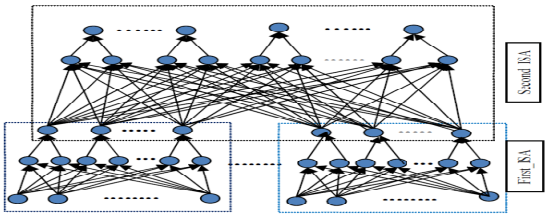

Figure 2: The structure of Stacked ISA network. The outputs of block-wise First_ISA are combined as inputs to the Second_ISA.

3.3 Stacked ISA for Deep Metric Learning

Traditionally, convolution neural network architecture is designed to scale up the algorithm to large inputs. The key idea of this approach is that they first train the pretraining algorithm on small input patches, and then take this learned network to convolve with a larger region of the input image. It is based on the assumption that the distribution over features is stationary in an image with respect to position. However, for images belonging to a specific object class, such as faces, this assumption is no longer true. One strategy for removing this stationary assumption is to connect each basic network to only a local receptive field in the image. In our experiment, we divide the image into a number of non-overlapping block regions, and connect each ISA network to only one block, which we call block-wise ISA network (as in Figure 2). We first train the block-wise ISA networks by the algorithm NLML-ISA, and we regard them as FIRST_ISA networks. Then we combine all the responses of FIRST_ISA network, and treat them as inputs to the next ISA network, which is regarded as SECOND_ISA.

The whole model can be seemed as a Stacked ISA network. Similar to other algorithms proposed in the deep learning literature [7, 8], our Stacked ISA model is trained greedily layerwise in the pretraining phase, but we use the discriminative pretraining algorithm (NLML_ISA). In the fine-tuning phase, the objective function is similar to (9), but the mapping function is a stacked-ISA network. We also adopt the conjugate gradients methods for objective optimization, and the gradient-computing steps are similar to those steps in section 3.2.

4. Experimental Results

In this section, we will numerically compare our algorithm against the state-of-art metric learning algorithms on the LFW [5] benchmark. We implement the proposed method in MATLAB, and the source code is available upon request.

The LFW benchmark is designed for unconstrained face verification. The dataset is challenging and difficult due to large variations in pose, age, expression, race and illumination, as the faces are detected in images in the wild, taken from *Yahoo! News*. It contains13,233 target images of 5749 persons. The database is divided into ten folds where the subject identities are mutually exclusive. There are two evaluation settings by the authors of the LFW: the restricted and unrestricted setting. Under the restricted setting, the identity information of each face is not given. The only available information is a pair of input images which are labeled as the similar or dissimilar pair. Under the unrestricted setting, the identity information of each image is available. So it allows us to generate more image pairs for training. In our experiments, we follow the unrestricted settings, and the performance is measures by ten-fold cross validation.

All face images are cropped to 80*150 pixels just by simply cutting out the center of the aligned version images. To test the robustness of our method for different types of features, we carry out experiments on four descriptors: SIFT[11], Local Binary Pat-

tern (LBP), and two LBP's variations: Hierarchical Local Binary Pattern (HLBP)[12], and Patterns of Oriented Magnitudes (POEM)[13]. We used SIFT features provided by[11], which are computed at the fixed facial key-points (e.g. corners of eyes and nose). For LBP, and its variations, the face image is divided into non-overlapping 5*4 blocks, and then each block is partitioned into 2*2 patches. Histogram features are extracted in each patch, and we concatenate the patch descriptor in one block to form the block-wise descriptors. And in our experiment, we set the number of units for the three layers of ISA network as the dimension of input descriptor, 100, and 50 respectively.

4.1 Effectiveness of DNLML-ISA and Deep Learning

In this experiment, we want to evaluate the effectiveness of our NLML-ISA and the deep learning architecture. For simplicity and fair comparison to LDML and ITML [11], which are state-of-art metric learning methods, we focus on the SIFT descriptor to evaluate our algorithm. We generate 1000 and 2000 pairs per cross-validation fold for training. Table 1 shows the performance of our method in the discriminative pre-training and fine-tuning phases. We can see that the fine-tuning can further improve the performance about 1~2 percents than single layer ISA(First ISA). Table 2 show the comparison between our method, LDML and ITML[11], when using an increasing number of training pairs: 1000, 2000. As expected, all the methods benefit from an increasing number of training pairs. And the performance of our method significantly outperforms LDML and ITML in their best setting.

4.2 Comparison with state-of-art methods

To compare with the state-of-art methods in face verification, we further investigated three types of features for the aligned images: LBP, HLBP [12], and POEM[13]. For each feature, we use both the original and square root feature. And we combine the 8 scores from 4 different descriptors using Support Vector Machine (SVM). Table 3 shows the performance of our method for different types of descriptors when using an increasing number of training pairs: 1000, 2000, and 3000. The highest performance of our method for one single descriptor (SIFT) is (**88.75±0.40**)%, and the result by combining four descriptors is (**92.28±0.42**)%, which is very comparable to the state-of-art methods in Table 4. The best result reported is (92.58±1.36) %, which is a commercial system. It uses high dimensional features (nearly half million), and it considers all possible pairs generated from training samples, while our method just considers 3000 pairs per fold. Finally, the performance of our method may be further improved by exploring different settings of the ISA network, and different types of descriptors.

5. Conclusion

In this paper, we proposed a deep nonlinear metric learning method by using ISA network (DNLML-ISA), to enhance face verification.We use ISA as the basic network, and stack the blockwise ISA networks for large scale inputs. We apply the deep learning with the side information constraints to get the optimal network. With the stacked ISA networks, every instance can be transformed nonlinearly to a compact vector for efficient verification.We evaluate our method on LFW benchmark, and achieve very comparable performance to state-of-art methods. Although initially our DNLML-ISA is designed for face verification, it has a wide range of applications, which we plan to explore in future works.

Table 1 Performance of our method in different phases with SIFT descriptor, when varying the number of training pairs per fold.

	Pre-training		Fine-tuning
	First_ISA	Second_ISA	
1000	85.17±0.45	86.00±0.38	**86.17±0.40**
2000	85.58±0.40	87.25±0.35	**87.67±0.36**

Table 2 Performance comparison of our method, LDML and ITML, when varying the number of training pairs per fold.

	1000	2000
DNLML-ISA	**86.17±0.40**	**87.67±0.36**
LDML[11]	76.30±0.60	78.50±0.50
ITML[11]	77.60±0.30	78.40±0.50

Table 3 Performance of our method with different descriptors, when varying the number of training pairs per fold.

Descriptor		1000	2000	3000
HLBP	Original	83.10±0.40	84.35±0.45	86.67±0.49
	Sqrt_Root	85.25±0.39	86.34±0.40	88.23±0.43
POEM	Original	81.30±0.35	82.52±0.41	84.90±0.50
	Sqrt_Root	82.41±0.45	84.50±0.35	86.30±0.46
SIFT	Original	85.14±0.39	86.10±0.40	87.63±0.45
	Sqrt_Root	**86.17±0.35**	**87.67±0.38**	**88.75±0.40**
LBP	Original	84.25±0.32	85.23±0.41	86.37±0.42
	Sqrt_Root	85.10±0.37	86.30±0.36	87.43±0.40
Feature Combined		**88.50±0.40**	**89.90±0.3**9	**92.28±0.42**

Table 4 Performance comparison of our method and other state-of-art methods in unrestricted configuration.

Method	Accuracy
LDML-MKNN[11]	87.50±0.40
Combined multishot [14]	89.50±0.51
Combined PLDA [15]	90.07±0.51
CMD[16]	91.70±1.10
SLBP[16]	90.00±1.33
CMD+SLBP. Combined[16]	**92.58±1.36**
DNLML-ISA ,combined(this work)	**92.28±0.42**

6. References

[1] A. K. Jain, A. Ross, and S. Prabhakar, An introduction to biometric recognition. IEEE Transactions on circuits and systems for video technology, 14(1): 4–20, 2004.

[2] Y. Zhang and Z.-H.Zhou. Cost-sensitive face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(10): 1758-1769, 2010

[3] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2007.

[4] Q.V.Le, W.Zou, S.Y.Yeung, and A.Y.Ng. Learning hierarchical spatial temporal features for action recognition with independent subspace analysis. In *CVPR* 2011.

[5] G.B.Huang, M.Ramesh, T.Berg and E.L.Miller. Labeled faces in the wild: a database for studying face recognition in unconstrained environment. Technical Report. 2007.

[6] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. in Proc. NIPS, 2003

[7] G.Hinton and R.Salakhutdinov. Reducing the dimensionality of data with neural networks. Science,313(5786): 504-507, 2006

[8] W.K.Wong and M.Sun. Deep learning regularized fisher mappings. IEEE Transactions on neural networks, 22(10): 1668-1675, 2011

[9] X.-Y.Cai, B.-H.Xiao,C.-H.Wang. Quadratic-chi similarity metric learning for histogram feature. In ACPR 2011

[10] G. Lebanon. Flexible metric nearest neighbor classification. In *Proc.* Uncertainty in Artificial Intelligence, 2003

[11] M.Guillaumin, J.Verbeek, and C.Schmid. Is that you? Metric learning approaches for face identification. In *ICCV* 2009.

[12] Z.Guo, L.Zhang, D.Zhang, and X.Mou. Hierarchical multiscale LBP for face and palmprint recognition. In *ICIP* 2010.

[13] N.Vu and A.Caplier. Enhanced pattern of oriented edge magnitudes for face recognition and image matching. *IEEE TIP*, 21(3):1352-1365, 2012

[14] Y.Taigman, L.Wolf, and T.Hassner. Multiple one shots for utilizing class label information. In BMVC 2009

[15] P.Li,Y.Fu,U.Mohammed,J.H.Elder and S.Prince. Probabilistic Models for inference about identity. IEEE TPAMI 34(1), 144-157, 2012.

[16] C.Huang, S.Zhu,and K.Yu. Large scale strongly supervised ensemble metric learning, with application to face verification and retrieval. NEC Technical Report, 2011