# Discriminant Metric Learning Approach for Face Verification

**Ju-Chin Chen[1]*, Pei-Hsun Wu[2], and Jenn-Jier James Lien[2]**
[1] Department of Computer Science and Information Engineering
National Kaohsiung University of Applied Sciences, Kaohsiung, Kaohsiung, Taiwan, ROC
[e-mail: jc.chen@cc.kuas.edu.tw]
[2] Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan, ROC
[e-mail: jjlien@csie.ncku.edu.tw]
*Corresponding author: Ju-Chin Chen

*Received September 3, 2014; revised November 12, 2014; accepted December 13, 2014; published February 28, 2015*

## Abstract

In this study, we propose a distance metric learning approach called discriminant metric learning (DML) for face verification, which addresses a binary-class problem for classifying whether or not two input images are of the same subject. The critical issue for solving this problem is determining the method to be used for measuring the distance between two images. Among various methods, the large margin nearest neighbor (LMNN) method is a state-of-the-art algorithm. However, to compensate the LMNN's entangled data distribution due to high levels of appearance variations in unconstrained environments, DML's goal is to penalize violations of the negative pair distance relationship, i.e., the images with different labels, while being integrated with LMNN to model the distance relation between positive pairs, i.e., the images with the same label. The likelihoods of the input images, estimated using DML and LMNN metrics, are then weighted and combined for further analysis. Additionally, rather than using the *k*-nearest neighbor (*k*-NN) classification mechanism, we propose a verification mechanism that measures the correlation of the class label distribution of neighbors to reduce the false negative rate of positive pairs. From the experimental results, we see that DML can modify the relation of negative pairs in the original LMNN space and compensate for LMNN's performance on faces with large variances, such as pose and expression.

***Keywords:*** Metric learning, face verification, k-nearest neighbor

**http://dx.doi.org/10.3837/tiis.2015.02.015** **ISSN : 1976-7277**

## 1. Introduction

Face recognition is an active research issue in the field of computer vision and has been studied for more than two decades [1]-[10]. It has a wide range of practical applications, including surveillance and border-control systems. For a traditional access system, users can enter a site using keys or integrated circuit (IC) cards; however, security is doubtful because these keys can be easily pirated. Recently, bioinformatics has become popular for access systems, and popular approaches include recognizing fingerprints and palm prints. However, these methods are inconvenient due to the required contact process. In contrast, the non-contact process of using the human face to convey a subject's identity as an access key is more attractive.

Not exclusive to the private security field, face recognition/verification also plays an important role in public security. We are surrounded by surveillance systems and every crossroad is equipped with cameras to record every moment. Consider the occurrence of an urgent event; for example, the police pursuing a criminal and attempting to determine his movements. Checking every frame in all camera records is manually impossible. A more efficient technique would be a face verification system that could filter results and present frames of possible suspects for further analysis. However, the development of robust face verification is challenging since facial images contain an immense variety of expressions, orientations, lighting conditions, occlusions, and so forth. In recent years, researchers have focused on raising the accuracy rate with respect to these variations, including the design of good features for face representation[4], [6], [9], [11], [12] and distance metric learning [10], [11], [13].

Due to its wide use in security applications, we focus our efforts in this study on the problem of face verification to determine whether a pair of facial images is the same or a different person. The fundamental problem is how to measure the similarity between facial examples. A surge of recent research [13-20] has focused on Mahalanobis metric learning to improve the performance of k-nearest neighbor (*k*-NN) classifications. In an uncontrolled environment, it is assumed that facial images can only be detected by a face detector [21], and thus a high degree of variances result, due to lighting conditions, poses, occlusions, and background clutter that make verification challenging. To tackle the highly entangled data distributions caused by the above factors, we propose a distance metric algorithm that can heavily penalize violations in the distance relationship for between-class data while preserve those remaining within-class. In addition, we propose a validation approach that measures the correlation between the label distributions of neighbors to improve the true positive rate.

## 2. Related Work

Face recognition has been an active research topic for more than two decades because of its practical applications. Of particular note, face verification has attracted more attention in recent years, which latter aims to verify if an input image is that of the subject. This differs from face identification in that the test subject(s) in the input pair need not be included in the training dataset. However, in practical applications, the uncontrolled environment causes problems in terms of the immense variations in facial pose, expression, lighting conditions, occlusion, and so on, and the reliability of previous research developed with controlled settings are thus limited. In 2009, Kumar et al. [9] analyzed these failure cases and found that these mistakes would be avoidable if more facial attributes can be analyzed separately before classification. The authors proposed two methods for facial verification in uncontrolled setting. The first method used high-level face representation to recognize the presence or absence of 65 attributes, such as a round face, gender, and so on. The second method was based on a simile classifier and was aimed at recognizing the similarities between the facial regions of the test image pairs with an extra identity dataset as prior knowledge. 60 people were used in the study, which was a significant improvement for the LFW dataset [22]. However, this approach has to define a number of reliable and relevant features [5]. Inspired by the results detailed in [9], many alternative approaches addressed unconstrained face recognition/verification via robust feature learning [11], [23], [24-26], or the similarity measures between feature descriptors [14], [27].

For feature extraction, texture-based local features are applied to face recognition/verification, including LBP [28], SIFT [29], and Gabor [30], [31]. In 2004, Ahonen et al. [4] investigated the LBP feature, which encodes the relationship intensity between each pixel and its neighboring pixels, and the authors found that this approach yielded good results. LBP can be insensitive to lighting changes and provides promising results when compared to global feature and other texture-based approaches [4]. Further modifications of the LBP approach have since been proposed [32-34]. Rather than using one specific feature, the strategy of combining multiple features has been applied to face verification [35], [36]. In [35], the authors combined multiple texture-based features in the score-level fusion and shown that such a combination can provide better verification results than the use of one specific feature by approximately 5.7% on average. Color information is another important feature and the integration of color information can improve recognition performance compared to methods relying solely on color or texture information [37-39]. In [36], the authors proposed new features including color local Gabor wavelets (CLGWs) and color local binary patterns (CLBP), to combine texture and color features. The use of texture and color information remains however an open problem [36]. Instead of designing handcrafted encoding methods, the approaches of [6] and [40] applied learning frameworks to select the discriminant features in order to avoid the difficulties associated to obtaining optimal encoding methods manually. In [6], an unsupervised learning-based method was proposed to encode the local micro-structures of a face into a set of more uniformly distributed discrete codes. Middle-level features using unsupervised feature learning with deep network architectures [11], [24-26] have also been applied to face verification. Further information regarding this study can be found in [41] and [42].

After obtaining the feature descriptors, the subsequent process for face verification is to measure the similarity between the two descriptors. Inspired by the idea of “One-Shot Learning” techniques [43], [44], Wolf et al. [45] proposed the one-shot similarity (OSS) approach to classify a pair of test images via a discriminant model learned from a single positive sample, and a set of prior background negative samples to solve the problem of limited positive samples. Following this, in [46] the authors extended the OSS approach by combining multiple OSS scores to improve the recognition rate and further considering the ranking results of the query image to propose the “Two-Shot Similarity” (TSS) approach [35]. Although the discriminative models are produced as per the vectors being compared [35], [45], [46] and are often better suited to comparing the test pair, two classifiers need to be trained each time when two images are compared. Depending on the classifiers used in the implementation, this may lead to additional computational cost in the test process. With the aim of discovering recognition capability for two faces in an uncontrolled environment, Yin et al. [10] proposed an “Associate-Predict” (AP) model for face recognition based on the conjecture that the transition process is performed with given prior knowledge in a person's brain. The model is built on a prior identity dataset, which differs from the extra unlabeled datasets of [9], [35], [45], [46], where each identity has multiple facial images with large intra-class variations. When a pair of test images is compared, the input face with a number of the most similar identities from the identity dataset is first associated, and the new appearance of the input face in different settings is predicted using appearance-prediction matching. In addition, one person-specific classifier is trained for likelihood-prediction matching. The accuracy of facial component extraction and the selection of the correct identity for appearance prediction have influenced the performance of the association-prediction model. Different from [9] of using the global unlabeled dataset, the authors built the person-specific model on a prior identity dataset to classify the input face against the most similar faces to improve recognition. As in [46], [35], the use of the on-line classifier may have additional computational cost.

In spite of the new frameworks that have been proposed for face verification [9], [35], [45], [46], the similarity measure between facial descriptors is the core of these research. The information theoretic metric learning (ITML) [16] approach for example is applied for each OSS score [46]. In order to tackle the highly entangled data distribution captured in the uncontrolled environment [47], the *k*-NN classifier is the simplest non-linear classifier that is most often applied on the basis of the Euclidean distance metric for recognition. However, the Euclidean distance metric ignores the statistical properties of data that might be estimated from a large training set of labeled examples [13]. Several other distance metric algorithms [14-20] have also been proposed to obtain a new distance metric to investigate data properties from class labels. In [27], cosine similarity metric learning (CSML) was proposed to learn a transformation matrix by measuring the cosine similarity between an image pair. In addition, the Mahalanobis distance metric learned based on the various objective functions. Relevant component analysis (RCA) [15] is intermediate between the unsupervised method of PCA and supervised methods of LDA using the chunklet information, a subset of a class, to learn a full-ranked Mahalanobis distance metric. Unlike LDA, since between-class information is not explicitly imposed in the objective function, the improvement for the *k*-NN based classification with the RCA metric is limited. Similar to the goal of LDA of minimizing the distance between for within-classes whilst maximizing the distance for between-classes [48], Xing *et al.* [19] proposed a Mahalanobis metric for clustering (MMC) with side information that represented the first convex objective function for distance metric learning. Because MMC was built with the normal or unimodel assumption for clusters, it is not particularly appropriate for *k*-NN classifiers [13]. In contrast to RCA and MMC, the large margin nearest neighbor (LMNN) classification [13] is the first method for imposing a constraint on the distance metric for the *k*-NN classification. Thus, via the metric, the *k*-nearest neighbors always belong to the same class, while examples from different classes are separated by a large margin. A series of experiments have been conducted to prove that the LMNN approach yields better results than PCA, LDA, RCA, and MMC [13]. In order to extend the LMNN approach to the binary-classes problem for face verification, Guillaumin *et al.* [14] proposed a logistic discriminant-based metric learning (LDML) to modify LMNN constraints with a probability formulation by learning a metric from a set of labeled image pairs. LDML can provide better results than [16] in the binary-classes problem. In addition, the comparison mechanism, a marginalized *k*-NN classifier (MkNN), was also proposed to verify a test pair by a set of labeled images. However, an incorrect classification results when two images of the same subject receive a low similarity rating if the class labels for their corresponding *k* nearest neighbors are uniformly distributed.

## 3. Overview of the Proposed Face Verification System

**Fig. 1** shows an overview of the training and test process in the proposed face verification system. The training process commences by detecting faces from training images and normalizing face geometry according to the locations of eyes. By considering the spatial information of the face, the face image is divided into $3 \times 3$ regions. Then the 59-dimensional feature of the local binary pattern is extracted from each region and the features are further concatenated into one 531-dimensional feature vector $f_i$. To develop a discriminant metric for verification, $N_p$ positive pairs (two images of the same subject) and $N_n$ negative pairs (two images of different subjects) are generated from the training dataset. Then, these training pairs are used to learn the distance metric $M_{Dis}$, which is composed of two ideas, $M_{LMNN}$ and $M_{DML}$. The distance relationship of the positive pairs is learned via the distance metric $M_{LMNN}$ [13], and the distance relationship of the negative pairs is learned via our proposed metric $M_{DML}$. Hence we can minimize not only the within-class distance, but we can maximize the between-class distance. Note that violations of the distance relationship for the negative pairs is heavily penalized via $M_{DML}$ to reduce the false positive rate for unconstrained verification.

In the test process, a test pair of two facial images is input and the LBP features are extracted for each test image as in the training process. Then the similarity between each test image and the training images are evaluated based on the trained distance metrics $M_{LMNN}$ and $M_{DML}$, respectively. The proposed verification mechanism, correlation of the $k$-nearest neighbor (CkNN), constructs the corresponding $k$-NN code for each test image and then measures the correlation between two $k$-NN codes. This measurement is then applied to decide whether the test pair are the same subject or not.

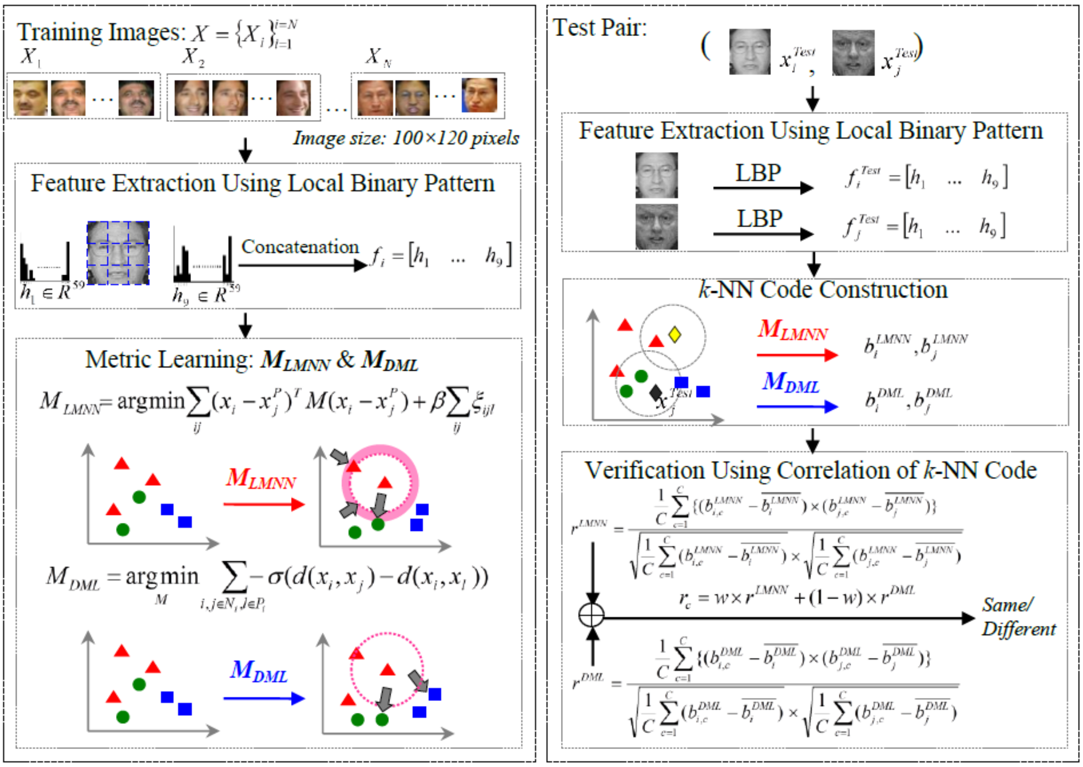

Training Images: $X = \{X_i\}_{i=1}^{i=N}$
$X_1$
$X_2$
$X_N$
Image size: 100×120 pixels
Feature Extraction Using Local Binary Pattern
$h_1 \in R^{59}$
$h_9 \in R^{59}$
Concatenation
$f_i = [h_1 \ \dots \ h_9]$
Metric Learning: $M_{LMNN}$ & $M_{DML}$
$M_{LMNN} = \arg\min \sum_{ij} (x_i - x_j^P)^T M (x_i - x_j^P) + \beta \sum_{ij} \xi_{ijl}$
$M_{LMNN}$
$M_{DML} = \arg\min_M \sum_{i,j \in N_i, l \in P_i} -\sigma(d(x_i,x_j) - d(x_i,x_l))$
$M_{DML}$
Test Pair:
$(x_i^{Test}, x_j^{Test})$
Feature Extraction Using Local Binary Pattern
LBP
$f_i^{Test} = [h_1 \ \dots \ h_9]$
LBP
$f_j^{Test} = [h_1 \ \dots \ h_9]$
$k$-NN Code Construction
$x^{Test}$
$M_{LMNN}$
$b_i^{LMNN}, b_j^{LMNN}$
$M_{DML}$
$b_i^{DML}, b_j^{DML}$
Verification Using Correlation of $k$-NN Code
$r^{LMNN} = \frac{\frac{1}{C}\sum_{c=1}^{C}\{(b_{i,c}^{LMNN} - \overline{b_i^{LMNN}}) \times (b_{j,c}^{LMNN} - \overline{b_j^{LMNN}})\}}{\sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{i,c}^{LMNN} - \overline{b_i^{LMNN}})} \times \sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{j,c}^{LMNN} - \overline{b_j^{LMNN}})}}$
$r_c = w \times r^{LMNN} + (1-w) \times r^{DML}$
Same/ Different
$r^{DML} = \frac{\frac{1}{C}\sum_{c=1}^{C}\{(b_{i,c}^{DML} - \overline{b_i^{DML}}) \times (b_{j,c}^{DML} - \overline{b_j^{DML}})\}}{\sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{i,c}^{DML} - \overline{b_i^{DML}})} \times \sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{j,c}^{DML} - \overline{b_j^{DML}})}}$

**Fig. 1.** Flowchart of the proposed face verification system. (a) The training process (b) The test process

## 4. Face Verification System

In this section, we discuss the details of the proposed distance metric and the verification mechanism. First, we introduce the extracted features, and then the design concept for the proposed distance metric and the optimization process. Lastly, we show the coding and verification mechanism, namely CkNN. In the following discussion, the training data set is composed of *N* subjects with $n_i$ images, denoted as $X = \{X_i\}_{i=1}^{i=N}$ , and the size of each image is $w \times h$ .

### 4.1 Feature extraction by LBP

The LBP is a kind of texture-based feature that has been demonstrated to perform face recognition very well [28]. Its mechanism is the use of binary codes to present the intensity (gray-value) relationship between the processing pixel and its surrounding pixels. For each processing pixel these binary codes are then transformed into a decimal value; and then the statistical distribution of the decimal values from all pixels are represented by a histogram as the facial feature vector. **Fig. 2** shows examples of giving an image a gray value according to its corresponding decimal value. We can see that the dark pixels correspond to those facial components and facial contours. Note that in [28], binary codes are further divided into uniform and non-uniform patterns. A uniform pattern is one with the changes between binary codes, i.e., 0 to 1 or 1 to 0, occurring fewer than two times, and the remaining patterns are designated as non-uniform patterns. From **Fig. 3**, we can see that the uniform patterns can capture the local important features such as corners and edges, and hence they are recorded in one specific bin in the histogram, and all of the non-uniform patterns are recorded in one bin. The resulting facial image can be represented by a 59-dimensional histogram.

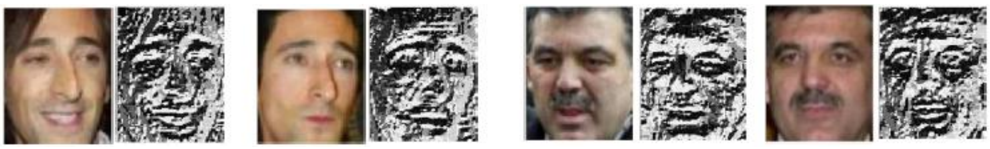


**Fig. 2.** Coding results of a local binary pattern

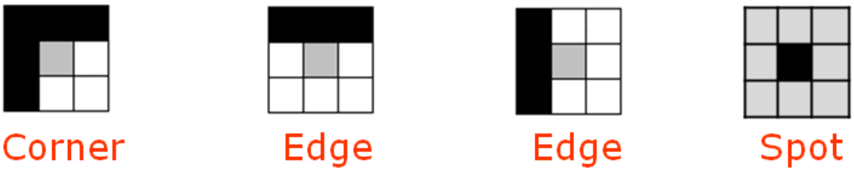


**Fig. 3.** Examples of a uniform pattern corresponding to locally important facial features

The usage of a statistical histogram as a feature representation is popular [49]. One advantage is that it can overcome rotation variance, but the disadvantage is the loss of spatial information. When the geometry is important, the damage is obvious. In order to cope with this situation, one way to maintain the geometric relationship is to divide the object (face) into multiple regions [28]. Then one histogram can be used to represent each region, and the final feature vector can be obtained by concatenating all histograms. In our work, therefore, we divide one facial image into $3 \times 3$ regions, each of which is represented by a 59-dimensional feature vector. In the end, 9 histograms are concatenated into one 531-dimensional feature vector of the data for each training face.

## 4.2 Distance Metric Learning

LMNN [13] is one of the state-of-the-art metrics designed for the Mahalanobis distance measure which can reduce within-class distance and enlarge between-class distance. Hence, using this metric, the *k*-NN classifier can benefit from these modified distance relationships. However, the distribution of unconstrained facial data for the same class is highly non-linear, and even they are entangled for different classes [47]. Hence, we use LMNN to minimize within-class distance and discriminant metric distance metric learning (DML), which is designed to penalize violations of between-class distance relationships.

### 4.2.1 Large-Margin Nearest Neighbour Metric Learning

LMNN metric learning [13] derives a metric favored by the *k*-NN classifier to calculate the Mahalanobis distance between data values $x_i$ and $x_j$ via matrix $M_{LMNN}$ as

$$d(x_i, x_j) = (x_i - x_j)^T M_{LMNN} (x_i - x_j) \tag{1}$$

If the matrix is degenerated into one identity matrix, Eq. (1) can estimate the Euclidean distance between $x_i$ and $x_j$. The idea of LMNN, as shown in **Fig. 4**, is to minimize the within-class distances (the distance between the blue squares) while maximizing the between-class distances larger than one unit (the distance between blue squares and black triangles). In other words, for each data value $x_i$, the main object is to minimize the distance for the positive pairs ($x_i$, $x_j^P$) where $x_j^P$ is one target neighbor [13], i.e., one of the k nearest neighbors having the same class label as $x_i$, while maximizing the distance for the negative pairs ($x_i$, $x_l^N$) where $x_l^N$ is one of the *k*-NNs having a different class label for $x_i$. Thus the objective function can be derived as in [13]:

$$\begin{aligned} & M_{LMNN} = \arg\min \sum_{ij} (x_i - x_j^P)^T M (x_i - x_j^P) + \beta \sum_{ij} \xi_{ijl} \\ & s.t.\ \ \xi_{ijl} \geq 0, \quad M \succeq 0 \\ & (x_i - x_l^N)^T M (x_i - x_l^N) - (x_i - x_j^P)^T M (x_i - x_j^P) \geq 1 - \xi_{ijl} \end{aligned} \tag{2}$$

where $M_{LMNN}$ is a semi-definite matrix, i.e., all the eigenvalues are equal or larger than 0, $\beta$ is the parameter that controls the importance of the within-class distance and the between-class distance, and ξ is a slack variable to penalize violations of distance conditions between ($x_i$, $x_j^P$) and ($x_i$, $x_l^N$). In Eq. (2), the first term minimizes the distance of the positive pairs, and the second term uses ξ to maintain the distance of the negative pairs as greater than the distance of a positive pair within one unit. More details about the optimization process can be referenced in [13].

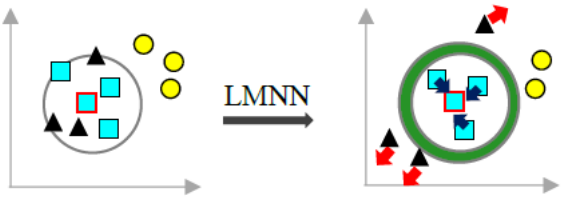


**Fig. 4.** Schematic illustration of LMNN. Each data value is represented by an icon, with three classes (square, triangle, and circle) shown. The left figure shows the data relationship before using the LMNN metric. After using the LMNN metric, the right figure shows the modified data relationship for $x_i$. with its neighbors, and specifically the within-class distances. The distance between blue squares is minimized while the between-class distances, shown as distance between the blue squares and black triangles, are maximized.

### 4.2.2 Discriminant Metric Learning

In order to tackle entangled data distributions to reduce the false positive rate, one discriminant metric is designed to enhance the penalization for negative pairs violating the distance relationship, i.e., where the between-class distance is smaller than within-class distance in a unit save range. **Fig. 5** shows a schematic illustration of the discriminant metric wherein if the distance of the negative pairs, i.e., the distance between the processing of data $x_i$ and its neighbors having different class labels (as shown in black triangles), is smaller than the distance of the positive pairs, a larger cost is assigned. On the other hand, if the distance between negative pairs is larger than the distance between positive pairs, a smaller cost is assigned. Hence, via the sigmoid function, which ranges from 0 to 1, the objective function for the discriminant metric is designed as follows:

$$M_{DML} = \arg\min_{M} \sum_{i,l\in N_i, j\in P_i} -\sigma(d(x_i,x_l)-d(x_i,x_j)) \tag{3}$$

where $\sigma(z)=\dfrac{1}{1+\exp(-z)}$ is the sigmoid function, $P_i$ and $N_i$ are $x_i$'s neighbor sets whose class labels are the same as $x_i$ or not, respectively, and $d(\bullet)$ is the Mahalanobis distance between $x_i$ and $x_j$ (or $x_l$), which is measured with the distance metric *M*. When the parameter z is larger, the function value $\sigma(z)$ is closer to 1, while if z is smaller, $\sigma(z)$ is closer to 0. In contrast to Eq.(1) , the sigmoid function is a convex function and a first-order derivative. Thus the optimal value for $M_{DML}$ can be obtained by taking the derivation for *M* as

$$\begin{aligned}
\frac{\partial f}{\partial M} &= \frac{-1}{1+\exp(d_P-d_N)}\frac{1}{\partial M} \\
&= \frac{-\exp(d_N)}{\exp(d_P)+\exp(d_N)}\frac{1}{\partial M} \\
&= \frac{-\exp(d_N)X_N}{\exp(d_P)+\exp(d_N)} + \frac{\exp(d_N)\{X_N\exp(d_N)+X_P(\exp(d_P))\}}{\{\exp(d_P)+\exp(d_N)\}^2} \\
&= c_2X_P - c_1X_N
\end{aligned} \tag{4}$$

where $c_1 = [\frac{\exp(d_N)}{\exp(d_P)+\exp(d_N)}]^2 - \frac{\exp(d_N)}{\exp(d_P)+\exp(d_N)}$, $c_2 = \frac{\exp(d_P)\exp(d_N)}{[\exp(d_P)+\exp(d_N)]^2}$, $X_P = (x_i - x_j)(x_i - x_j)^T$, $X_N = (x_i - x_l)(x_i - x_l)^T$, and $d_P$ and $d_N$ are the distances of the positive and negative pairs, respectively. Using the gradient descent method and setting the initial value as an identity matrix, the optimal value of $M_{DML}$ can be obtained. According to the optimization process, the time taken for each iteration includes: 1) the distance calculation time for training examples, and 2) the distance metric updating time, where the computational complexities are O($nd^2+n^2d$) and O($knd^2$) respectively, in which $n$ is the number of training data, $k$ is the number of neighbors, and $d$ is the dimension of LBP feature. Because the distance calculation of the training data is independent to each other, this step can be parallelized in distributed computing framework like MapReduce to speed up the offline processes. In such framework, each computing node can serve a part of the training samples to accelerate the training process time for DML, and therefore the execution of the offline processes can be fast.

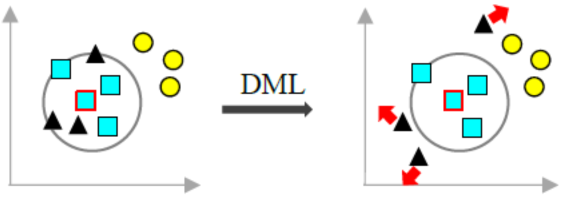


**Fig. 5.** Schematic illustration for DML.The left figure shows the data relationship before using the DML metric. After applying the DML metric, the right figure shows the modified data relationship for $x_i$ with its neighbors that have a different class label from $x_i$. Specifically, the between-class distances denoted as black triangles are maximized.

### 4.3 Correlation of the *k*-NN Code

For verification, the output predicts whether a pair of images belongs to the same class. In [14], the authors considered the neighbor's class labels and proposed the MkNN [14] to measure the label distribution similarity between neighbors. However, when the data distribution is heavily entangled in the transformed metric space and this leads to two images of the same subject surrounded by data from different subjects, the worst case is that k neighbors are from different k subjects, and the classification might be wrong due to the low distribution probability.

With MkNN, instead of estimating only the data distance via Eqs. (2) or (3) to predict whether they are the same class or not, their corresponding neighbors' class information is considered. After learning the distance metrics $M_{LMNN}$ and $M_{DML}$, during the verification process each data value $x_i$ is extracted from the local binary patterns and then the distances from the training images based on $M_{LMNN}$ and $M_{DML}$ are measured to obtain the corresponding k nearest neighbors, denoted as $S_i^{LMNN}$ and $S_i^{DML}$, respectively. Then the *k*-NN code $b_i^{LMNN}$ and $b_i^{DML}$ for $x_i$ is defined as follows:

$$b_{i,j}^{LMNN} = \sum_{k=1}^{K} \delta(y(S_{i,k}^{LMNN}) - j) \tag{5}$$

and

$$b_{i,j'}^{DML} = \sum_{k=1}^{K} \delta(y(S_{i,k}^{DML}) - j') \tag{6}$$

where the dimensions of $b_i^{LMNN}$ and $b_i^{DML}$ are the number of classes, $\delta(\bullet)$ is an indicator function, K is the number of nearest neighbors, $S_{i,k}^{LMNN}$ and $S_{i,k}^{DML}$ are the *k*-th nearest neighbor for $x_i$ measured by $M_{LMNN}$ and $M_{DML}$, respectively, and the corresponding labels are denoted as $y(S_{i,k}^{LMNN})$ and $y(S_{i,k}^{DML})$. In other words, the *k*-NN code contains the class label distribution of neighbors surrounding $x_i$. **Fig. 6** shows an example and the corresponding *k*-NN code $b_i = [3, \quad 2, \quad 0, \quad 2]^T$.

After obtaining the *k*-NN code for each image of the test image pair $(x_i, x_j)$, verification is performed by computing the correlation coefficients of *k*-NN codes by

$$r^{LMNN} = \frac{\frac{1}{C}\sum_{c=1}^{C}\{(b_{i,c}^{LMNN} - \overline{b_i^{LMNN}}) \times (b_{j,c}^{LMNN} - \overline{b_j^{LMNN}})\}}{\sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{i,c}^{LMNN} - \overline{b_i^{LMNN}})} \times \sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{j,c}^{LMNN} - \overline{b_j^{LMNN}})}} \tag{7}$$

and

$$r^{DML} = \frac{\frac{1}{C}\sum_{c=1}^{C}\{(b_{i,c}^{DML} - \overline{b_i^{DML}}) \times (b_{j,c}^{DML} - \overline{b_j^{DML}})\}}{\sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{i,c}^{DML} - \overline{b_i^{DML}})} \times \sqrt{\frac{1}{C}\sum_{c=1}^{C}(b_{j,c}^{DML} - \overline{b_j^{DML}})}} \tag{8}$$

where $b_{i,c}^{LMNN}$ ,$b_{i,c}^{DML}$ , $b_{j,c}^{LMNN}$ , and $b_{j,c}^{DML}$ are the *c*-th elements in the corresponding *k*-NN code $b_i^{LMNN}$ and $b_i^{DML}$, $b_j^{LMNN}$, and $b_j^{DML}$, respectively, and $\overline{b_i^{LMNN}}$, $\overline{b_i^{DML}}$, $\overline{b_j^{LMNN}}$, and $\overline{b_j^{DML}}$ are the corresponding means. The final verification result $V(x_i, x_j)$ is defined as

$$V(x_i, x_j) = \begin{cases} 1, & if \;\; r_c > \gamma \\ 0, & otherwise \end{cases} \tag{9}$$

where $\gamma$ is the threshold. $V(x_i, x_j) = 1$ indicates that the test image pair $(x_i, x_j)$ are the same subject, else they are different subjects, and $r_c$ is a weighted similarity of $r^{LMNN}$ and $r^{DML}$ with the coefficient w given by

$$r_c = w \times r^{LMNN} + (1 - w) \times r^{DML} \tag{10}$$

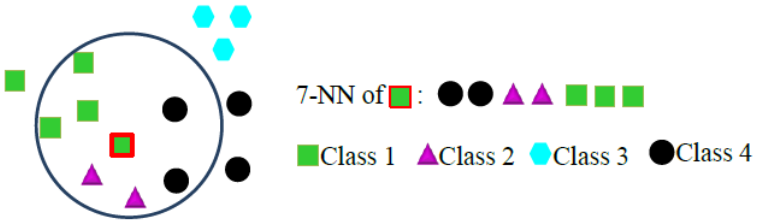


**Fig. 6.** Example of *k*-NN code construction.The code size is the number of classes (4, in this example). For $x_i$, 7 nearest neighbors measured by the distance metrics $M_{LMNN}$ and $M_{DML}$ is used to find the k nearest neighbors and the *k*-NN code is $b_i = [3, \quad 2, \quad 0, \quad 2]^T$.

## 5. Experimental Results

We evaluate the performance of the proposed verification system using the LFW dataset [22], which is a challenging benchmark for face verification. We first describe the training and test protocols for the experiments, then describe the experiments conducted to investigate the optimal parametric setting of the proposed approaches. Finally, we compare our proposed approach with existing algorithms.

### 5.1 Training and test protocol

The LFW database is a challenging database as it comprises 5749 subjects of 13,233 images downloaded from Yahoo! News between 2002-2003. It contains a large variety in facial poses, expressions, lighting conditions, and occlusions, as shown in **Fig. 7**. Because the image numbers for subjects vary, according to our proposed approach to learn the distance relation between positive and negative pairs, only subjects that contain more than 10 images are selected and 116 subjects with 1691 images are used in our experiments. Note that aligned versions of faces are used in the following experiments, and after face detection, only the central part of the 100×120 pixels is cropped. For each experiment, ten runs are performed and each run randomly selected 10 images for each subject in the training process. The remaining images are used for the test

**Fig. 7.** Facial examples in the LFW database

### 5.2 DML parametric settings

In Eq. (10), our approach measures the distance of one image from the other training images using two distance metrics: LMNN [13] and our proposed metric DML. We used the value *w* for a weighted combination of the measured distances using these two metrics. In other words, the weighted value *w* indicates the importance of each of the distance metrics. We set *w* to be from 0.4 to 0.7. We then used the receiver operating characteristic (ROC) curves with the x-axis for a false positive rate (FPR) and the y-axis for a true positive rate (TPR) to show the experimental results **Fig. 8**. We can see that the effect of the weighted value is not obvious. **Table 1** lists the true positive rate when the false positive rate is set to 0.3. Because *w*=0.6

yielded the best results, this value was set for the following experiments.

In the second experiment, we investigated the *k* value of the *k*-NN code for verification, which can be seen as a range in the feature space. For each image the *k*-NN code estimates the label distribution of *k* neighbors from 1160 training images. Two images of the same subject should have similar label distributions. From the results shown in **Fig. 9**, we can see that when *k* is smaller than 40 (about 0.035 times the training images), performance is unsatisfactory due to insufficient statistical information for comparison. However, when a large *k* value is used, a confusing situation happens. This is why when *k* is larger than 100 (about 0.086 times the training images), the performance is degraded as well. This is because in our experimental settings each subject has ten images in the training database and with the *k* value set to 100, we see that the estimate of the label distribution for each image is based on approximately 10 subjects. However, the LFW database has a great variety of appearances for subjects, and the large distribution of similar data measurements causes confusion. **Table 2** shows the true positive rate with the false positive rate set to be 0.3. According to our results, *k* is set as 60 in our experiments.

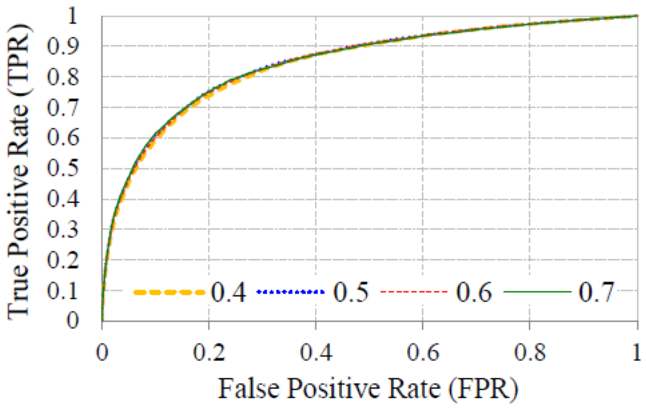


**Fig. 8.** ROC curves by setting various weighted values for *w* in Eq. (10).

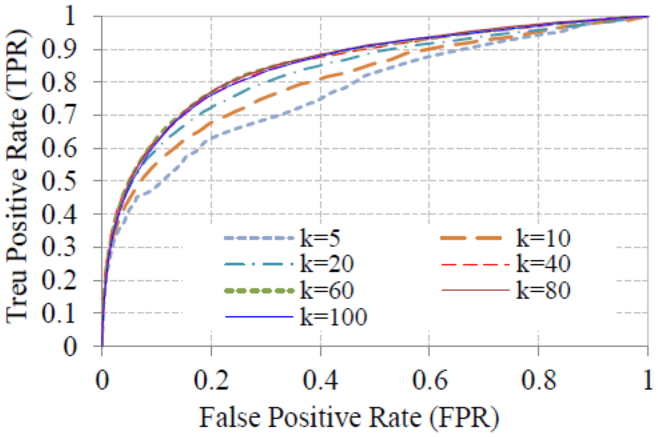


**Fig. 9.** ROC curves by setting various *k* values of the *k*-NN code for verification. As shown, a *k* value that goes from 40 to 80, about 0.035 to 0.070 times the training data size, is recommended.

**Table 1.** True positive rate with a 30% false positive rate by setting various weighted values for *w*

| Weighting value *w* | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|
| True positive rate | 82.41±0.5% | 82.95±0.4% | 84.32±0.4% | 82.89±0.5% |

**Table 2.** True positive rate with a 30% false positive rate by setting various values for *k* in *k*-NN code

| *k* values | 5 | 10 | 20 | 40 |
|---|---|---|---|---|
| True positive rate | 68.83±1.0% | 75.74±0.7% | 80.48±0.7% | 83.52±0.5% |
| *k* values | 60 | 80 | 100 | |
| True positive rate | 84.32±0.4% | 84.01±0.5% | 83.71±0.6% | |

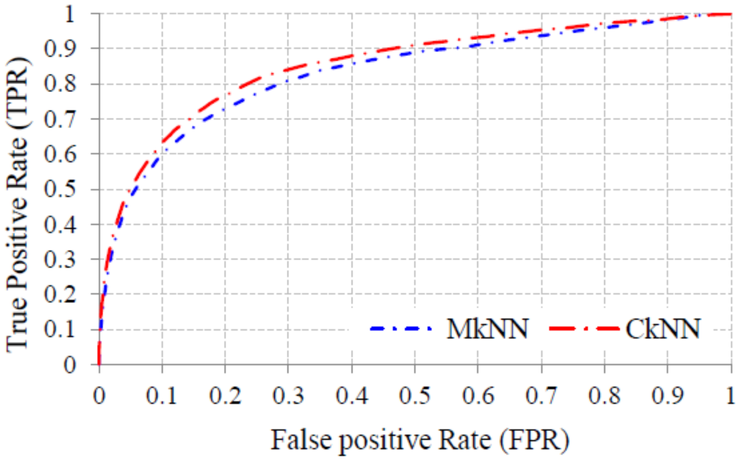


**Fig. 10.** ROC curves by different measurement approaches for the *k*-NN code: MkNN and the proposed approach, CkNN.

In addition, in the test process, when obtaining the *k*-NN code for each input test image, rather than using probability to measure the *k*-NN code similarity as with MkNN [14], we propose the CkNN approach to measure the correlation coefficients of the *k*-NN code. **Fig. 10** compares the ROC curve with MkNN [14]. When the false positive rate is set to be 0.3, the true positive rate for MkNN and the proposed measurement approach for the *k*-NN code are 81.22% and 84.32%, respectively. The proposed approach has a better result than MkNN because MkNN makes an incorrect classification when two images of the same subject receive a low similarity rating if the class labels of their corresponding *k* nearest neighbors are uniformly distributed.

### 5.3 System performance comparison

The proposed metric LMNN + DML using CkNN is compared with the existing metric learning algorithms LMNN [13] and LDML [14] with the classification mechanism MkNN. **Fig. 11** shows the ROC curves. Using the classification mechanism CkNN, LMNN + DML can compensate for the drawbacks of LMNN and DML and provide better results than LMNN or DML alone. In addition, if using only the proposed metric DML with CkNN, better results are provided than when using LDML with MkNN [14]. **Table 3** lists the true positive rate value when the false positive rate is set to be 0.3. Compared with LMNN alone, integrating DML with LMNN can improve the verification rate by 4.3%.

To further analyze the verification power of LMNN and DML together, for each test image $x_{test}$, we generated *m* test pairs with training images as $(x_{test}, x^{i}_{train})$ *i*=1~*m* (*m*=1160 in our work) to verify whether or not $(x_{test}, x^{i}_{train})$ are the same subject. **Figs. 12** and **13** show examples of images, which were incorrectly classified more than by $0.5 \times m$ times by LMNN and DML, respectively. We see that the error examples for LMNN are those images with larger variations in pose, expression, and occlusion while the errors for DML are frontal images. This is because only between-class information is considered by DML and its estimated data distribution for within-class is not as compact as that by LMNN. Hence, compared with LMNN, DML is expected to modify the distance relationships for between-class data to cope with large variations. To verify DML's abilities we selected 51 outlier examples (as shown in **Fig. 14**) including 20 images with non-frontal poses, 20 images with exaggerated expressions, and 11 images with heavy occlusions for test, and the ROC curves are shown in **Fig. 15**. The true positive rate with a false positive rate of 0.3 for LMNN and DML are 66.86% and 71.96%, respectively. We can see that DML has more tolerance for facial variations than LMNN. Therefore, in our method, we integrate LMNN and DML so they can compensate for each other.

**Table 3.** Performance comparison of true positive rate with 30% false positive rate

| Metric Algorithms | LMNN+DML+CkNN | LMNN+CkNN | DML+CkNN | LDML+MkNN |
|---|---|---|---|---|
| True positive rate | 84.32±0.4% | 80.5±0.6% | 74.91±1.0% | 41.67±1.2% |

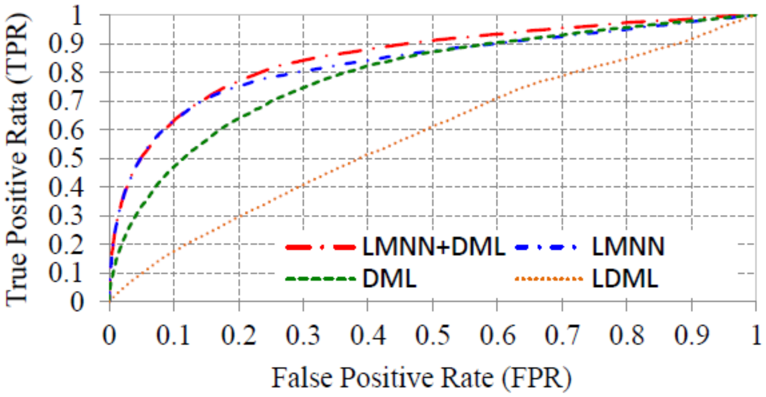


**Fig. 11.** Comparison of ROC curves obtained by the proposed approach and other existing methods: LMNN+DML+CkNN, LMNN+CkNN, DML+CkNN, and LDML+MkNN

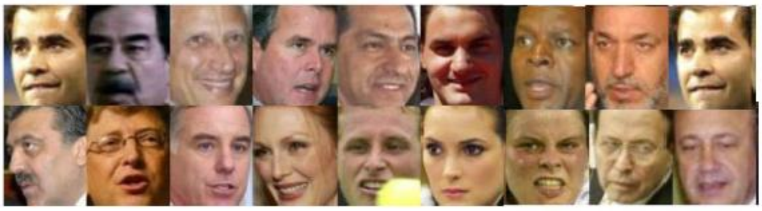


**Fig. 12.** Examples that are incorrectly classified more than one half of test data by LMNN

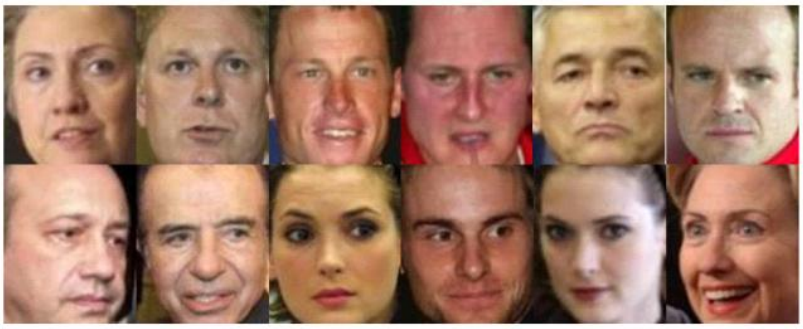


**Fig. 13.** Examples that are incorrectly classified more than one half of test data by DML

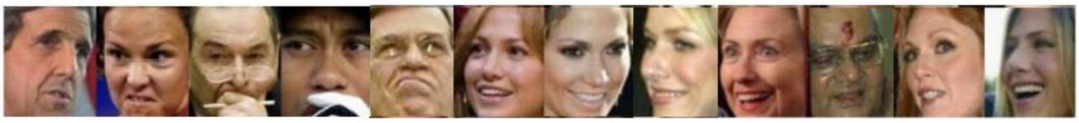


**Fig. 14.** Outlier examples used to analyse DML's abilities in between-class verification

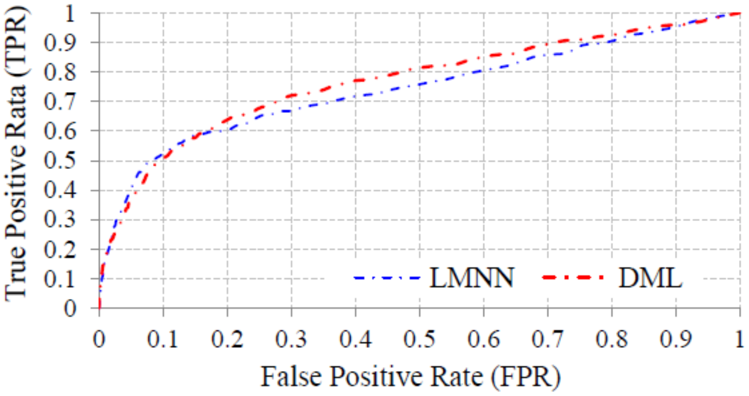


**Fig. 15.** ROC curves obtained by LMNN and DML using 51 outlier examples

In addition, we study the impact on number of failure cases by using the different appearance factors including frontal view (the cases with $\pm 15^{\circ}$ out-of-plan rotation), expression (excluding the cases of natural expression), pose (excluding the cases of frontal view), and others (the remaining cases from the above factors) for LMNN+DML and LMNN as shown in **Fig. 16**. The three examples shown in **Fig. 16 (a)-(c)** are the most improved cases and the three ones in **Fig. 16 (d)-(f)** are the worst cases. The test example in **Fig. 16 (a)** is a slightly rotated smile face. Compared with LMNN, LMNN+DML has the improved rates, 7.1%, 43.3%, 77.8% and 71.4% for frontal view, expression, pose and other factors, respectively. The test examples in **Fig. 16 (b)** and **Fig. 16 (c)** are with occlusion and non-frontal views with expression, respectively. It is not surprisingly that the number of failure cases of the frontal view factor is more than that of the ones in **Fig. 16 (a)** and **(c)** due to the occlusion and the missing of facial information. The integration of DML and LMNN can reduce the error rate 26.4% and 58.1 % for frontal view and pose, respectively. In **Fig. 16 (c)**, because the variations in the rotation angle and the expression degree are higher than those in **Fig. 16 (a)**, more failure cases happened for both LMNN+DML and LMNN, especially in the expression and pose factors. For the three cases shown in **Fig. 16 (a)-(c)**, we have the significant improved rate 52.8% in the pose factor on average.

The test example of **Fig. 16 (d)** is a woman with a hat, which caused uneven shadow on her face. For LMNN+DML, the number of failure cases of the frontal view is higher than that of expression, pose or other, and we also observed that the degrading ratio is significant as compared with LMNN. Although the distance relations of the negative pairs which violated the constraint are modified in the training process, those negative data are still close in the transformed space. **Fig. 16 (e)** is the test example of being occluded by a white head band. The numbers of failure cases of the four factors are higher than those of pure LMNN. **Fig. 16 (f)** shows the results for a grin. Because many female subjects are collected with expression in the LFW dataset, especially smile and grin expressions, the data distribution is much entangled and error rate is therefore higher than that of the other five cases for both LMNN and LMNN+DML. We observed that the degrading degree is small in the pose and expression variations and DML considering only the distance relation of negative pairs has limitation for inter-class variations of frontal views.

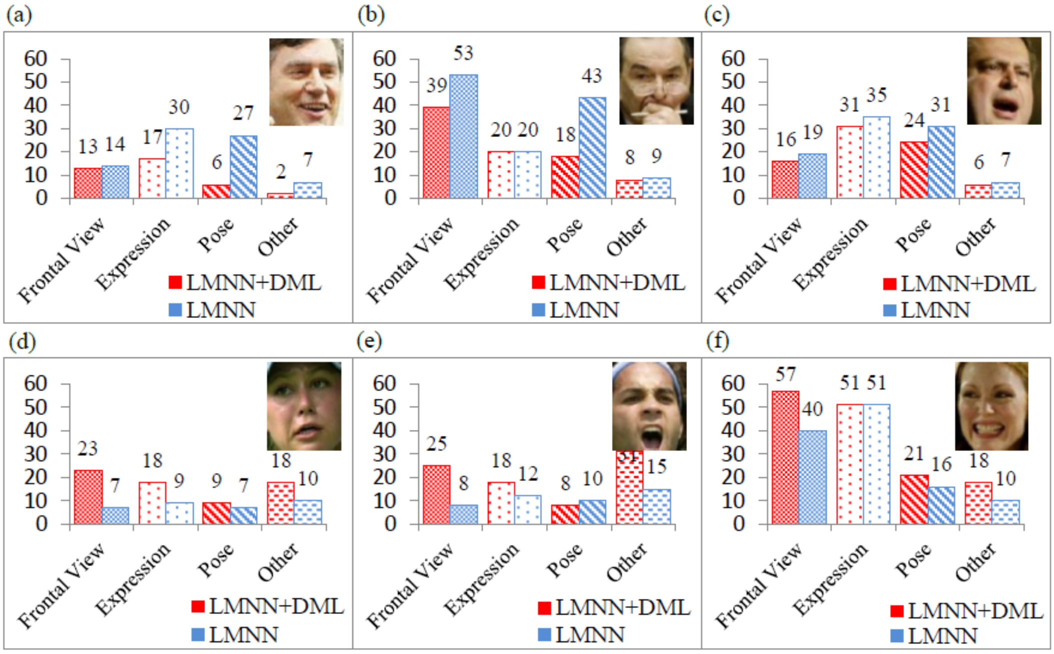


**Fig. 16.** The impact on number of failure cases by using the different appearance factors including frontal view (the cases with $\pm 15°$ out-of-plan rotation), expression (excluding the cases of natural expression), pose (excluding the cases of frontal view), and others (the remaining cases from the above factors) for LMNN+DML and LMNN with the test example shown in the top-right corner.

**Fig. 17** has been modified in accordance with the test examples as in **Fig. 16** to show the failure examples classified by LMNN and LMNN+DML. From **Fig. 17 (a)** and **(c)** we see that DML can help to correctly classify images with large variations that were incorrectly classified by LMNN. For examples, the facial images with expression and slightly out-of-plan rotation angle (the first row in **Fig. 17(a)**), and even larger rotation angle and occlusion in the cheek (the second row in **Fig. 17(a)**) can be correctly classified by DML. **Fig. 17 (c)** shows that the DML is able to classify cases with higher degree of expression and grinning expression than LMNN (**Fig. 17 (a)**). We observed that the images with higher degree of expression (the first row in **Fig. 17 (c)**) and rotated facial images (the second row in **Fig. 17 (c)**) can be rescued by DML from LMNN. However, DML has limited correction ability for images with small variations, as shown in **Fig. 17 (b)** and **(d)**. Regardless of inter-class variations, we can observe these failure cases are near frontal view images with lower degree of smile expression (the first row in **Fig. 17 (b)** and the first row in **Fig. 17 (d)**) as compared with the examples in **Fig. 17 (a)** and **(c)** or with natural expression (the second row in **Fig. 17 (c)** and the second row in **Fig. 17 (d)**). The conclusion is consistent to the results as observed in Fig. 16 that the improved rate of the frontal view in the **Fig. 16 (a)-(c)** is smaller than that of the pose factor, and the test examples might be incorrectly classified, leading the low accuracy rate as shown in **Fig. 16 (d)-(f)**. The reason is that considering only the class label distribution of nearest neighbors in the test process may cause this misclassification, especially for the entangled data distribution like the LFW dataset. In order to improve the accuracy rate, combining multiple complementary features like texture and color [36] or learning more robust features by deep network architectures [41], [42] from facial images can reduce the inter-class variation. Some researchers also tried to classify images by using the ranking results from the training data or extra data set [35] to improve the accuracy for the entangled data.

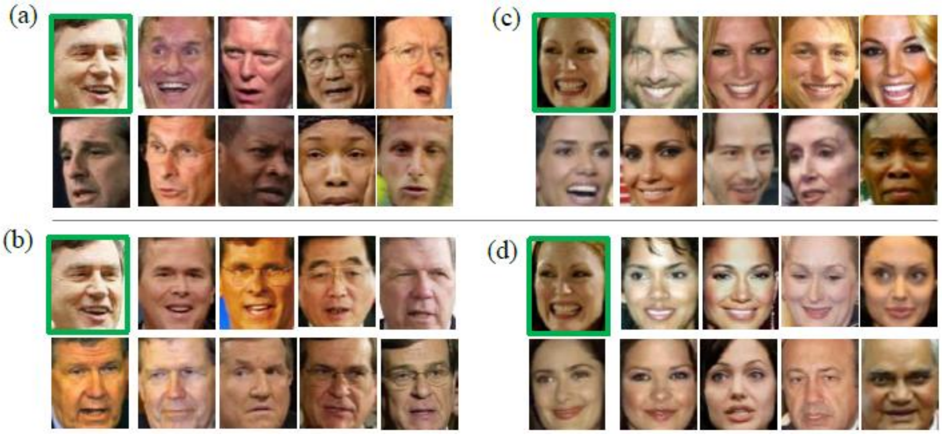


**Fig. 17.** (a) and (c) are misclassified examples incorrectly classified by LMNN but correctly classified by LMNN+DML. (b) and (d) are misclassified examples incorrectly classified by both LMNN and LMNN+DML. The test image is shown with green edge.

## 6. Conclusion

In this paper, we propose a face verification framework that uses a distance metric based on two concepts. First, we propose a distance metric "DML" that penalizes violations of the distance relationship of negative pairs. Second, the distance relationship of positive pairs is optimized via LMNN. The experimental results confirm that the proposed verification framework can reduce the false positive rate than that by using only LMNN. Moreover, the proposed classification mechanism, by measuring the label distribution of a *k*-NN code in two images, can modify the errors caused by low probability for an entangled data distribution and provide better performance than MkNN. In this study, only texture-based local features are extracted from facial images. Inspired by the impressive recognition rate improvements achieved by combining texture and color features [36], [40], we plan to investigate the use of this effective combination approach in the metric learning framework in future research.

## Acknowledgment

This work is supported by National Science Council (NSC), Taiwan, under Contract of MOST103-2221-E-151-031. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## References

[1] X. Wang and X. Tang, "Dual-space linear discriminant analysis for face recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 564-569, 2004. Article (CrossRef Link).

[2] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1222-1228, 2004. Article (CrossRef Link).

[3] X. Wang and X. Tang, "Random sampling for subspace face recognition," *International Journal of Computer Vision*, Vol. 70, No. 1, pp. 91-104, 2006. Article (CrossRef Link).

[4] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 12, pp. 2037-2041, 2006. Article (CrossRef Link).

[5] T. Berg and P. Belhumeur, "Tom-vs-Pete classifiers and identity-preserving alignment for face verification," in *Proc. of British Machine Vision Conference*, 2012. Article (CrossRef Link).

[6] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2707-2714, 2010. Article (CrossRef Link).

[7] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proc. of European Conference on Computer Vision*, pp. 566-579, 2012. Article (CrossRef Link).

[8] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3025-3032, 2013. Article (CrossRef Link).

[9] N. Kumar, A.C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. of International Conference on Computer Vision*, pp. 365-372, 2009. Article (CrossRef Link).

[10] Q. Yin, X. Tang, and J. Sun, "An associate-predict model for face recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 497-504, 2011. Article (CrossRef Link).

[11] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proc. of International Conference on Computer Vision*, pp. 113-120, 2013. Article (CrossRef Link).

[12] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. of British Machine Vision Conference*, 2013. Article (CrossRef Link).

[13] K.Q. Weinberger, J. Blitzer, and L.K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, Vol. 10, pp. 209-244, 2009. Article (CrossRef Link).

[14] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. of International Conference on Computer Vision*, pp. 489-505, 2009. Article (CrossRef Link).

[15] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, Vol. 6, pp. 937-965, 2005. Article (CrossRef Link).

[16] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon, "Information theoretic metric learning," in *Proc. of International Conference on Machine Learning*, pp. 209-216, 2007. Article (CrossRef Link).

[17] A. Globerson and S. Roweis, "Metric learning by collapsing classes," *Advances in Neural Information Processing Systems*, pp. 451-458, 2005. Article (CrossRef Link).

[18] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components

analysis," *Advances in Neural Information Processing Systems*, pp. 513-520, 2004. Article (CrossRef Link).

[19] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," *Advances in Neural Information Processing Systems*, pp. 505-512, 2002. Article (CrossRef Link).

[20] D. Kedem, S. Tyree, F. Sha, G.R. Lanckriet, and K.Q. Weinberger, "Non-linear metric learning," *Advances in Neural Information Processing Systems*, pp. 2573-2581, 2012. Article (CrossRef Link).

[21] OpenCV http://opencv.org/

[22] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," *University of Massachusetts*, 2007. Article (CrossRef Link).

[23] Y. Liang, S. Liao, L. Wang, and B. Zou, "Exploring regularized feature selection for person specific face verification," in *Proc. of International Conference on Computer Vision*, pp. 1676-1683, 2011. Article (CrossRef Link).

[24] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proc. of International Conference on Computer Vision*, pp. 1489-1496, 2013. Article (CrossRef Link).

[25] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701-1708, 2014. Article (CrossRef Link).

[26] G.B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2518-2525, 2012. Article (CrossRef Link).

[27] H.V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. of Asian Conference on Computer Vision*, pp.709-720, 2010. Article (CrossRef Link).

[28] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No.7, pp. 971-987, 2002. Article (CrossRef Link).

[29] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," in *Proc. of International Conference on Computer Vision*, Vol. 60, No.2, pp. 91-110, 2004. Article (CrossRef Link).

[30] L. Wiskott, J.M. Fellous, N. Krger, and C.V.D. Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775-779, 1997. Article (CrossRef Link).

[31] N. Pinto, J. DiCarlo, and D. Cox, "How far can you get with a modern face recognition test set using only simple features?," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2591 - 2598, 2009. Article (CrossRef Link).

[32] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *Lecture Notes in Computer Science*, Vol. 4778, pp. 168-182, 2007. Article (CrossRef Link).

[33] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," *Lecture Notes in Computer Science*, Vol. 4642, pp. 11-18, 2007. Article (CrossRef Link).

[34] S. Brahnam, L.C. Jain, L. Nanni, and A. Lumini, "Local binary patterns: new variants and applications," Springer 2014. Article (CrossRef Link).

[35] L. Wolf, T. Hassner, and Y. Taigman, "Effective Unconstrained Face Recognition by Combining Multiple Descriptors and Learned Background Statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 10, pp. 1978 - 1990, 2010. Article (CrossRef Link).

[36] J.Y. Choi, Y.M. Ro, and K. N. Plataniotis, "Color local texture features for color face recognition," *IEEE Transactions on Image Processing,* Vol. 2, No. 3, pp. 1366-1380, 2012. Article (CrossRef Link).

[37] P. Shih and C. Liu, "Improving the face recognition grand challenge baseline performance using color configurations across color spaces," in *Proc. of International Conference on Image Processing*, pp. 1001–1004, 2006. Article (CrossRef Link).

[38] Z. Liu and C. Liu, “A hybrid color and frequency features method for face recognition,” *IEEE Transactions on Image Processing*, Vol. 17, No. 10, pp. 1975–1980, 2008. Article (CrossRef Link).

[39] J.Wang and C. Liu, “Color image discriminant models and algorithms for face recognition,” *IEEE Transactions on Neural Network*, Vol. 19, No. 12, pp. 2088–2097, 2008. Article (CrossRef Link).

[40] C.J. Young, Y.M. Ro, and K. N. Plataniotis, “Boosting color feature selection for color face recognition,” *IEEE Transactions on Image Processing*, Vol. 20, No. 5, pp. 1425-1434, 2011. Article (CrossRef Link).

[41] M. Ranzato, Y. Boureau, and Y. LeCun, “Sparse feature learning for deep belief networks,” *Advances in Neural Information Processing Systems*, pp. 1185-1192, 2007. Article (CrossRef Link).

[42] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: closing the gap to human-level performance in face verification,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. Article (CrossRef Link).

[43] F.F. Li, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 594–611, 2006. Article (CrossRef Link).

[44] M. Fink, “Object classification from a single example utilizing class relevance pseudo-metrics,” *Advances in Neural Information Processing Systems*, pp. 449-456, 2005. Article (CrossRef Link).

[45] L. Wolf, T. Hassner, and Y. Taigman, “Descriptor based methods in the wild,” in *Proc. of Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008. Article (CrossRef Link).

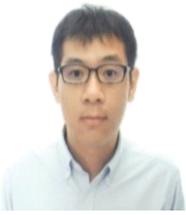
[46] Y. Taigman, L. Wolf, and T. Hassner, “Multiple one-shots for utilizing class label information,” in *Proc. of British Machine Vision Conference*, pp. 1-12, 2009. Article (CrossRef Link).

[47] W.S. Chu, J.C. Chen, and J.J. James Lien, “Kernel discriminant transformation for image set-based face recognition,” *Pattern Recognition*, Vol. 44, No. 8, pp. 1567-1580, 2011. Article (CrossRef Link).

[48] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 711–720, 1997. Article (CrossRef Link).

[49] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 216-2178, 2006. Article (CrossRef Link).

**Ju-Chin Chen** received her B.S., M.S. and Ph.D. degrees in computer science and i nformation engineering from the National Cheng Kung University, Tainan, Taiwan, i n 2002, 2004 and 2010, respectively. She is now an assistant professor in the Depart ment of Computer Science and Information Engineering at the National Kaohsiung U niversity of Applied Sciences, Taiwan. Her research interests lie in the fields of mach ine learning, pattern recognition, and image processing.



**Pei-Hsun Wu** received his B.S. degree in computer science and information enginnering from National Chung Cheng Univeristy, Chiayi, Taiwan, in 2009. He received his M.S. degree in computer science and information engineering from the National Cheng Kung University, Tainan, Taiwan, in 2011. His research interests lie in the fields of machine learning and computer vision.

**Jenn-Jier James Lien** received his M.S. and Ph.D. degrees in electrical engineering from Washington University, St. Louis, MO, and the University of Pittsburgh, Pittsburgh, PA, in 1993 and 1998, respectively. From 1995 to 1998, he was are search assistant at the Vision Autonomous Systems Center in the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA. From 1999 to 2002, he was a senior research scientist at L1-Identity (formerly Visionics) and a project lead for the DARPA surveillance project. He is now professor in the Department of Computer Science and Information Engineering at the National Cheng Kung University, Taiwan.