# Multi-view Deep Network for Cross-view Classification

Meina Kan[1,2] Shiguang Shan[1,2] Xilin Chen[1]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]CAS Center for Excellence in Brain Science and Intelligence Technology
{kanmeina, sgshan, xlchen}@ict.ac.cn

## Abstract

*Cross-view recognition that intends to classify samples between different views is an important problem in computer vision. The large discrepancy between different even heterogenous views make this problem quite challenging. To eliminate the complex (maybe even highly nonlinear) view discrepancy for favorable cross-view recognition, we propose a multi-view deep network (MvDN), which seeks for a non-linear discriminant and view-invariant representation shared between multiple views. Specifically, our proposed MvDN network consists of two sub-networks, view-specific sub-network attempting to remove view-specific variations and the following common sub-network attempting to obtain common representation shared by all views. As the objective of MvDN network, the Fisher loss,* i.e. *the Rayleigh quotient objective, is calculated from the samples of all views so as to guide the learning of the whole network. As a result, the representation from the topmost layers of the MvDN network is robust to view discrepancy, and also discriminative. The experiments of face recognition across pose and face recognition across feature type on three datasets with 13 and 2 views respectively demonstrate the superiority of the proposed method, especially compared to the typical linear ones.*

## 1. Introduction

The images of an object can be captured by different cameras, different sensors, or from different view angles, which brings about a great challenge of matching images from these different views to recognize them. This kind of problem is generally called as cross-view recognition or heterogenous recognition. Usually, the appearance of samples from different views are quite different from each other, and the large view discrepancy makes it quite challenging to directly compare them based on the image appearance. Substantial efforts have been dedicated to eliminate the view discrepancy or extract view-invariant feature presentations.

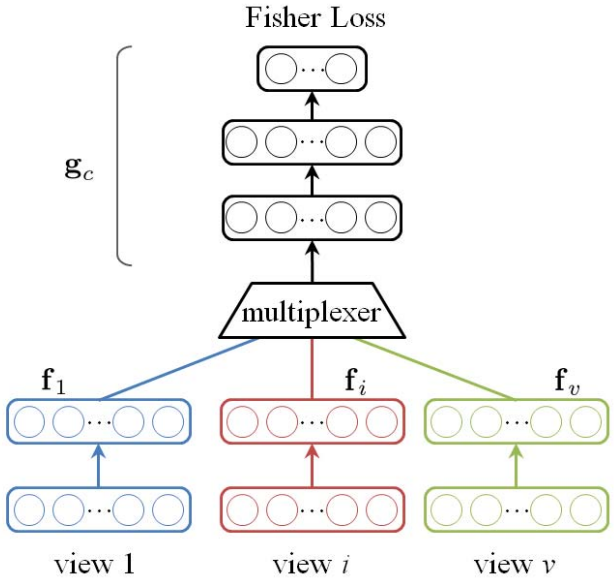


Figure 1. An overview of Multi-view Deep Network (MvDN). MvDN consists of two sub-networks, the view-specific sub-network $\mathbf{f}_i|_{i=1}^{v}$ and the common sub-network $\mathbf{g}_c$, along with a discriminant Fisher objective.

A straightforward way to deal with the view discrepancy is to project all views into a common space, like the Canonical Correlation Analysis (CCA) [10][2]. CCA attempted to learn two transforms, one for each view, to respectively project the samples from the two views into a common subspace, by maximizing the cross correlation between two views. CCA is only applicable for two-view scenario. To deal with multi-view cases, the pairwise strategy is usually exploited or a more efficient and robust solution is to learn a unified common space shared by all views rather than only two views. For this purpose, the Multiview CCA (MCCA) [25][27] was proposed to obtain one common space for $v$ views. In MCCA, $v$ view-specific transforms, one for each view, are obtained by maximizing the total correlations between any two views. Another commonly used method was

1063-6919/16 $31.00 © 2016 IEEE
DOI 10.1109/CVPR.2016.524



IEEE computer society

proposed in [28][5], which employed Partial Least Squares (PLS) regression to regress the samples from one view to another. Specifically for photo vs. sketch face recognition, a coupled information-theoretic projection tree [36] was proposed to reduce the modality gap by maximizing the mutual information between photos and sketches in the quantized feature spaces. In [34], a pair of semi-coupled dictionaries were proposed to characterize both views with a mapping function modeling the intrinsic relationship between the two views, and this work was further extended by using a unified model for coupled dictionary and feature space learning in [11]. Besides, some methods employed either view as the common space, *e.g.*, a pseudo-sketch of photo was synthesized for photo-sketch recognition [32][20].

Although the view discrepancy can be minimized by the above methods, the discriminant information, *e.g.*, class label, is not explicitly taken into account, which is unfavorable for recognition or classification. Therefore, quite a few methods has examined how to incorporate the discriminant information as well as the view discrepancy.

In [21][31][15], CCA was extended to Correlation Discriminant Analysis and Discriminative Canonical Correlation Analysis by maximizing the within-class correlation and minimizing between-class correlation across two-view for a discriminant common space. In [7][8], Multiview Fisher Discriminant Analysis was proposed to employ the label information for binary classification. In [19], Common Discriminant Feature Extraction (CDFE) was proposed to minimize the intra-class scatter and meanwhile maximize the inter-class separability, resulting in very encouraging performance. In [6], a large margin approach was proposed to discover a predictive latent subspace representation shared by two views based on an undirected latent space Markov network. In [17], Coupled Spectral Regression (CSR) learnt a projection from the observation to the common low-dimensional embedding of the class label through least squares regression. Similarly, in [33], two coupled linear regression models were used to project data from different modalities into a common subspace that is directly defined by the class label. In [16], a local feature-based discriminant analysis method was proposed to match a forensic sketch and a mug shot photo. Besides, some other methods proposed to apply discriminant classifier in the common space achieved from some unsupervised method, like [18]. Recently, a generalized multi-view analysis (GMA) framework was proposed in [29], in which the supervised information of each view was coupled with the correlation between views, leading to a discriminant common subspace. Furthermore, a multi-view discriminant analysis (MvDA) approach [13] was proposed by considering intra-view discriminancy, inter-view discriminancy, and view correlations all together in an unified framework. Benefitted from the supervised information, these discriminant methods usually outperform those unsupervised ones.

Most of these methods are linear ones, and may become insufficient for challenging scenarios. At first thought, easily they can be extended to non-linear models with kernel trick, such as Kernel Canonical Correlation Analysis (KCCA) [2][22]. However, it is trival to design a favorable kernel and also it is inefficient to deal with the out-of-sample problem. So recently, several works proposed to employ the more flexible deep neural network to handle the non-linear discrepancy between views, and achieved promising results.

In [24], a Multimodal Deep Auto-encoder was proposed, which took the two views as input and outputted two views too, so as to learn shared representation of both views. In [30], a Multimodal Deep Botlzman Machine (DBM) was proposed to jointly model the distribution over two views. In [3], Deep Canonical Correlation Analysis (DCCA) was proposed to learn complex nonlinear transformations for each of the two views so that the resulting representations are highly linearly correlated. As evaluated, DCCA performs better than Kernel CCA [2][22]. Specifically for the face recognition across view, *e.g.* pose and illumination, [37] proposed a convolutional-like deep neural network to learn the face identity-preserving features, in which all views are reconstructed to a common view, *i.e.* the canonical view. Furthermore, [38] proposed a Multi-view Perceptron (MVP) to untangle the identity and view features, and in the meanwhile infer a full spectrum of multi-pose images given a single 2D face image.

All these methods [24][30][3][37][38] employ the deep neural network to model the view distribution and achieved quite promising performance benefited from the favorable ability of non-linearly modeling. However, they are all unsupervised methods, so generally need a successive supervised feature extraction or classifier inducing a better performance for classification or recognition.

Over all speaking, much of the research has well examined how to deal with the view-discrepancy for recognition or classification across view. However, they are linear which cannot well hand the challenging non-linear scenarios, unsupervised deep approaches which are incapable of recognition, or kernelized supervised methods which may get suck with the out-of-sample problem. To cope with all these challenges, we propose an explicitly non-linear and supervised method, named as Multi-view Deep Network.

Our proposed Multi-view Deep Network (MvDN) considers the view-discrepancy and discriminancy simultaneously through a deep architecture, resulting in a discriminant and view-invariant representation shared between multiple views. Specifically, our proposed MvDN consists of two sub-networks, the view-specific sub-network and the common sub-network shared by all views. As the loss function, the Rayleigh quotient objective of samples from all views is employed to ensure the discriminancy of the whole

network. As a result, the feature representation from the topmost layers of MvDN is robust to view variations, and also discriminative.

In the following, Section 2 presents the formulation of Multi-view Deep Network followed by the optimization and some discussions, and Section 3 evaluates the MvDN on two databases, followed by a conclusion.

## 2. Multi-view Deep Network (MvDN)

In this section, we firstly introduce the overview of the proposed Multi-view Deep Network (MvDN) method, and then present the formulation of MvDN, followed by the optimization and some discussions.

### 2.1. Overview

The problem this work mainly attempts to deal with is recognition or classification across view. To deal with this problem, the proposed MvDN seeks for a discriminant and view-invariant representation shared between multiple views. Specifically, MvDN consists of two types of sub-networks, the view-specific sub-networks and the common sub-network, as shown in Figure 1. The view-specific sub-networks $\mathbf{f}_i|_{i=1}^{v}$ are expected to reduce the discrepancy between that view and the commonality of all views. The commonality of all views is enforced by the following common sub-network $\mathbf{g}_c$ with the Rayleigh quotient objective.

The common sub-network is shared by all views in a multiplex way, which means that the common sub-network $\mathbf{g}_c$ is connected to each view-specific sub-network $\mathbf{f}_i$ independently. As a result, the common sub-network can extract a view-invariant representation for any single input view. The Fisher loss is calculated with the samples of all views, which ensures the discriminancy and view-invariancy of the representation from the common sub-network.

### 2.2. Formulation

For clear description in the following, we first define some notations. In the whole text, upper-case and lower-case characters represent the matrices and vectors respectively. Given $v$ views, the $j$-th sample of $i$-th view is denoted as $\mathbf{x}_j^i \in \mathbb{R}^{p_i \times 1}$, and all the $n_i$ samples of $i$-th view are denoted as $\mathbf{X}_i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n_i}^i] \in \mathbb{R}^{p_i \times n_i}$. The corresponding class label is denoted as $\mathbf{L}_i = [l_1^i, l_2^i, \cdots, l_{n_i}^i]$, where $l_j^i \in \{1, 2, 3, \cdots, c\}$ representing the class label of sample $\mathbf{x}_j^i$. Different views contains samples from the same classes but in different views.

For any sample $\mathbf{x}_j^i$ from $i^{th}$ view, its representation from MvDN $\mathbf{y}_j^i$ is generated by passing through the $i^{th}$ view-specific sub-network and following the common sub-network successively, formulated as below:

$$\mathbf{y}_j^i = \mathbf{g}_c\left(\mathbf{f}_i\left(\mathbf{x}_j^i\right)\right). \tag{1}$$

In Eq. (1), $\mathbf{f}_i$, *i.e.* the sub-network specific to the $i^{th}$ view, is responsible for eliminating the particular information of $i^{th}$ view, and $\mathbf{g}_c$, *i.e.* the common sub-network shared by all views, further extracts the discriminant representation shared by all views. With the view-invariant feature representation $\mathbf{y}_j^i$, the samples of different views can be effectively compared for recognition or classification. Generally, the two sub-networks is designed with non-linear activation function so as to characterize the challenging discrepancy between views, *e.g.* sigmoid, tanh, ReLU, etc. Besides, each of the two sub-networks can contain one or more layers resulting in a deep architecture of MvDN.

To ensure that the representation $\mathbf{y}_j^i$ from MvDN is discriminant and robust to view discrepancy, the Rayleigh quotient of samples from all views is employed as the objective function as follows:

$$[\mathbf{g}_c^*, \mathbf{f}_1^*, \cdots, \mathbf{f}_v^*] = \arg\min_{\mathbf{g}_c, \mathbf{f}_1, \cdots, \mathbf{f}_v} Tr\left(\frac{\mathbf{S}_W^y}{\mathbf{S}_B^y}\right), \tag{2}$$

where $Tr(\cdot)$ denotes the trace of a matrix, $\mathbf{S}_W^y$ denotes the within-class scatter of samples from all $v$ view, and $\mathbf{S}_B^y$ denotes the between-class scatter of samples from all $v$ views. The within-class scatter and between-class scatter in Eq. (2) are both calculated with samples from all views, meaning that not only the view-discrepancy but also the intra-view and inter-view discriminancy are considered, inducing a discriminant and view-invariant representation shared between all views. More detailed illustration will be given in the following.

The within-class scatter $\mathbf{S}_W^y$ is calculated as below:

$$\mathbf{S}_W^y = \sum_{k=1}^{c}\sum_{i=1}^{v}\sum_{j=1}^{n_{ki}}\left(\mathbf{y}_{jk}^i - \boldsymbol{\mu}_k\right)\left(\mathbf{y}_{jk}^i - \boldsymbol{\mu}_k\right)^T, \tag{3}$$

where $\mathbf{y}_{jk}^i$ representing the $j^{th}$ sample of $i^{th}$ view of $k^{th}$ class, and $n_{ki}$ representing the number of samples in $i^{th}$ view of $k^{th}$ class. $\boldsymbol{\mu}_k$ is the mean of $k^{th}$ class, calculated as $\boldsymbol{\mu}_k = \frac{1}{n_{ki}}\sum_{i=1}^{v}\sum_{j=1}^{n_{ki}} y_{jk}^i$. Eq. (3) can be equivalently reformulated as below with a scale of $\gamma$:

$$\mathbf{S}_W^y = \gamma\sum_{k=1}^{c}\sum_{i,i'=1}^{v}\sum_{j,j'=1}^{n_{ki}}\left(\mathbf{y}_{jk}^i - \mathbf{y}_{j'k}^{i'}\right)\left(\mathbf{y}_{jk}^i - \mathbf{y}_{j'k}^{i'}\right)^T, \tag{4}$$

As seen, the within-class scatter $\mathbf{S}_W^y$ calculated from sample of all view not only ensures the closeness of samples from the same class and same view (*i.e.* when $i = i'$), but also ensures the closeness of sample from the same class but different views (*i.e.* when $i \neq i'$). In other words, minimizing the within-class scatter $\mathbf{S}_W^y$ ensures the closeness of samples from the same class regardless of the view.

Similarly, the between-class scater $\mathbf{S}_B^y$ is calculated as below:

$$\mathbf{S}_B^y = \sum\nolimits_{k=1}^{c} n_k\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}\right)^T, \tag{5}$$

where $n_k = \sum_{i=1}^{v} n_{ki}$ is the number of samples from all views in $k^{th}$ class, $\boldsymbol{\mu}_k$ is the mean of the $k^{th}$ class defined as in Eq. (3), and $\boldsymbol{\mu}$ is the mean of all samples from all views, calculated as $\boldsymbol{\mu} = \frac{1}{n}\sum_{k=1}^{c}\sum_{i=1}^{v}\sum_{j=1}^{n_{ki}} y_{jk}^{i}$ with $n = \sum_{k=1}^{c} n_k$. In Eq. (5), the between-class scatter is computed with samples from all views, and as so it can maximize the distance of different classes regardless of view.

As in Eq. (3), the within-class scatter enforces the samples of different view but from the same class close to each other so as to obtain a view-invariant representation shared between all views. Furthermore, as in Eq. (3) and Eq. (5), the within-class scatter and between-class scatter consider both the intra-view discriminancy and inter-view discriminancy can achieve a discriminant representation. Therefore, maximizing the between-class scatter and minimizing the within-class scatter calculated from samples of all view simultaneously as in Eq. (2) can result in a discriminant and view-invariant representation shared between all views.

## 2.3. Optimization

Following the most existing works of deep learning, we employ the gradient descent to optimize the multi-view deep network in Eq. (2). In the following, we will give the details about how to conduct the gradient descent for the Fisher loss, the common sub-network, and the view-specific sub-networks respectively.

Overall speaking, the whole MvDN is optimized by the gradient descent following the chain rule, *i.e*. firstly compute the loss of objective, and then prorogate the loss to each layer so as to compute the gradient of each layer, and finally employ gradient descent to update the whole network.

**Step 1: Feed forward and calculate the loss.** For each of the $v$ views, the samples of $\mathbf{X}_i$ are fed forward to the MvDN as in Eq. (1), and the output of the MvDN is denoted as $\mathbf{Y}_i$, with $\mathbf{Y}_i = [\mathbf{y}_1^i, \mathbf{y}_2^i, \cdots, \mathbf{y}_{ni}^i]$. Then based on the samples of all views $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_v]$, the loss of the whole network is calculated as in Eq. (2), denoted as $J = Tr\left(\frac{\mathbf{S}_W^y}{\mathbf{S}_B^y}\right)$.

**Step 2: Gradient of the loss layer.** As the loss $J$ is directly calculated with the output of the MvDN, there is no parameter involved, so we firstly need to calculate the gradient of $J$ with respect to $\mathbf{Y}$.

According to [14], the gradient of Fisher loss can be calculated as follows:

$$\begin{aligned}&\frac{\partial Tr\left(\frac{\mathbf{X}^T\mathbf{A}_1\mathbf{X}}{\mathbf{X}^T\mathbf{A}_2\mathbf{X}}\right)}{\partial \mathbf{X}}\\ =&-2\mathbf{A}_2\mathbf{X}\left(\mathbf{X}^T\mathbf{A}_2\mathbf{X}\right)^{-1}\left(\mathbf{X}^T\mathbf{A}_1\mathbf{X}\right)\left(\mathbf{X}^T\mathbf{A}_2\mathbf{X}\right)^{-1}\\ &+2\mathbf{A}_1\mathbf{X}\left(\mathbf{X}^T\mathbf{A}_2\mathbf{X}\right)^{-1}\end{aligned} \tag{6}$$

According to [35], the within-class scatter matrix $\mathbf{S}_W^y$, the total scatter matrix $\mathbf{S}_T^y$, and the between-class scatter matrix $\mathbf{S}_B^y$ can be reformulated as below:

$$\mathbf{S}_W^y = \mathbf{Y}\left(\mathbf{I} - \sum_{k=1}^{c}\frac{1}{n_k}\mathbf{e}^k\mathbf{e}^{kT}\right)\mathbf{Y}^T = \mathbf{Y}\mathbf{A}_W\mathbf{Y}^T, \tag{7}$$

$$\mathbf{S}_T^y = \mathbf{Y}\left(\mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^T\right)\mathbf{Y}^T = \mathbf{Y}\mathbf{A}_T\mathbf{Y}^T, \tag{8}$$

$$\mathbf{S}_B^y = \mathbf{S}_T^y - \mathbf{S}_W^y = \mathbf{Y}\mathbf{A}_B\mathbf{Y}^T \tag{9}$$

with $\mathbf{A}_B = \mathbf{A}_T - \mathbf{A}_W$. Here, $\mathbf{I} \in \mathbb{R}^{n\times n}$ is an identity matrix, $\mathbf{e}$ is an $n$-dimensional vector with all elements as 1, and $\mathbf{e}^k$ is an $n$-dimensional vector with $\mathbf{e}^k(i) = 1$ if the $i^{th}$ sample of $\mathbf{Y}$ belongs to $k^{th}$ class.

With Eq. (10), Eq.(7) and Eq. (9), the gradient of $J$ w.r.t. $\mathbf{Y}$ can be calculated as follows:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{Y}} =&-2\mathbf{A}_B\mathbf{Y}^T\left(\mathbf{Y}\mathbf{A}_B\mathbf{Y}^T\right)^{-1}\left(\mathbf{Y}\mathbf{A}_W\mathbf{Y}^T\right)\left(\mathbf{Y}\mathbf{A}_B\mathbf{Y}^T\right)^{-1}\\ &+2\mathbf{A}_W\mathbf{Y}^T\left(\mathbf{Y}\mathbf{A}_B\mathbf{Y}^T\right)^{-1}\end{aligned} \tag{10}$$

**Step 3: Gradient of common sub-network $\mathbf{g}_c$.** Denote the output of the view-specific sub-network, *i.e*. the input of the common sub-network, as $\mathbf{z}_j^i = \mathbf{f}_i(\mathbf{x}_j^i)$. The $\mathbf{Z}_i = \left[\mathbf{z}_1^i, \mathbf{z}_2^i, \cdots, \mathbf{z}_{n_i}^i\right]$ and $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_v]$ denote the output of the view-specific sub-network for samples of $i^{th}$ view and all views respectively.

With the gradient of the loss $J$ as in Eq. (10), nextly according to the chain rule we can compute $\frac{\partial \mathbf{Y}}{\partial \mathbf{g}_c}$ (the gradient of $\mathbf{Y}$ w.r.t. the common sub-network $\mathbf{g}_c$) to update the $\mathbf{g}_c$ afterwards. Besides, we also need compute $\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}}$ (the gradient of $\mathbf{Y}$ w.r.t. $\mathbf{Z}$) to propagate the loss to its next layers, *i.e*. the view-specific sub-network.

The common sub-network can include one or more layers, and each layer can be non-linear by using a non-linear activation function, *e.g*. the sigmoid, tanh, or relu. The gradient of $\frac{\partial \mathbf{Y}}{\partial \mathbf{g}_c}$ and $\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}}$ w.r.t. different activation functions can be easily computed according to the 'UFLDL Tutorial' website [23].

**Step 4: Gradient of view-specific sub-network $\mathbf{f}_i$.** $\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}}$ computes the gradient of $\mathbf{Y}$ w.r.t. to all views samples of $\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_v$, *i.e*. $\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}} = \left[\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}_1}, \frac{\partial \mathbf{Y}}{\partial \mathbf{Z}_2}, \cdots, \frac{\partial \mathbf{Y}}{\partial \mathbf{Z}_v}\right]$. As $\mathbf{Z}_i|_{i=1}^{v}$ are computed from different view-specific sub-networks. Therefore, the loss can be propagated independently, and the gradient of each view-specific sub-network $\mathbf{f}_i$ can be calculated independently too. Similarly, the gradient of $\frac{\partial \mathbf{Z}_i}{\partial \mathbf{f}_i}$ w.r.t. different activation functions can be easily computed according to the 'UFLDL Tutorial' website [23].

**Step 5: MvDN update via gradient descent.** Let us denote the parameter of the whole MvDN as $\theta = [\mathbf{g}_c, \mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_v]$. At the iteration $t+1$, the network parameters are updated through the Limited-memory BFGS

(L-BFGS) optimization algorithm, with the gradient calculate as follows:

$$\begin{aligned}\triangle\theta^t &= \left[\triangle\theta^t_{g_c}, \triangle\theta^t_{f_1}, \triangle\theta^t_{f_2}, \cdots, \triangle\theta^t_{f_v}\right] \\ \triangle\theta^t_{g_c} &= \frac{\partial J}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{g}_c} \\ \triangle\theta^t_{f_i} &= \frac{\partial J}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{Z}_i} \cdot \frac{\partial \mathbf{Z}_i}{\partial \mathbf{f}_i}, i = 1, 2, \cdots, v.\end{aligned} \tag{11}$$

Here, $J, \mathbf{Y}, \mathbf{g}_c, \mathbf{f}_i$, and $\mathbf{Z}_i$ are all calculated from the network in the $t^{th}$ iteration.

### 2.4. Discussions

**Differences from CCA, MCCA and PLS.** CCA [10], MCCA [27], and PLS [28] are all unsupervised and linear approaches. Among them, CCA and PLS can only deal with two-view problem, so usually the pair-wise strategy is employed for multiple views. MCCA can naturally deal with multi-view problem. In contrast, our MvDN is not only non-linear and supervised, but also can be applicable for multi-view problem.

**Differences from MvDA, GMA and CDFE.** Compared with CCA-like methods, MvDA [13], GMA [29] and CDFE [19] are supervised. Generally, they can achieve better performance benefited from the considering of discriminancy. However, they are all linear methods, which may be hard to well characterize challenging view discrepancy. In contrast, our MvDN is non-linear and discriminative.

**Differences from Kernel CCA [2].** There are general theories on CCA, so if having some idea of the underlying distribution, we can easily select proper kernel functions to transform it to a new distribution that is suitable for CCA forming an effective KCCA. In most real-world problems, however, we barely have any idea of the underlying distribution, and it will be difficult to design a favorable kernel function. In contrast, our MvDN equipped with deep neural work can automatically be optimized to form a proper distribution for CCA or Fisher objective. Besides, kernel methods suffer from scalability issue, while deep learning methods including our MvDN naturally scale to large-scale problems benefited from the explicit non-linear mapping. Although deep neural network does not have sophisticated theories yet, there are lots of works about how to design an effective network, *e.g*. those from Geoffrey Hinton, Yann LeCun, Yoshua Bengio, etc. Moreover, our MvDN is discriminative, while KCCA is not.

**Differences from DCCA, FIP and MVP.** To model challenging view variations, DCCA [3], FIP [37] and MVP [38] propose to employ the deep neural network to capture the highly non-linear discrepancy between views, and have achieved promising performance. However, they are unsupervised, and with a risk of discarding the discriminancy. Take DCCA for example, the deep network may overfit a given dataset with sacrificing the discriminancy. On the contrary, our MvDN is supervised, and it simultaneously considers discriminancy as well as the view discrepancy, which can ensure a representation that is robust to view discrepancy and also discriminant. Besides, FIP and MVP are designed especially for face recognition across pose, which is inapplicable for the general cross-view problem when the dimension of each view is different. On the contrary, DCCA and our MvDN is applicable even the dimension of each view is different.

**Differences from a vanilla CNN.** A vanilla CNN taking all samples from multiple views as one view may be applicable for cross-view problems with homogeneous view representations (*e.g*. cross-pose), but will fail in general heterogenous cross-view problems where each view is of different dimensionality. Contrastly, our MvDN is general for more than 2 views, even for heterogenous views.

Overall speaking, the existing methods for cross-view recognition problem are unsupervised linear approaches, supervised linear approaches, unsupervised deep non-linear approaches, or implicitly kernel-based supervised non-linear approaches. Differently, our multi-view deep network is a supervised and explicitly deep non-linear approach, which can efficiently characterize more challenging view discrepancy for a better discriminant and view-invariant representation shared between multiple views.

## 3. Experiments

In this section, we evaluate the proposed MvDN and a few existing methods on two cross-view face recognition tasks, *i.e.*, face recognition across pose on MultiPIE [9], and face recognition across feature type on FRGC [26] and LFW [12] datasets. The existing methods of CCA [10], KCCA [2], MCCA [27], PLS [28], MvDA [13], GMA [29], FIP [37] and MVP [38] are evaluated. For the experiments on MultiPIE and FRGC, the face images are cropped into 64x80 with manually labeled eye locations, and for experiments on LFW, the face images are cropped into 80x120. The intensity is used as the feature in all experiments unless otherwise stated. To reduce the feature dimension, Principal Component Analysis (PCA) is applied for each view, and the reduced dimension is set as 200, 300 and 400 on MultiPIE, FRGC and LFW respectively to preserve more than 95% energy.

### 3.1. Face recognition across pose

Face recognition across view angle endeavors to recognize the probe images from one view angle by comparing them with the gallery images that are from another view angle. Face recognition across pose is evaluated on MultiPIE dataset by taking each pose as one view. The MultiPIE dataset [9] contains images of 337 subjects under various poses, illuminations and expressions.

Table 1. Evaluation of face recogntion across view angle on the MultiPIE dataset.

| Methods | -90° | -75° | -60° | -45° | -30° | -15° | 15° | 30° | 45° | 60° | 75° | 90° | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PLS | 0.319 | 0.775 | 0.892 | 0.934 | 0.883 | 0.981 | 0.981 | 0.934 | 0.906 | 0.873 | 0.723 | 0.268 | 0.789 |
| MCCA | 0.409 | 0.742 | 0.822 | 0.723 | 0.685 | 0.920 | 0.906 | 0.798 | 0.747 | 0.779 | 0.714 | 0.376 | 0.718 |
| PLS+LDA | 0.380 | 0.798 | 0.869 | 0.944 | 0.920 | 0.995 | 0.986 | 0.967 | 0.883 | 0.850 | 0.709 | 0.319 | 0.802 |
| MCCA+LDA | 0.488 | 0.662 | 0.817 | 0.887 | 1.00 | 1.00 | 1.00 | 0.995 | 0.831 | 0.803 | 0.676 | 0.568 | 0.811 |
| MvDA | 0.568 | 0.723 | 0.845 | 0.920 | 0.967 | 1.00 | 1.00 | 0.991 | 0.897 | 0.864 | 0.714 | 0.559 | 0.837 |
| GMA | 0.526 | 0.732 | 0.845 | 0.901 | 1.00 | 1.00 | 1.00 | 1.00 | 0.906 | 0.859 | 0.718 | 0.573 | 0.838 |
| **MvDN (Ours)** | **0.704** | **0.822** | **0.883** | **0.911** | **0.991** | **1.00** | **1.00** | **0.991** | **0.930** | **0.911** | **0.798** | **0.709** | **0.887** |

Table 2. Performance of MvDN with different neurons for the second layer from last in the common-subnetwork on the MultiPIE dataset.

| Neurons | -90° | -75° | -60° | -45° | -30° | -15° | 15° | 30° | 45° | 60° | 75° | 90° | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 400 | 0.695 | 0.812 | 0.887 | 0.897 | 0.972 | 0.995 | 1.000 | 0.972 | 0.892 | 0.878 | 0.779 | 0.676 | 0.871 |
| 600 | 0.695 | 0.793 | 0.850 | 0.897 | 0.981 | 1.000 | 1.000 | 0.991 | 0.887 | 0.887 | 0.803 | 0.695 | 0.873 |
| 800 | 0.723 | 0.822 | 0.873 | 0.897 | 0.986 | 0.995 | 1.000 | 0.981 | 0.878 | 0.864 | 0.808 | 0.690 | 0.876 |
| 1000 | 0.747 | 0.808 | 0.873 | 0.901 | 0.977 | 0.995 | 1.000 | 0.977 | 0.897 | 0.869 | 0.812 | 0.737 | 0.883 |
| 1200 | 0.704 | 0.822 | 0.883 | 0.911 | 0.991 | 1.00 | 1.00 | 0.991 | 0.930 | 0.911 | 0.798 | 0.709 | 0.887 |
| 1400 | 0.751 | 0.836 | 0.878 | 0.897 | 0.981 | 1.000 | 1.000 | 0.986 | 0.916 | 0.897 | 0.831 | 0.737 | 0.892 |
| 1600 | 0.723 | 0.840 | 0.859 | 0.892 | 0.977 | 0.991 | 1.000 | 0.977 | 0.892 | 0.878 | 0.798 | 0.732 | 0.880 |
| 1800 | 0.676 | 0.822 | 0.864 | 0.916 | 0.991 | 1.000 | 1.000 | 0.986 | 0.906 | 0.869 | 0.798 | 0.695 | 0.877 |

Specifically, a subset including images from all subjects at 13 poses (-90°, -75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°, 75°, 90°) under no flush illumination from 4 collecting sessions is selected as the evaluation dataset. This evaluation dataset is divided into 13 subsets according to view angle. For each view angle, 708 randomly selected images of the first 229 subjects are used for training, and 213 randomly selected images of the remaining 108 subjects are used for testing. For training data, all 13 poses are divided into only 3 views, *i.e*. [-90°, -45°], [-30°, 30°], and [45°, 90°], to simulate a challenging scenario. In the process of testing, the view of 0° is used as the gallery, and the rest 12 views are used as the probe. All methods are evaluated in terms of rank-1 recognition rate.

For CCA, MCCA, PLS and MvDA, the main parameter is the dimension of the projected representation, and we tune the dimension with step of 50 to report a best result. For GMA, following the suggestions in [29] we set $\mu = 1$, $\gamma$ = trace ratio, tune the $\lambda$ in [0.1 500], and tune the dimension of the projected representation also in step of 50 to report the best result. In our MvDN, each of the three view-specific sub-networks consists of one input layer with 200 neurons(*i.e*. the PCA dimension) and one hidden layer with 300 neurons equipped with ReLU activation function, and the common sub-network consists of one hidden layer with 1200 neurons equipped with ReLU activation function followed by a linear hidden layer with 200 neurons, resulting in a four-layer deep network including the input layer. That is, $\mathbf{f}_i$ is a 200-300 network, and $\mathbf{g}_c$ is 300-1200-200 network, where the input layer of $\mathbf{g}_c$ is also the output of $\mathbf{f}_i$ with 300 neurons. The neurons of the last layer should be smaller than the number of classes to ensure the $\mathbf{S}_W^y$ and $\mathbf{S}_B^y$ non-singular.

The evaluation results are shown in Table 1. As seen, the MCCA and PLS performs the worst, *e.g*. the rank-1 recognition rate is only about 26%$\sim$40% for recognition between 90° and 0°, which is mainly due to no consideration of supervised information. So, a straightforward idea is to apply a supervised method after them. In this work we apply the Linear Discriminant Analysis (LDA) [4] after the MCCA and PLS, denoted as MCCA+LDA and PLS+LDA respectively. As expected it perform better than both MCCA and PLS. However, as MCCA/PLS and LDA are learnt separately, some discriminancy may be lost in MCCA/PLS which cannot be recalled in the following LDA. Furthermore, the

Table 3. Performance of face recognition across view angle on MultiPIE with 7 poses.

| Methods | -45° | -30° | -15° | 15° | 30° | 45° |
|---|---|---|---|---|---|---|
| CCA[11] | 0.732 | 0.959 | 1.00 | 0.999 | 0.961 | 0.688 |
| KCCA(RBF)[3] | 0.801 | 0.977 | 0.999 | 1.00 | 0.979 | 0.717 |
| FIP+LDA[36]* | 0.934 | 0.964 | 1.00 | 0.985 | 0.956 | 0.898 |
| MVP+LDA[37]* | 0.934 | 1.00 | 1.00 | 1.00 | 0.993 | 0.956 |
| MvDN(Ours) | 0.991 | 0.995 | 1.00 | 1.00 | 0.991 | 0.976 |

MvDA and GMA perform better than MCCA+LDA, which are benefited from the simultaneous consideration of the view discrepancy and discriminancy. MvDA and GMA are linear, so we tried to make them non-linear for a better performance via the kernel trick, however we hardly got a promising result. Moreover, our MvDN performs the best, with significant improvement even up to 13% for large pose, *e.g.* 90°. This is because that the pose varies non-linearly, especially for large degree, leading to much more challenging view discrepancy, so the existing linear methods, e.g. MvDA and GMA, are unable to model so large discrepancy, while our MvDN is flexible to model high non-linearity effectively, benefited from the advantages of the deep network. Another observation is that the improvement of MvDN compared to the existing methods becomes larger as the view discrepancy expands, further demonstrating the good robustness of MvDN to view discrepancy.

We also compare the proposed MvDN with the deep unsupervised methods FIP and MVP in Table 3. The methods including CCA , KCCA and our MvDN are evaluated following the same protocol as [37][38], *i.e.* the first 200 subjects are used for training and the rest 137 are used for testing, but with much less sample per subject. The results of FIP and MVP are directly from [37] and [38] for reference and thus are marked with superscript of $*$ in Table 3.

Besides, we evaluate the performance of MvDN w.r.t. the number of neurons in the common-subnetwork. For the MvDN on this dataset, the second from the last is actually the first layer in the common sub-network $\mathbf{g}_c$, and this layer captures the variations from all views serving as the bases of the representation in last layer of $\mathbf{g}_c$. So, the final performance is essentially influenced by this layer which heavily depends on the number of neurons. Therefore, we mainly investigate the number of neurons in the second layer from last. The results are shown in Table. 2, and the averaged rank-1 recognition rate is also shown in Figure 2. As seen, MvDN can achieve a better performance even if with only 400 neurons in the second layer from last, and can achieve a further improvement with more neurons, *e.g.* 1200, 1400. However, it begins to degenerate when the number of neurons is too large to overfit.

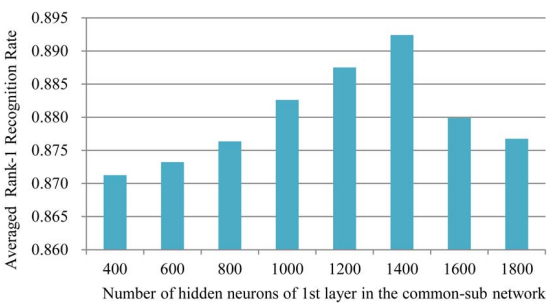


Figure 2. Averaged recognition rate of MvDN with different neurons for the $1^{st}$ in the common-subnetwork on MultiPIE dataset.

### 3.2. Face recognition across feature type

In some scenarios, different types of feature are favorable for different views. For example, in scenario of images vs. video, intensity and covariance of intensity are preferred for representing the images and videos respectively, or different lighting pre-processing are preferred for different images. In these scenarios, classification is conducted across feature type. To simulate face recognition across feature type, we firstly conduct experiment on the Face Recognition Grand Challenge (FRGC) [26] with two views, *i.e.* intensity feature (64x80=5120 dim) as one view and Local Binary Pattern (LBP) [1] feature (8850 dim) as another view.

Face Recognition Grand Challenge (FRGC) [26] is a large-scale face recognition evaluation system. It presents six challenging experiments along with data corpus of 50,000 recordings, taken under both controlled and uncontrolled conditions. We follow the protocol of the challenging Experiment 4 to evaluate our approach. In standard Experiment 4, training set consists of 12,776 images of 222 subjects, the target set consists of 16,028 controlled images, and the query set consists of 8,014 uncontrolled images. In our experiments, we randomly select 50% of the training images for training, and 25% of the target and query images for testing, *i.e.* 6388 training images, 4007 target images, and 2004 query images.

For training images, both intensity and LBP feature are available, each type of feature as one view. For target images, only intensity feature is provided, and for query images, only LBP feature is provided. So, the task is to do face verification between the target and query images with different type of features respectively. The performance is measured in terms of ROC curve.

Similarly as that in the experiment on MultiPIE, CCA, CCA+LDA, PLS, MvDA and GMA is tuned to report the best result. In our MvDN, each of two view-specific subnetworks $\mathbf{f}_i$ is a 300-300 network, and the common subnetwork $\mathbf{g}_c$ is a 300-1000-200 network.

All methods are evaluated in Figure 3. The same conclusion can be obtained as that on the face recognition across pose. The unsupervised CCA and PLS perform the worst followed by CCA+LDA. Furthermore, MvDA and GMA perform much better. Finally, the proposed MvDN performs the best, with a significant improvement, benefited from the deep non-linear and discriminant architecture.

Following the protocol on FRGC, we further compare the linear and deep methods on a more challenging dataset LFW [12]. The LFW is a large data set consisting of 13,233 uncontrolled images from 5,749 individuals. On this dataset, only those subjects with more than 1 image are employed. Among these subjects sorted according to the number of images of each subject, the first 1000 subjects are used for testing, one image of each subject as target and the rest as query. The rest 680 subjects are used for training.

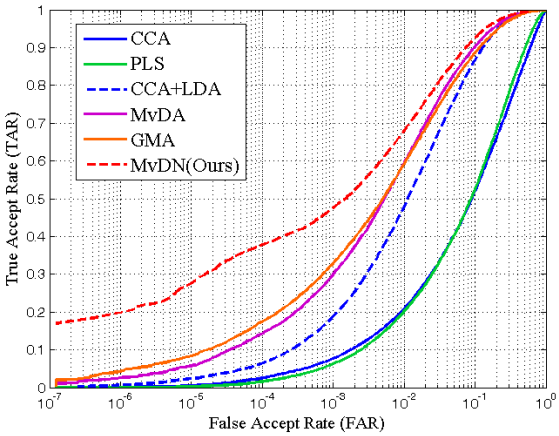


Figure 3. Face verification across feature type on FRGC dataset in terms of ROC.

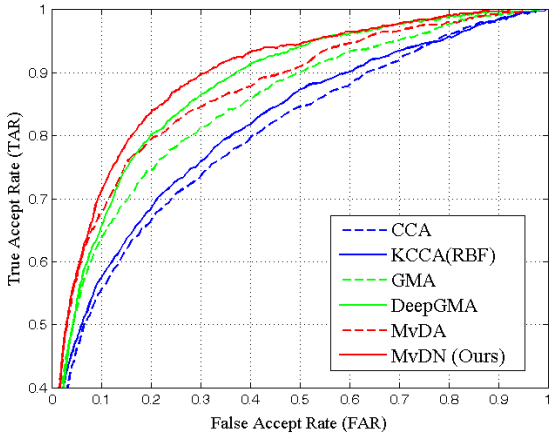


Figure 4. Face verification across feature type on LFW dataset in terms of ROC.

In total, there are 1000 images from 1000 subjects in the Target set, 1221 images from the same 1000 subjects in the Query set, 6943 images from the rest 680 subjects in the Training set. We conduct experiment on the LFW with two views, i.e. intensity feature (120x80=9600 dim) as one view and LBP feature (8850 dim) as another view. For training images, both intensity and LBP feature are available, each type of feature as one view. For target images, only intensity feature is provided, and for query images, only LBP feature is provided. Same as that on FRGC, the task is to do face verification between the target and query images with different type of features respectively. The performance is measured in terms of ROC curve.

The evaluation results are as shown in Figure 4. In Figure 4, the linear methods for evaluation include CCA, GMA and MvDA, denoted in dashed lines. For the deep methods we attempt to evaluate the Deep CCA, Deep GAM and our MvDN which share similar objective as the linear methods. However, we cannot get a reasonable performance for Deep CCA even with the authors' released code, so we instead evaluate the Kernel CCA. Besides, as GMA has no deep version, we extend it as Deep GMA following our MvDN scheme, i.e., replace the objective in our MvDN with the GMA objective but with the same deep architecture. As can be seen, both MvDN and Deep GMA perform much better than the linear MvDA and GMA, and our MvDN outperforms Deep GMA, demonstrating the effectiveness of our deep multi-view scheme.

## 4. Conclusions and Future Works

In this work, we propose a multi-view deep network, which attempts to learn a discriminant and view-invariant representation shared between multiple views. The MvDN consist of two sub-networks, the view-specific sub-network that endeavors to eliminate the discrepancy between each view and the commonality, and the common sub-network with the Fisher loss further aims for a discriminant and view-invariant representation. As evaluated, the proposed MvDN achieves quite promising performance, with significant improvement. In future, we will explore how to extend this framework for feature fusion.

## 5. Acknowledgements

This work was partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61402443, 61222211, 61532018, and the Strategic Priority Research Program of the CAS (Grant XDB02070004).

## References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 469–481. 2004.

[2] S. Akaho. A kernel method for canonical correlation analysis. In *International Meeting of Psychometric Society (IMPS)*, 2001.

[3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning (ICML)*, 2013.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):711–720, 1997.

[5] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou. Regularized latent least square regression for cross pose face recognition. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1247–1253, 2013.

[6] N. Chen, J. Zhu, and E. P. Xing. Predictive subspace learning for multi-view data: A large margin approach. In *Advances in Neural Information Processing Systems (NIPS)*.

[7] T. Diethe, D. R. Hardoon, and J. Shawe-Taylor. Multiview fisher discriminant analysis. In *Advances in Neural Information Processing Systems Workshop(NIPSW)*, 2008.

[8] T. Diethe, D. R. Hardoon, and J. S. Taylor. Constructing nonlinear discriminants from multiple data views. In *European Conference on Machine learning and knowledge discovery in databases*, 2010.

[9] R. Gross, I. Matthews, J. Cohn, T. kanada, and S. Baker. The cmu multi-pose, illumination, and expression (multi-pie) face database. Technical report, Carnegie Mellon University Robotics Institute. TR-07-08, 2007.

[10] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936.

[11] D.-A. Huang and Y.-C. F. Wang. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *International Conference on Computer Vision (ICCV)*, 2013.

[12] G. B. Huang, Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007.

[13] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen. Multi-view discriminant analysis. In *European Conference on Computer Vision (ECCV)*, pages 808–821, 2012.

[14] F. Keinosuke. Introduction to statistical pattern recognition. *Academic Press, Boston*, 1990.

[15] T.-K. Kim, J. Kittler, and R. Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *European Conference on Computer Vision (ECCV)*, pages 251–262, 2006.

[16] B. F. Klare, Z. Li, and A. K. Jain. Matching forensic sketches to mug shot photos. *IEEE Transactions on Pattern Analysis And Machine Intelligence (TPAMI)*, 33(3):39–646, 2011.

[17] Z. Lei and S. Z. Li. Coupled spectral regression for matching heterogeneous faces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1123–1128, 2009.

[18] W. Li and X. Wang. Locally aligned feature transforms across views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[19] D. Lin and X. Tang. Inter-modality face recognition. In *European Conference on Computer Vision (ECCV)*, pages 13–26, 2006.

[20] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma. A nonlinear approach for face sketch synthesis and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1005–1010, 2005.

[21] Y. Ma, S. Lao, E. Takikawa, and M. Kawade. Discriminant analysis in correlation similarity measure space. In *International Conference on Machine Learning (ICML)*, pages 577–584, 2007.

[22] T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks (ICANN)*, pages 353–360. 2001.

[23] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, and C. Suen. Unsupervised feature learning and deep learning tutorial, 2005. http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial.

[24] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, 2011.

[25] A. A. Nielsen. Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data. *IEEE Transactions on Image Processing (TIP)*, 11(3):293–305, 2002.

[26] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, volume 1, pages 947–954, 2005.

[27] J. Rupnik and J. Shawe-Taylor. Multi-view canonical correlation analysis. In *Slovenian KDD Conference on Data Mining and Data Warehouses (SiKDD)*, pages 1–4, 2010.

[28] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 2011.

[29] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[30] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2222–2230, 2012.

[31] T. Sun, S. Chen, J. Yang, and P. Shi. A novel method of combined feature extraction for recognition. In *IEEE International Conference on Data Mining (ICDM)*, pages 1043–1048, 2008.

[32] X. Tang and X. Wang. Face sketch recognition. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 14(1):50–57, 2004.

[33] K. Wang, R. He, W. Wang, L. Wang, and T. Tan. Learning coupled feature spaces for cross-modal matching. In *International Conference on Computer Vision (ICCV)*, 2013.

[34] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(1):40–51, 2007.

[36] W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 513–520, 2011.

[37] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *IEEE International Conference on Computer Vision (ICCV)*, pages 113–120, 2013.

[38] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 217–225, 2014.