Deep Tensor CCA for Multi-view Learning

Hok Shing Wong, Li Wang, Raymond Chan and Tieyong Zeng

Abstract—We present Deep Tensor Canonical Correlation Analysis (DTCCA), a method to learn complex nonlinear transformations of multiple views (more than two) of data such that the resulting representations are linearly correlated in high order. The high-order correlation of given multiple views is modeled by covariance tensor, which is different from most CCA formulations relying solely on the pairwise correlations. Parameters of transformations of each view are jointly learned by maximizing the high-order canonical correlation. To solve the resulting problem, we reformulate it as the best sum of rank-1 approximation, which can be efficiently solved by existing tensor decomposition method. DTCCA is a nonlinear extension of tensor CCA (TCCA) via deep networks. Comparing with kernel TCCA, DTCCA not only can deal with arbitrary dimensions of the input data, but also does not need to maintain the training data for computing representations of any given data point. Hence, DTCCA as a unified model can efficiently overcome the scalable issue of TCCA for either high-dimensional multi-view data or a large amount of views, and it also naturally extends TCCA for learning nonlinear representation. Extensive experiments on four multi-view data sets demonstrate the effectiveness of the proposed method.

Index Terms—Multi-view learning, canonical correlation analysis, tensor decomposition, deep networks

1 Introduction

Multi-view learning [1], [2] has been receiving increased attention in many scientific domains since data sets are usually sampled from diverse variables of each object. Due to their heterogeneous properties, these variables can be naturally partitioned into groups. Each group of variables is referred to as a view. Such data sets with multiple views collectively are referred to as multi-view data sets, such as text content of each web page and the anchor text of other web pages linking to this page in web page classification [3] and various descriptors used to extract features of each image for image classification [4].

Subspace-based multi-view learning as one of the most representative categories in the multi-view learning paradigm has been extensively studied for high-dimensional multi-view data sets [2]. It aims to obtain a latent subspace shared by multiple views based on the assumption that each view of the data is generated from the unknown distribution conditioned on the same latent subspace [5]. The "curse of dimensionality" can be effectively alleviated by learning a latent subspace with the dimensionality less than any of the input views. Canonical correlation analysis (CCA), originally designed for measuring the linear correlation between two sets of variables [6], was formally introduced as a multi-view dimensionality reduction method in [7] for its ability of reducing the labeled instance complexity under certain weak assumptions. Another appealing property of CCA is that the learned subspace will not contain the noise in the uncorrelated dimensions if there is noise in either view that is uncorrelated to the other view [8]. A reformulation of CCA [9] was proposed to facilitate the learning of sparse projection matrices, but the results are not straightforwardly generalized to more than two views. In the last decade, CCA has received a renewed interest in the machine learning community [8], [10], [11], and in many scientific fields its usefulness and those of its variants have already been well demonstrated [12]. Moreover, its capability could also be extended to domain adaption [13] and heterogeneous cloud computing [14], [15].

We in this paper are particularly interested in the multi-view data sets with more than two views and the inherent nonlinear property between the latent subspace and the input views. Two representative nonlinear representation techniques have been applied to CCA for two views: kernel trick [10] and deep learning [8]. Kernel CCA (KCCA) [10] extends CCA for finding maximally correlated nonlinear projections in reproducing kernel Hilbert space (RKHS) [16]. The nonlinearity of KCCA is represented by kernel function, so this representation is limited by the fixed kernel. Moreover, the kernel trick increases the time complexity for learning the projections and computing the representation of new data points since it scales poorly with the size of the training data. To overcome the above drawbacks, deep CCA (DCCA) [8] was proposed by simultaneously learning two deep nonlinear mappings of two views that are maximally correlated. Since the deep networks are parametric and not limited to RKHS, it does not face the above drawbacks of kernel trick, and they have showed the empirical success on various tasks [17]. Some variants of DCCA have been studied including deep canonically correlated autoencoders [18] by simultaneously maximizing canonical correlation and minimizing the reconstruction errors of the autoencoders, deep variational CCA [19] extended from variational autoencoders [20] based on the probabilistic CCA model [5],

- *Hok Shing Wong is with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong. E-mail: hswong@math.cuhk.edu.hk.*
- *Li Wang is with the Department of Mathematics and Department of Computer Science and Engineering, University of Texas at Arlington, Texas, 76019 USA. Supported in part by NSF DMS-2009689. E-mail: li.wang@uta.edu. Corresponding Author.*
- *Raymond Chan is with the Department of Mathematics, City University of Hong Kong, Hong Kong. Supported in part by HKRGC Grants No. CUHK14306316, CUHK14301718, CityU11301120, CityU Grant 9380101, CRF Grant C1007-15G, AoE/M-05/12. E-mail: rchan.sci@cityu.edu.hk.*
- *Tieyong Zeng is with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong. Supported in part by NSFC Grant No. 11671002, CUHK start-up and CUHK DAG 4053342, RGC 14300219, RGC 14302920 and NSFC/RGC N_CUHK 415/19. E-mail: zeng@math.cuhk.edu.hk.*

and deep discriminative CCA [21] by considering one labeled data as one view in the setting of supervised learning.

The aforementioned CCA variants are mainly designed for data sets of two views. Various learning criteria have been proposed to extend CCA for more than two views. In work [11], [22], five multiset correlation formulations and four sets of constraints are discussed for multiset CCAs. Among them, CCA with the sum of pairwise correlations (SUMCOR) criterion enjoys a nice analytic solution by generalized eigen-decomposition, which was reformulated as the least square problem in order to develop an adaptive learning algorithm [23]. Generalized CCA (GCCA) [24] takes a different perspective by learning a common representation and imposing orthogonality on the common representation. We will build the connection between GCCA and CCA with SUMCOR and show in Section 2 that GCCA does not really maximize the canonical correlation so it does not reduce to CCA for two views. For more than two views, multiset CCA and GCCA can only capture the pairwise correlations. To generalize CCA for handling more than two views, tensor CCA (TCCA) [25] was proposed by maximizing the high-order correlation represented by the covariance tensor [26] over the data sets from all views, so it is a natural way to extend CCA for arbitrary number of views. We also note that tensor technique has been used for multi-view representation learning [27], [28], but affinity matrix based tensor is very different from the one used in TCCA, and they are often specifically designed for clustering problem. Nonlinear extensions of these multi-view CCAs have also been explored. Kernel TCCA [25] extends CCA based on kernel trick, so it encounters the same drawbacks as KCCA. Deep multiset CCA [29] extends multiset CCA via deep networks, but it only can deal with very special case of multi-view data sets where views have to be sampled from the same input space. Deep GCCA (DGCCA) [30] extends GCCA via deep networks, but it does not reduce to DCCA for two views.

In this paper, we propose deep TCCA (DTCCA) by extending TCCA for learning nonlinear projections via deep networks. DTCCA not only inherits the high-order canonical correlation of multiple views but also overcomes the drawbacks brought by kernel TCCA. The main contributions of this paper are summarized as follows:

- We build the connections among three representative categories of existing CCAs for more than two views and their nonlinear generalizations. Based on the connections, the pros and cons of these methods are discussed in detail.
- We further propose DTCCA model which can simultaneously learn the nonlinear projections and TCCA via deep networks. Comparing to kernel TCCA, DTCCA can effectively overcome the drawbacks caused by kernel function and make TCCA practical for large-scale and high-dimensional multi-view data sets. To the best of our knowledge, there is no prior work on the nonlinear generalization of TCCA via deep networks.
- Extensive experiments are performed on four multi-view data sets by comparing with various representative baselines in terms of various settings including varied views, the dimensions of latent subspace, and the ratios of training data. Moreover, the impact on the number of layers of networks are also investigated. Experimental results show that DTCCA significantly outperform TCCA and other methods especially on small amount of training data, and it shows consistent better results over data sets with more than two views and varied latent dimensions.

TABLE 1
Notation and Definitions

Notation	Definition
n	the number of samples
k	the number of views
m	the dimension of the common latent space
$\mathbf{x}_r^i \in \mathbb{R}^{d_r}$	the ith data of the rth view in d_r-dimensional space
$X_r \in \mathbb{R}^{d_r \times n}$	the matrix representation of the rth view
$P_r \in \mathbb{R}^{d_r \times m}$	the projection matrix of view r, and the l th column is $\mathbf{p}_i^l \in \mathbb{R}^{d_r}$.
$\mathbf{z}_r^l \in \mathbb{R}^n$	the canonical variable of the rth view projected onto base $\mathbf{p}_i^l$
$C_{r,s}$	the intra-view covariance of view r and view s
f_r	the transformation function of the rth view
$\mathcal{C}$	the k-order tensor of size $d_1 \times \ldots \times d_k$

In the following of this paper, we first review CCA and their multi-view extensions including nonlinear generalization. In Section 3, the proposed model is presented, and the optimization problem is reformulated as tensor decomposition. Extensive experiments are conducted in Section 4. Finally, we draw our conclusions in Section 5.

2 CCA AND ITS MULTI-VIEW EXTENSIONS

CCA was originally proposed to find a pair of bases for two set of variables so that their corresponding projected variables onto these bases are maximally correlated [6]. Its generalization to multiple data sets (more than two views) has been widely studied due to the emerging of multi-view data sets in various real world applications. In this section, we will review three representative multi-view CCA methods and their nonlinear extensions according to their inherent criteria.

Denote by $\{(\mathbf{x}_1^i, \ldots, \mathbf{x}_k^i)\}_{i=1}^n$ the data sets of k views with n data points, where the ith data of the rth view is $\mathbf{x}_r^i \in \mathbb{R}^{d_r}$ and d_r is the dimension of the rth view. Let $X_r = [\mathbf{x}_r^1, \ldots, \mathbf{x}_r^n] \in \mathbb{R}^{d_r \times n}$ be the matrix representation of the rth view data set and $P_r = [\mathbf{p}_r^1, \ldots, \mathbf{p}_r^m] \in \mathbb{R}^{d_r \times m}$ be the projection matrix for the rth view consisting of m bases in d_r-dimensional space. Denote $P = [P_1; \ldots; P_k] \in \mathbb{R}^{d \times m}$ with $d = \sum_{r=1}^k d_r$. Without loss of the generality, we assume that data of each view is centered, that is, $X_r \mathbf{1}_n = \mathbf{0}_{d_r}, \forall r = 1, \ldots, k$, where $\mathbf{1}_n$ is an n-dimensional column vector of all 1s and $\mathbf{0}_{d_r}$ is a d_r-dimensional column vector of all 0s. The cross-view covariance between view r and view s is defined as $C_{r,s} = X_r X_s^T \in \mathbb{R}^{d_r \times d_s}$ and the intra-view covariance of view r is defined as $C_{r,r} = X_r X_r^T \in \mathbb{R}^{d_r \times d_r}$. $\|A\|_F$ is the Frobenius norm of matrix A. For the ease of reference, we summarize some important notation and definitions in Table 1, which will be used throughout the whole paper.

2.1 Multiset CCA

Multiset CCA (MCCA) [11], [22] has been studied for analyzing linear relations between more than two views, and

various formulations have been explored. The straightforward extension of CCA to multiset CCA is to maximize the sum of the pairwise correlations (CCA-SUMCOR):

$$\max_{\{P_r\}_{r=1}^k} \sum_{r=1}^{k}\sum_{s=1}^{k} \text{tr}(P_r^T C_{r,s} P_s) \text{ s.t.} \sum_{r=1}^{k} P_r^T C_{r,r} P_r = I_m, \quad (1)$$

where the orthogonal constraint over the projected data is added to prevent trivial solution. For $k = 2$, problem (1) is reduced to the conventional CCA problem. The Langrange multiplier technique can be used to solve the above constrained maximization problem. With the multiplier diagonal matrix Λ, we can formulate the Lagrangian function $L(\{P_r\}, \Lambda)$ as

$$\sum_{r=1}^{k}\sum_{s=1}^{k} \text{tr}(P_r^T C_{r,s} P_s) - \text{tr}\left(\Lambda(\sum_{r=1}^{k} P_r^T C_{r,r} P_r - I_m)\right). \quad (2)$$

The optimality condition is

$$\sum_{s=1}^{k} C_{r,s} P_s = C_{r,r} P_r \Lambda, \forall r = 1, \ldots, s, \quad (3)$$

which is equivalent to the following matrix representation

$$\begin{bmatrix} C_{1,1} & C_{1,2} & \ldots & C_{1,k} \\ C_{2,1} & C_{2,2} & \ldots & C_{2,k} \\ \vdots & \vdots & \ddots & \ldots \\ C_{k,1} & C_{k,2} & \ldots & C_{k,k} \end{bmatrix} P = \begin{bmatrix} C_{1,1} & 0 & \ldots & 0 \\ 0 & C_{2,2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \ldots \\ 0 & 0 & \ldots & C_{k,k} \end{bmatrix} P\Lambda. \quad (4)$$

It is well-known that the optimal projections $\{P_r\}_{r=1}^k$ can be obtained by solving the generalized eigenvalue decomposition problem (4). Moreover, problem (1) can be equivalently rewritten as

$$\min_{\{P_r\}} \sum_{r=1}^{k}\sum_{s=1}^{k} \|P_r^T X_r - P_s^T X_s\|_F^2 \text{ s.t.} \sum_{r=1}^{k} P_r^T C_{r,r} P_r = I_m, \quad (5)$$

since $\sum_{r=1}^{k}\sum_{s=1}^{k} \|P_r^T X_r\|_F^2 = k\text{tr}(\sum_{r=1}^{k} P_r^T C_{r,r} P_r) = km$ is a constant. The pairwise least square formulation (5) (LSCCA) was proposed to develop an adaptive learning algorithm based on the recursive least squares [23]. Another reformulation can be achieved by introducing the average representation of the k views denoted by

$$M = \frac{1}{k}\sum_{r=1}^{k} P_r^T X_r, \quad (6)$$

and we have the following equalities:

$$\begin{aligned} &\sum_{r=1}^{k}\sum_{s=1}^{k} \|P_r^T X_r - P_s^T X_s\|_F^2 \\ =&\sum_{r=1}^{k}\sum_{s=1}^{k} \|(P_r^T X_r - M) - (P_s^T X_s - M)\|_F^2 \\ =&2k\sum_{r=1}^{k} \|P_r^T X_r - M\|_F^2. \end{aligned} \quad (7)$$

Accordingly, the reformulated problem of (1) based on (6) and (7) is written as

$$\min_{\{P_r\}_{r=1}^k, M} \sum_{r=1}^{k} \|P_r^T X_r - M\|_F^2 \text{ s.t.} \sum_{r=1}^{k} P_r^T C_{r,r} P_r = I_m. \quad (8)$$

The equivalence between LSCCA and CCA-MAXVAR [22] has been proved in [23], so (8) is also equivalent to LSCCA and CCA-MAXVAR. Problem (1) with alternating constraints $P_r^T C_{r,r} P_r = I_m, \forall r$ was also explored in [11], but it loses the nice analytic solution (4) that (1) has. Moreover, the supervised extension of MCCA has also been explored by incorporating label data via linear discriminant analysis [31], [32].

Deep CCA (DCCA) [8] is proposed to learn two nonlinear transformations f_1 and f_2 by simultaneously maximizing the correlation between two views:

$$\max_{f_1, f_2, P_1, P_2} \frac{\text{tr}(P_1^T \hat{C}_{1,2} P_2)}{\sqrt{\text{tr}(P_1^T \hat{C}_{1,1} P_1)\text{tr}(P_2^T \hat{C}_{2,2} P_2)}}, \quad (9)$$

where f_1 and f_2 can be multiple stacked layers of nonlinear transformations with output dimension as m, $\hat{C}_{1,2} = \hat{f}_1(X_1)\hat{f}_2(X_2)^T$, $\hat{C}_{r,r} = \hat{f}_r(X_r)\hat{f}_r(X_r)^T, \forall r = 1, 2$, and $\hat{f}_r(X_r) = f_r(X_r)H$ is a centered transformed matrix with centering matrix $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \in \mathbb{R}^{n\times n}$. Let $\hat{P}_1 = \hat{C}_{1,1}^{1/2} P_1$ and $\hat{P}_2 = \hat{C}_{2,2}^{1/2} P_2$. Problem (9) is equivalent to the following maximization problem

$$\max_{f_1, f_2, \hat{P}_1, \hat{P}_2} \text{tr}(\hat{P}_1^T T \hat{P}_2) \text{ s.t. } \hat{P}_1^T \hat{P}_1 = \hat{P}_2^T \hat{P}_2 = I_m, \quad (10)$$

where $T = \hat{C}_{1,1}^{-1/2} \hat{C}_{1,2} \hat{C}_{2,2}^{-1/2}$. For fixed f_1 and f_2, problem (10) can be solved optimally by singular value decomposition (SVD). Let U_m and V_m be the matrices of the top m left- and right-singular vectors of T. We have the optimal solution $P_1 = \hat{C}_{1,1}^{-1/2} U_m$ and $P_2 = \hat{C}_{2,2}^{-1/2} V_m$. And the optimal objective becomes $\text{tr}(T^T T)^{1/2}$, which is a function of f_1 and f_2. The work [18] further explored the autoencoder to regulate DCCA. Unfortunately, the special reformulation (10) for CCA is not applicable for MCCA since n^2 pairs of projections are coupled. dMCCA [29] extends MCCA for nonlinear transformation via deep networks, but it only can deal with very special case that all view data sets are sampled from the same input space, that is $d_r = d_s, \forall s \neq r$, and $P_r = P_s, \forall r \neq s$. These strong assumptions prevent dMCCA from being used for general multi-view data sets.

2.2 Generalized CCA

Generalized CCA (GCCA) [24] finds $\{P_r\}_{r=1}^k$ by constructing a common representation $G \in \mathbb{R}^{m\times n}$ so that the sum of the squared losses between each view and G is minimized. GCCA is formulated as the optimization problem

$$\min_{G, \{P_r\}_{r=1}^k} \sum_{r=1}^{k} \|G - P_r^T X_r\|_F^2 \text{ s.t. } GG^T = I_m, \quad (11)$$

where the orthogonal constraint over G is added so as to prevent the trivial solution. It is worth noting that (11) resembles (8) in terms of the objective function, but the constraints are very different. Problem (11) can be transformed to an eigenvalue decomposition problem. First, given a matrix G, problem (11) with respect to P_r can be solved independently, and it is a convex quadratic programming so it can be solved globally by the first order optimality condition, that is

$$-2X_r(G - P_r^T X_r)^T = 0 \Rightarrow P_r = (X_r X_r^T)^{-1} X_r G^T. \quad (12)$$

By substituting (12) back to (11), we then reformulate (11) as

$$\max_{G} \text{tr}(GQG^T) \text{ s.t. } GG^T = I_m, \tag{13}$$

where $Q = \sum_{r=1}^{k} X_r^T (X_r X_r^T)^{-1} X_r$. Hence, the optimal solution G consists of the eigenvectors corresponding to the top m eigenvalues of Q. Once G is obtained, $\{P_r\}_{r=1}^k$ can be recovered by (12). Some other extensions of GCCA have also been explored, such as the $\ell_{2,1}$-norm regularized GCCA model is proposed to facilitate the interpretability of the learning representation [33].

Deep GCCA (DGCCA) [30] extends (11) for nonlinear multi-view learning, where the input data of each view is replaced by the transformed data via some nonlinear function $\{f_r\}_{r=1}^k$, e.g., the multi-layer perception network. This can effectively resolve the drawback of GCCA for only learning linear projections. The optimization problem is formulated as

$$\begin{aligned} \min_{G,\{P_r\}_{r=1}^k,\{f_r\}_{r=1}^k} & \sum_{r=1}^{k} \|G - P_r^T f_r(X_r)\|_F^2 \\ \text{s.t. } & GG^T = I_m. \end{aligned} \tag{14}$$

The same reformulated problem as GCCA is obtained

$$\max_{G,\{f_r\}_{r=1}^k} \text{tr}(GQ_f G^T) \text{ s.t. } GG^T = I_m, \tag{15}$$

where $Q_f = \sum_{r=1}^{k} f_r(X_r)^T (f_r(X_r) f_r(X_r)^T)^{-1} f_r(X_r)$. The gradient with respect to f_r can be calculated as $\partial_{f_r} = 2P_r G - 2P_r P_r^T f_r(X_r)$.

2.3 Tensor CCA

Tensor CCA (TCCA) [25] is proposed for multi-view learning by exploiting high-order tensor correlation among multiple views. Let $\mathbf{z}_r^l = [z_r^l(1), \ldots, z_r^l(n)]^T = X_r^T \mathbf{p}_r^l \in \mathbb{R}^n$ be the canonical variable of the rth view projected onto the lth base. The high-order canonical correlation over k views is defined as

$$\rho = \sum_{l=1}^{m} \rho_l \tag{16}$$

$$\rho_l = \sum_{i=1}^{n} \prod_{r=1}^{k} z_r^l(i), \forall l = 1, \ldots, m, \tag{17}$$

with the constraints used in [22]

$$P_r^T C_{r,r} P_r = I_m, \forall r = 1, \ldots, m. \tag{18}$$

In the case of $k = 2$, the high-order correlation is reduced to the canonical correlation, which can be verified by the following derivations:

$$\begin{aligned} \text{tr}(P_1^T X_1 X_2^T P_2) &= \sum_{l=1}^{m} (\mathbf{z}_1^l)^T \mathbf{z}_2^l \\ &= \sum_{l=1}^{m} \sum_{i=1}^{n} z_1^l(i) z_2^l(i) = \rho. \end{aligned}$$

As a result, maximizing (16) is equivalent to CCA-MAXVAR in the case of $k = 2$ and constraints (18). For $k > 2$, the high-order correlation will be captured by (16). For multi-view learning, this makes TCCA different from others based on pairwise correlations. As a result, maximizing (16) is transformed to the best sum of rank-1 approximation, e.g., the best rank-m CANDECOMP/PARAFAC decomposition [34]. The well-known alternating least squares (ALS) algorithm [35], [36] is used. The nonlinear extension of TCCA via kernel trick is also explored in [25].

2.4 Discussions and Summary

We are now ready to compare the above three representative multi-view extensions of CCA from two perspectives: learning criterion and the nonlinear extension.

As the learning criterion, their correlation definitions are different. Both MCCA and TCCA generalize CCA since they reduce to exact CCA for two views. However, GCCA does not possess this property. By comparing (8) with (11), it is easy to see that GCCA enforces orthogonality on the common representations, while MCCA takes the mean of all view representations (6) as the common representation. This implies that GCCA is suitable for visualizing multi-view data in the orthogonal coordinate space, while MCCA is good to maximize the pairwise correlation of any two views by assuming the common representation variable as the mean of projected data of all views. In contrast, TCCA is very different from MCCA for $k > 2$ and GCCA since the high-order correlations among views can be captured by TCCA, but MCCA can be only able to model the pairwise correlation and GCCA only captures the linearly transformed intra-view correlation according to (12) and (13).

Two techniques are popularly used for learning nonlinear projections of multi-view data: kernel trick and deep networks. Kernel CCA (KCCA) [10] models the nonlinear transformation via kernel functions. Kernel TCCA extends TCCA via kernel trick. However, kernel trick faces two crucial issues: restricted representation power of a fixed kernel function and the high-computational complexity for large-scale data, even though kernel learning [37] and kernel approximation [38] techniques have been studied. Fortunately, DCCA [8] effectively alleviates the two issues for nonlinear CCA by learning deep networks as the nonlinear transformation functions. For multi-view CCAs, the nonlinear representation learning is still limited. For example, dMCCA is only applicable for special data sets and DGCCA does not align well with correlation maximization.

In summary, the extension of TCCA for learning nonlinear projections using deep networks not only inherits the advantages of TCCA but also makes nonlinear representation practical by deep networks. To the best of our knowledge, there is no such a prior work. In this paper, we will propose a novel deep version of TCCA to fill up this gap.

3 Deep Tensor CCA

In this paper, we propose Deep Tensor CCA (DTCCA), which computes the representations of multiple views by passing them through multi-layer perception (MLP) networks with layers of nonlinear transformations, and the networks are tuned automatically by maximizing the high-order canonical correlation (16). Fig. 1 illustrates the workflow of DTCCA with a data set consisting of three views.

Without loss of the generality, we assume that the ith intermediate layer in the network for the rth view has c_r^i

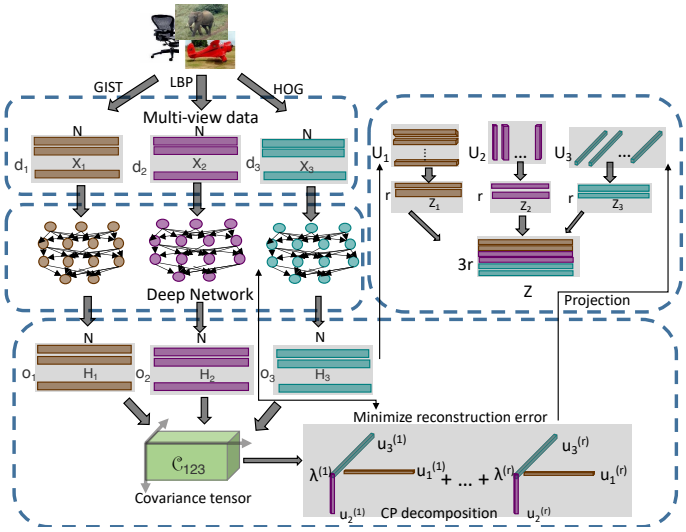

Fig. 1. The workflow of the proposed DTCCA model by maximizing three-order covariance defined by covariance tensor via independent deep networks with illustrated example consisting of three views from descriptors GIST, LBP and HOG.

units, and the output layer has m units. The output of the first layer for the input data $\mathbf{x}_r$ from the rth view is $h_r^1 = \sigma(W_r^1\mathbf{x}_r + b_r^1) \in \mathbb{R}^{c_r^1}$, where $W_r^1 \in \mathbb{R}^{c_r^1 \times d_r}$ is the weight matrix, $b_r^1 \in \mathbb{R}^{c_r^1}$ is the vector of biases, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function. The output h_r^1 can then be used as the input to the next layer for computing the output $h_r^2 = \sigma(W_r^2 h_r^1 + b_r^2) \in \mathbb{R}^{c_r^2}$, and this is recursively constructed v times until the final output $f_r(\mathbf{x}_r) = \sigma(W_r^v h_r^{v-1} + b_r^v) \in \mathbb{R}^m$ is reached. The same construction process can be used for each of the k views. As a result, we have a set of nonlinear functions $\{f_r\}_{r=1}^k$ with the number of intermediate layers v and their associated parameters $\{W_r^i, b_r^i\}, \forall r = 1, \ldots, k, i = 1, \ldots, v$. To simplify the annotation, we assume f_r implicitly associates to its network parameters.

With the above defined nonlinear transformation $\{f_r\}_{r=1}^m$, DTCCA aims to maximize the high-order canonical correlation by solving the following optimization problem

$$\begin{aligned} \max_{\{f_r\}_{r=1}^k, \{P_r\}_{r=1}^k} & \sum_{l=1}^{m}\sum_{i=1}^{n}\prod_{r=1}^{k} z_r^l(i) \qquad (19) \\ \text{s.t. } & \mathbf{z}_r^l = \hat{f}_r(X_r)^T\mathbf{p}_r^l, \forall r = 1, \ldots, k, l = 1, \ldots, m, \\ & P_r^T \hat{f}_r(X_r)\hat{f}_r(X_r)^T P_r = I_m, \forall r = 1, \ldots, k, \\ & \hat{f}_r(X_r) = f_r(X_r)H, \forall r = 1, \ldots, k, \end{aligned}$$

where $\hat{f}_r(X_r) \in \mathbb{R}^{m \times n}$ is the centered matrix of $f_r(X_r)$. In order to jointly optimize the network and TCCA, we will first transform (19) to the best rank-m tensor decomposition problem. Define the covariance tensor of the network output data $\{\hat{f}_r(X_r)\}_{r=1}^k$ as a k-order tensor of size $d_1 \times \ldots \times d_k$

$$\mathcal{C} = \sum_{i=1}^{n} \hat{f}_1(\mathbf{x}_1^i) \circ \hat{f}_2(\mathbf{x}_2^i) \circ \ldots \circ \hat{f}_k(\mathbf{x}_k^i) \qquad (20)$$

where $\circ$ is the outer product of vectors. Let $U \in \mathbb{R}^{p \times d_r}$ be a matrix. The r-mode product of tensor $\mathcal{C}$ and U is defined as a tensor $\mathcal{A} = \mathcal{C} \times_r U$ of size $d_1 \times \ldots d_{r-1} \times p \times d_{r+1} \times \ldots \times d_k$ with element

$$\mathcal{A}(i_1, \ldots i_{r-1}, j_r, i_r, \ldots, i_k) = \sum_{i_r=1}^{d_r} \mathcal{C}(i_1, \ldots, i_k)U(j_r, i_r).$$

The high-order canonical correlation (17) can be rewritten as

$$\sum_{i=1}^{n}\prod_{r=1}^{k} z_r^l(i) = \mathcal{C} \times_1 (\mathbf{p}_1^l)^{\mathrm{T}} \times_2 (\mathbf{p}_2^l)^{\mathrm{T}} \ldots \times_k (\mathbf{p}_k^l)^{\mathrm{T}}. \qquad (21)$$

Similar to DCCA, the orthogonal constraints in (19) can be rewritten as, $\forall r = 1, \ldots, k$

$$(\mathbf{p}_r^l)^{\mathrm{T}}\hat{C}_{r,r}\mathbf{p}_r^{l'} = \begin{cases} 1, & l = l' \\ 0, & \text{otherwise.} \end{cases} \qquad (22)$$

where $\hat{C}_{r,r} = \hat{f}_r(X_r)\hat{f}_r(X_r)^T, \forall r = 1, \ldots, k$. Let $\mathbf{u}_r^l = \hat{C}_{r,r}^{1/2}\mathbf{p}_r^l$ and $U_l = [\mathbf{u}_r^1, \ldots, \mathbf{u}_r^m] \in \mathbb{R}^{m \times m}$. We can further reformulate (19) as

$$\begin{aligned} \max_{\{f_r\}_{r=1}^k, \{U_r\}_{r=1}^k} & \sum_{l=1}^{m} \mathcal{M} \times_1 (\mathbf{u}_1^l)^{\mathrm{T}} \times_2 (\mathbf{u}_2^l)^{\mathrm{T}} \ldots \times_k (\mathbf{u}_k^l)^{\mathrm{T}} \qquad (23) \\ \text{s.t. } & \mathcal{M} = \mathcal{C} \times_1 C_{1,1}^{-1/2} \times_2 C_{2,2}^{-1/2} \ldots \times_k C_{k,k}^{-1/2} \\ & (\mathbf{u}_r^l)^{\mathrm{T}}\mathbf{u}_r^{l'} = \begin{cases} 1, & l = l' \\ 0, & \text{otherwise.} \end{cases} \\ & \hat{f}_r(X_r) = f_r(X_r)H, \forall r = 1, \ldots, k. \end{aligned}$$

Problem (23) consists of the best sum of rank-1 approximation, e.g., the best rank-m CANDECOMP/PARAFAC decomposition [34]. This is given by

$$\hat{\mathcal{M}} = \sum_{l=1}^{m} \lambda_l \mathbf{u}_1^l \circ \mathbf{u}_2^l \ldots \circ \mathbf{u}_k^l. \qquad (24)$$

According to [39], the sum of rank-1 decomposition and orthogonality constraints of high-order tensor cannot be satisfied simultaneously. Although U_r is not enforced to be orthogonal, $(\mathbf{u}_r^l)^{\mathrm{T}}\mathbf{u}_r^l = 1$ holds. Hence, we resort to solving an approximation problem given by

$$\begin{aligned} \min_{\{f_r\}_{r=1}^k, \{U_r\}_{r=1}^k} & \|\mathcal{M} - \hat{\mathcal{M}}\|_F^2 \qquad (25) \\ \text{s.t. } & \mathcal{M} = \mathcal{C} \times_1 C_{1,1}^{-1/2} \times_2 C_{2,2}^{-1/2} \ldots \times_k C_{k,k}^{-1/2} \\ & (\mathbf{u}_r^l)^{\mathrm{T}}\mathbf{u}_r^l = 1, \forall r = 1, \ldots, k, l = 1, \ldots, m, \\ & \hat{f}_r(X_r) = f_r(X_r)H, \forall r = 1, \ldots, k, \end{aligned}$$

where $\|\mathcal{M}\|_F$ is the Frobenius norm of the tensor $\mathcal{M}$. Given an $\mathcal{M}$, problem (25) with respect to $\{U_r\}_{r=1}^k$can be solved by the ALS algorithm [35], [36]. The parameters of networks are then updated by minimizing the square loss. It is worth noting that for $k = 2$, TCCA is equivalent to CCA as shown in Section 2.3, so DTCCA with $k = 2$ is reduced to DCCA. Algorithm 1 for $k = 2$ provides an alternative approach for solving DCCA since ALS algorithm obtains an approximate solution, while DCCA takes singular value decomposition during the network learning. However, DCCA approach does not work for $k > 2$. We also notice that the recent SVD-based algorithms [40], [41] on solving (25) by alternating two factors simultaneously have better convergence than ALS with one factor a time. ALS is more compatible with deep neural networks than SVD-based algorithms for computing

Algorithm 1 Deep tensor CCA (DTCCA)

Input: data sets of k views: $\{X_r \in \mathbb{R}^{d_r \times n}\}_{r=1}^k$
Initialize the networks
for $i = 1$ to $epoch$ **do**
  compute $\{f_r(X_r)\}_{r=1}^k$ and then $\{C_{r,r}\}_{r=1}^k$
  construct tensor $\mathcal{M}$
  solve rank-m tensor decomposition using ALS
  form $\hat{\mathcal{M}}$ based on the solutions $\{\lambda_l\}_{l=1}^m$ and $\{U_r\}_{r=1}^k$
  compute gradients of loss $\|\mathcal{M} - \hat{\mathcal{M}}\|_F^2$ with respect to network weights $\{f_r\}$ by backpropagation
  update $\{f_r\}$ by the gradient descend method
end for
$P_r = C_{rr}^{-1/2} U_r, \forall r = 1, \ldots, k$
Output: $\{f_r\}_{r=1}^k$ and $\{P_r\}_{r=1}^k$.

gradients by backpropagation in the (stochastic) gradient descent method. We empirically observed that ALS can serve our needs for good performance.

Once $\{U_r\}_{r=1}^k$ and $\{f_r\}_{r=1}^k$ are obtained after training, we can recover the canonical variables for any given test data $\mathbf{x}_r$ from the rth view by

$$\mathbf{z}_r = \hat{f}_r(\mathbf{x}_r)^T C_{rr}^{-1/2} U_r, \forall r = 1, \ldots, k. \tag{26}$$

The learning algorithm for DTCCA is presented in Algorithm 1. During the training process, we take the full-batch optimization approach, as suggested in [8] for training DCCA. We implement Algorithm 1 in *Pytorch* [42] together with package *TensorLy* [43] for tensor operation and decomposition. The Adam optimizer is used with the learning rate set to be 10^{-3}, and others are set to be default values.

Remark: As to sparse CCA in [9], it is not easy to impose sparsity on P_s in DTCCA since the solution of ALS cannot guarantee the sparsity of $C_{rr}^{-1/2} U_r$ according to (26). Hence, our DTCCA generalizes CCA with nonlinear projections, but the extension of DTCCA for sparse CCA is not straightforward.

3.1 Complexity Analysis

The time and space complexities of our proposed DTCCA model are composed by three important components.

The first component is the complexity of the k MLP networks. The forward pass for computing $f_r(X_r)$ given $X_r, \forall r = 1, \ldots, k$ and the backward pass used to update network weights via the gradient descent method have complexities depending on the number of network layers, v, and the number of neurons in each layers. At layer l of view r, the network weights is of size $c_r^l \times c_r^{l-1}$ where c_r^l is the output dimension and c_r^{l-1} is the input dimension. For n samples, the time complexity is $O(nc_r^l c_r^{l-1})$. The total complexity of performing forward operation on k MLP networks takes $O(n \sum_{r=1}^k \sum_{l=1}^v c_r^l c_r^{l-1})$ where c_r^0 is the input dimension d_r of view r. The backward pass can be performed efficiently by backpropagation.

The second component is the complexity of computing tensor $\mathcal{M}$ with size $c_1^v \times c_2^v \ldots, c_k^v$ where c_r^v is the output dimension of the last layer of the rth network. According to (25), its complexity consists of two parts: $C_{r,r}^{-1/2}$ takes $O((c_r^v)^3)$, $\mathcal{C}$ takes $O(n \prod_{r=1}^n c_r^v)$ time complexity and $O(\prod_{r=1}^v c_r^v)$ for space complexity, so $\mathcal{M}$ takes $O(\sum_{r=1}^v (c_r^v)^2 + n \prod_{r=1}^v c_r^v + \sum_{r=}^v (c_r^v)^3)$ in total.

The third component is the complexity of the ALS algorithm for the rank-m decomposition. According to [36], the time complexity of ALS is $O(tm \prod_{r=1}^v c_r^v)$, where t is the number of iterations in ALS and m is the reduced dimension.

Without loss of generality, let $c_r^v = m, \forall r = 1, \ldots, k$. ALS takes $O(km^3 + (n + tm)m^k)$, which is independent of the input dimensions $\{d_r\}_{r=1}^k$. Due to the dominate term m^k, DTCCA limits m for a large number of views. Fortunately, $m \ll \min_{r=1}^k d_r$ is generally assumed, especially for high-dimensional data in multi-view subspace learning. In Section 4, we will show that our DTCCA works well on 6 views with $m \in [2, 10]$.

3.2 Comparisons to existing works

DTCCA is built on TCCA and deep neural networks, but it actually shows many good properties comparing with TCCA and other deep learning based CCAs.

Comparing to TCCA [25], DTCCA possesses the following advantages:

1) DTCCA jointly maximizes the high-order correlation and nonlinear transformations, but TCCA only works on linear transformation.
2) DTCCA can directly take the raw input views as the input since the internal representation learning automatically transforms the low-level features to a small number of high-level abstractions, which can significantly reduce the computational and storage costs, and also guarantee good performance. However, TCCA has to rely on other dimensionality reduction method as the preprocessing step, otherwise it is infeasible due to the high computational and storage costs on high-correlation tensor constructed from input views, so that suboptimal solutions can be expected. Hence, DTCCA makes TCCA practical and better for real world data from the perspectives of computation, storage and performance.
3) DTCCA provides the framework for different input data (e.g., 3-D image data and graph data), which is not limited to vectorized inputs, while TCCA cannot take other forms of data.

Comparing to other deep learning based CCAs [8], [19], [44], DTCCA possesses the following advantages:

- DTCCA naturally generalized DCCA for more than two views. DCCA is a special case of DTCCA when the number of input views is 2. However, most of deep learning based CCAs cannot be directly applied for more than two views.
- DTCCA models the high-order correlations among multiple views, but other CCAs can only capture pairwise correlation. Hence, DTCCA can leverage more information/relationships of multiple views.

4 Experiments

We conduct experiments on four data sets to demonstrate that DTCCA learns nonlinear transformations that not only

TABLE 2
Accuracies of 7 compared methods on Caltech101 with 6 views for all 10 folds.

	GCCA	LSCCA	TCCA$_p$	GCCA$_p$	LSCCA$_p$	DGCCA	DTCC
Fold 1	33.31	85.86	93.08	89.10	92.33	93.16	**95.26**
Fold 2	27.67	88.12	94.21	90.98	94.66	93.68	**95.49**
Fold 3	36.32	86.32	90.90	90.38	93.83	92.41	**94.66**
Fold 4	35.86	87.52	93.31	91.58	94.21	94.14	**95.34**
Fold 5	35.41	86.99	94.06	91.43	95.11	94.21	**95.94**
Fold 6	38.87	89.17	86.92	89.62	93.61	92.26	**94.36**
Fold 7	30.08	83.38	93.23	89.92	93.08	93.46	**94.96**
Fold 8	33.46	87.59	93.23	91.35	93.31	93.68	**95.11**
Fold 9	31.95	86.92	92.78	91.43	93.38	94.14	**94.96**
Fold 10	36.99	84.06	92.63	91.28	93.16	93.68	**95.19**
mean	33.99	86.59	92.44	90.71	93.67	93.48	**95.13**
std	3.40	1.78	2.14	0.89	0.82	0.69	0.44

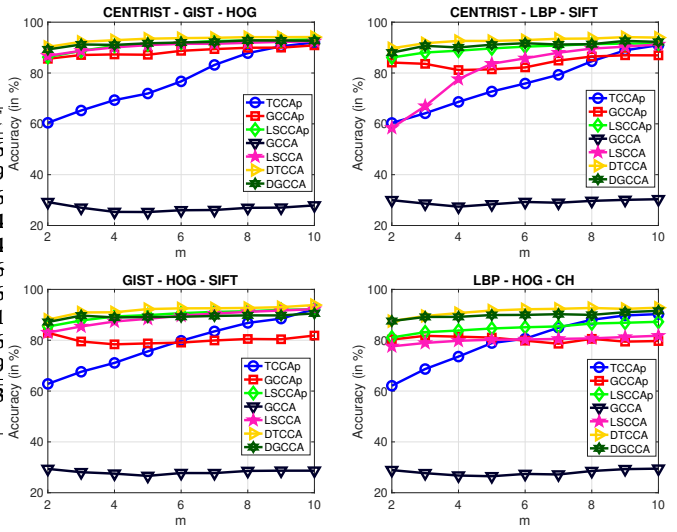

Fig. 2. The accuracy of seven compared methods on Caltech101 over three views by varying the dimension of the reduced space $m \in \{2, 3, \ldots, 10\}$.

outperforms TCCA but also shows competitive or better results comparing with other representative models. Specifically, we compare the following methods in our experiments:

- TCCA$_p$ [25]. The k-order tensor can be high memory intensive for $k \geq 3$ and moderate-dimensional data sets. To make it applicable for all data sets used in the experiments, we first apply PCA on the input data of each view and reduce their dimensions up to 20 to make sure TCCA is feasible (for the view with features less than 20, the PCA step is not applied). As noted in [25] that the kernel version of TCCA does not work for moderate size of data sets with multiple views due to the high memory requirement and computational complexity, so we will not include it in the experiments.
- LSCCA [23] and LSCCA+PCA (LSCCA$_p$). As the representative MCCA method, LSCCA is evaluated on the original input data. In addition, we also preprocess the input data using PCA by preserving 95% energy, which is analogous to the preprocess of TCCA and representation learning of deep learning methods. dMCCA [29] is not included due to its incapability to the data sets used in the experiments.
- GCCA [24], GCCA+PCA (GCCA$_p$), and DGCCA [30]. Even though GCCA and its variants specifically model the common representation, the learning performance is still evaluated based on the learned projections. The preprocessing using PCA is also applied to GCCA similar to LSCCA$_p$. DGCCA is implemented using the same multi-layer perceptron networks as the one used in the proposed DTCCA method for fair comparisons.
- DTCCA. The proposed method is implemented as the nonlinear extension of TCCA using multi-layer perceptron networks for each view as shown in Algorithm 1. DTCCA can handle high-dimensional data and large-scale data, so the preprocessing step using PCA is not applied. We refer to a DTCCA model with an output size of m and v layers (including output) as DTCCA-m-v.

TABLE 3
The accuracy of 7 compared methods over Caltech101 data set by varying the number of views.

	3 views	4 views	5 views	6 views
GCCA	31.02 $\pm$ 1.48	30.54 $\pm$ 0.91	30.81 $\pm$ 0.67	31.09
LSCCA	83.97 $\pm$ 8.97	87.11 $\pm$ 3.62	86.92 $\pm$ 2.95	86.48
TCCA$_p$	90.69 $\pm$ 1.65	90.99 $\pm$ 1.12	91.60 $\pm$ 0.36	91.56
GCCA$_p$	85.65 $\pm$ 3.55	87.66 $\pm$ 3.23	89.43 $\pm$ 1.52	90.35
LSCCA$_p$	90.76 $\pm$ 2.25	92.08 $\pm$ 1.20	92.93 $\pm$ 0.57	93.31
DGCCA	91.80 $\pm$ 1.07	92.70 $\pm$ 0.64	93.28 $\pm$ 0.27	93.26
DTCCA	**93.14 $\pm$ 1.14**	**93.62 $\pm$ 0.78**	**94.23 $\pm$ 0.41**	**94.83**

4.1 Experimental setting

Following the work [25], we first concatenate the projected points of all views obtained by CCA variants in the common space as the final representation for supervised classification problems, and then evaluate the classification performance in terms of accuracy based on linear support vector classifier (SVC) [45]. We split the data into training and test sets. Projections are learned by compared methods using the training data, and the final accuracy is reported based on the test data. To avoid the learning bias, we report the average accuracy and standard deviations over ten randomly drawn training/test splits and parameter C in SVC is tuned in the range $\{0.1, 1, 10\}$ by repeating experiments for 10 times.

The hyper-parameters of compared methods are summarized as follows. The reduced dimension m is one common parameters of all compared methods. We vary $m \in [2, 10]$ and further show the sensitivity analysis in terms of classification accuracy. Moreover, regularization parameter for all CCA methods is set to 10^{-4} in order to prevent the singularity of covariance matrix of each view. For deep learning methods such as DGCCA and DTCCA, the same deep architecture is used for fair comparisons: the number of output neurons is set to m, the default widths of the hidden layers are set to 500 for varied number of layers, and the Adam optimizer is used with the learning rate set to be 10^{-3}, and others are set to be default values in Pytorch. Both Sigmoid and Tanh activation functions are evaluated.

TABLE 4
The accuracy of seven compared methods on Caltech101 data by varying the number of training data with 6 views.

	10%	20%	30%	40%	50%	60%	70%
GCCA	31.09	24.45	28.62	26.26	26.31	33.18	40.92
LSCCA	86.48	79.43	76.17	77.34	79.05	78.04	79.93
TCCA$_p$	91.56	93.71	95.78	96.41	97.51	97.58	97.06
GCCA$_p$	90.35	93.33	94.50	95.40	95.91	96.10	96.14
LSCCA$_p$	93.31	95.59	96.26	96.74	97.21	97.35	97.22
DGCCA	93.26	95.33	96.49	96.88	97.09	97.80	97.40
DTCCA	**94.83**	**96.06**	**96.65**	**97.02**	**97.36**	**97.67**	**97.60**

4.2 Image classification

Image data Caltech101 is used for this analysis [46], which is publicly available[1]. Caltech101 consists of 8677 images belonging to 101 categories. We apply 6 different descriptors to extract features for each image, including: 254-d CENTRIST (CENT) [47], 512-d GIST [48], 1180-d LBP [49], 1008-d histogram of oriented gradient (HOG), 64-d color histogram (CH), and 1000-d SIFT [50]. For classification evaluation, we choose 7 categories with 1474 images in total by following [51]: Faces, Motorbikes, dollar_bill, garfield, snoopy, stop_sign, windsor_chair.

We first randomly selected 10% images from Caltech101 as the training set and the rest as the testing set by evaluating 7 compared methods over all 6 views. Regularization parameter for all CCA methods is set to 10^{-4} in order to prevent the singularity of covariance matrix of each view. The selected widths of the hidden layers for the DTCCA-m-3 and DGCCA-m-3 models are 500 and 500 with $m \in [2, 10]$. The Sigmoid activation function is used, and the dropout with ratio equal to 0.1 is placed in-between each linear layer and its corresponding nonlinear layer except the output layer. The best results of 7 methods for all 10 folders over different $m \in [2, 10]$ are reported in Table 2. We have the following observations: 1) GCCA shows the worst results compared with others; 2) PCA as the preprocessing is helpful to improve CCA classification; 3) nonlinear representation learning using deep networks including DGCCA and DTCCA can significantly improve their base methods GCCA and TCCA, respectively; 4) the proposed DTCCA demonstrates the best results among all compared methods in terms of mean accuracy and the smallest standard deviation, and shows significant better results than TCCA$_p$. These observations imply that the proposed nonlinear extension of TCCA works reasonably well.

We further explore the sensitivity of DTCCA in terms of different dimension of the common space, the number of views, and various training ratios. Since TCCA can naturally incorporate the high-order canonical correlation, we expect that DTCCA or TCCA can perform consistently well in regardless of the number of views. Following the above setting, we varied the number of views from 3 to 6 with 10% training data. We first investigate the impact of dimensions by varying $m \in [2, 10]$. The results are shown in Fig. 2 for some combinations of views. All methods except GCCA demonstrates better results when m increases, while DTCCA demonstrates consistently the best over all these dimensions. We observed the same trends for other combinations. Due to the space limitation, we will not report the results for every combination. To investigate the impact of the number of views, for the specific view, we report the averaged accuracy over all combinations of views, where the accuracy of each combination is again obtained based on the 10 folders. Results are shown in Table 3 for the number of views varying from 3 to 6. As the number of views increases, all methods demonstrate improved accuracy, so learning with multiple views becomes important. Also, our proposed DTCCA not only significantly outperforms TCCA and other linear models, but also better than DGCCA with same deep network architecture. Moreover, we evaluate all methods by varying the number of training data with ratio from 10% to 70%. Results in Table 4 demonstrates that DTCCA performs significantly better than others for small amount of training data, and the accuracies obtained by DGCCA, DGCCA, GCCA$_p$ and TCCA$_p$ converge to a similar value when enough training data becomes available. To show the performance of all combined views, the mean and standard deviation of accuracies of compared methods over 10 folders for each combination of views are shown in Table 5.

To determine the impact of the number of layers in the deep networks based models such as DGCCA and DTCCA, we conduct an experiment in which we increase the number of layers from two to seven. The width of each hidden layer is set to be 500. 10% training data split is used for this experiment with $m = 10$ for one of 10 folds. Table 6 gives the accuracy on the first fold by varying the number of layers from 2 to 7 with nonlinear activation function either Sigmoid or Tanh. We have the following observations: 1) Sigmoid function can obtain good results by using the network of 4 layers, but it becomes worse when the number of layers increases, which is because of the drawback of Sigmoid with zero gradient for deep depth of network; 2) the activation function Tanh does not have this issue so the accuracy continues increasing until 7 layers and the performance reaches saturation.

4.3 Handwritten numeral recognition

Multiple features (Mfeat) data set consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps[2] [52]. 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. These digits are represented in terms of the following six feature sets: 216-d profile correlations (fac), 76-d Fourier coefficients of the character shapes (fou), 64-d Karhunen-Love coefficients (kar), 6-d morphological features (mor), 240-d pixel averages in 2×3 windows (pix), and 47-d Zernike moments (zer). As a result, there are 6 views in total.

The same experimental setting as in Section 4.2 is used for Mfeat data. Results presented in Table 7 shows the classification performance of 7 compared methods on all combinations of views with 10% as training data and the rest as testing data. The impacts of the compared methods in terms of varied combinations of views, the reduced dimensions and the varied training ratio are shown in Table 8,

1. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

2. https://archive.ics.uci.edu/ml/data sets/Multiple+Features

TABLE 5
Mean accuracy and standard deviation of 7 compared methods on 42 data sets generated from Caltech101 by choosing all combinations of more than two views over 10 folders.

View combinations	GCCA	LSCCA	TCCA$_p$	GCCA$_p$	LSCCA$_p$	DGCCA	DTCCA
CENT - GIST - LBP	32.68 ± 6.16	91.91 ± 1.01	92.55 ± 1.04	91.25 ± 1.20	93.14 ± 1.22	92.69 ± 0.78	**93.89 ± 0.72**
CENT - GIST - HOG	29.22 ± 6.39	92.45 ± 0.84	92.01 ± 1.81	90.89 ± 1.72	93.21 ± 1.04	92.83 ± 0.97	**94.15 ± 0.76**
CENT - GIST - CH	34.10 ± 4.34	83.64 ± 1.86	88.53 ± 2.54	90.00 ± 1.08	91.53 ± 1.00	91.73 ± 0.76	**92.89 ± 1.01**
CENT - GIST - SIFT	33.11 ± 3.87	92.56 ± 0.87	89.27 ± 1.32	88.38 ± 1.90	93.11 ± 0.99	92.69 ± 0.96	**94.33 ± 0.89**
CENT - LBP - HOG	33.26 ± 5.85	91.35 ± 1.10	92.90 ± 1.09	89.33 ± 1.29	92.40 ± 0.86	93.14 ± 0.62	**94.29 ± 0.54**
CENT - LBP - CH	31.80 ± 2.73	84.12 ± 1.47	88.75 ± 3.81	89.58 ± 1.07	91.26 ± 0.83	91.12 ± 0.65	**91.76 ± 0.89**
CENT - LBP - SIFT	30.35 ± 1.95	91.23 ± 1.44	90.95 ± 3.88	86.98 ± 1.00	91.62 ± 1.16	92.64 ± 1.12	**94.12 ± 1.10**
CENT - HOG - CH	32.02 ± 4.66	84.53 ± 1.32	90.12 ± 1.38	88.06 ± 1.57	90.94 ± 1.11	91.96 ± 0.93	**92.98 ± 0.77**
CENT - HOG - SIFT	29.44 ± 4.49	92.03 ± 1.19	91.22 ± 2.61	85.44 ± 1.50	92.04 ± 1.88	92.45 ± 1.04	**94.38 ± 0.80**
CENT - CH - SIFT	30.57 ± 4.22	84.64 ± 1.07	89.11 ± 3.31	83.83 ± 3.10	89.60 ± 0.95	91.14 ± 0.68	**92.17 ± 0.94**
GIST - LBP - HOG	31.53 ± 7.65	77.87 ± 3.48	94.11 ± 0.65	85.95 ± 3.23	91.95 ± 1.45	92.90 ± 1.35	**94.37 ± 0.66**
GIST - LBP - CH	30.57 ± 6.26	82.32 ± 2.74	89.03 ± 2.72	85.93 ± 2.59	90.56 ± 0.72	91.83 ± 0.87	**92.40 ± 0.71**
GIST - LBP - SIFT	30.65 ± 2.90	72.08 ± 8.97	91.92 ± 3.27	84.93 ± 1.81	92.74 ± 1.19	92.58 ± 0.76	**93.98 ± 1.24**
GIST - HOG - CH	29.24 ± 5.00	82.61 ± 2.70	88.79 ± 2.04	81.77 ± 2.28	87.60 ± 1.07	90.39 ± 1.27	**92.17 ± 0.93**
GIST - HOG - SIFT	29.38 ± 5.88	92.25 ± 0.92	92.20 ± 2.83	82.94 ± 1.74	92.19 ± 1.62	90.56 ± 1.64	**93.85 ± 0.45**
GIST - CH - SIFT	29.42 ± 3.09	83.41 ± 2.25	89.19 ± 2.75	81.73 ± 3.45	87.98 ± 1.53	90.32 ± 1.07	**91.54 ± 1.16**
LBP - HOG - CH	29.47 ± 2.93	81.76 ± 2.04	90.29 ± 1.61	81.73 ± 1.77	87.25 ± 2.26	91.57 ± 0.92	**92.80 ± 0.54**
LBP - HOG - SIFT	31.51 ± 2.70	54.26 ± 0.24	92.50 ± 2.97	83.56 ± 1.79	92.39 ± 1.14	92.63 ± 1.25	**94.15 ± 0.92**
LBP - CH - SIFT	31.56 ± 3.69	82.19 ± 1.98	89.89 ± 3.08	80.53 ± 2.63	87.52 ± 1.43	91.74 ± 1.02	**91.75 ± 1.75**
HOG - CH - SIFT	30.49 ± 5.21	82.29 ± 1.98	90.50 ± 2.68	80.15 ± 3.24	86.18 ± 2.14	89.18 ± 1.28	**90.91 ± 0.86**
CENT - GIST - LBP - HOG	30.02 ± 5.39	92.49 ± 0.99	92.72 ± 0.91	91.17 ± 0.98	93.41 ± 1.11	93.42 ± 0.60	**94.55 ± 0.87**
CENT - GIST - LBP - CH	30.20 ± 4.39	85.08 ± 1.88	90.29 ± 1.83	91.20 ± 1.04	92.65 ± 1.02	92.38 ± 0.89	**93.09 ± 0.57**
CENT - GIST - LBP - SIFT	30.98 ± 6.21	92.55 ± 1.41	90.15 ± 3.47	89.08 ± 2.37	93.45 ± 0.86	93.38 ± 0.77	**94.35 ± 0.85**
CENT - GIST - HOG - CH	30.79 ± 5.52	84.75 ± 1.73	90.02 ± 2.31	90.71 ± 1.07	92.50 ± 1.04	92.48 ± 0.80	**93.46 ± 0.82**
CENT - GIST - HOG - SIFT	29.26 ± 3.50	92.89 ± 1.01	88.91 ± 4.84	88.84 ± 1.95	93.33 ± 1.32	93.04 ± 1.23	**94.95 ± 1.03**
CENT - GIST - CH - SIFT	32.10 ± 3.86	85.14 ± 1.61	89.63 ± 2.86	89.07 ± 1.31	92.42 ± 0.97	92.48 ± 0.82	**93.70 ± 0.56**
CENT - LBP - HOG - CH	30.03 ± 3.06	85.19 ± 1.49	91.74 ± 1.37	89.69 ± 1.55	92.16 ± 1.19	92.71 ± 0.69	**93.26 ± 0.68**
CENT - LBP - HOG - SIFT	30.02 ± 6.87	92.09 ± 1.08	91.93 ± 2.37	88.12 ± 2.14	92.53 ± 1.46	93.37 ± 0.76	**94.56 ± 0.36**
CENT - LBP - CH - SIFT	32.02 ± 3.84	85.32 ± 1.66	90.48 ± 3.89	88.79 ± 0.89	91.85 ± 0.82	92.10 ± 0.81	**92.95 ± 1.12**
CENT - HOG - CH - SIFT	29.81 ± 2.39	85.51 ± 1.34	90.75 ± 2.61	88.71 ± 2.32	91.44 ± 1.13	92.64 ± 0.87	**93.82 ± 1.17**
GIST - LBP - HOG - CH	29.84 ± 2.69	83.74 ± 1.71	91.03 ± 1.76	85.61 ± 3.75	91.55 ± 0.95	92.65 ± 1.18	**92.98 ± 0.95**
GIST - LBP - HOG - SIFT	29.50 ± 4.26	89.46 ± 1.73	92.81 ± 2.39	86.31 ± 2.29	93.15 ± 0.82	93.38 ± 0.77	**94.44 ± 0.94**
GIST - LBP - CH - SIFT	31.14 ± 4.37	83.91 ± 2.30	91.32 ± 3.77	85.55 ± 3.09	91.48 ± 1.29	**92.93 ± 0.57**	92.91 ± 1.20
GIST - HOG - CH - SIFT	29.62 ± 2.85	84.38 ± 1.93	91.98 ± 2.04	81.80 ± 2.82	89.95 ± 1.28	90.94 ± 0.87	**92.43 ± 1.43**
LBP - HOG - CH - SIFT	31.93 ± 3.46	84.10 ± 2.20	91.09 ± 3.07	80.24 ± 6.90	89.35 ± 2.25	92.65 ± 0.99	**92.84 ± 0.92**
CENT - GIST - LBP - HOG - CH	30.19 ± 3.96	86.00 ± 1.91	91.35 ± 1.34	91.30 ± 1.27	93.22 ± 1.18	93.32 ± 1.18	**93.95 ± 0.89**
CENT - GIST - LBP - HOG - SIFT	30.71 ± 6.44	92.88 ± 1.21	91.59 ± 3.31	89.52 ± 1.74	93.59 ± 0.99	93.79 ± 0.84	**95.03 ± 0.69**
CENT - GIST - LBP - CH - SIFT	31.76 ± 4.24	86.06 ± 1.73	91.46 ± 2.21	90.26 ± 1.25	93.06 ± 1.01	93.01 ± 0.76	**94.05 ± 1.11**
CENT - GIST - HOG - CH - SIFT	31.53 ± 3.95	85.78 ± 1.64	91.34 ± 3.09	89.75 ± 1.21	93.17 ± 1.00	93.21 ± 0.79	**94.32 ± 0.92**
CENT - LBP - HOG - CH - SIFT	30.32 ± 3.53	85.89 ± 1.57	91.56 ± 1.96	89.00 ± 1.26	92.52 ± 1.20	93.14 ± 1.00	**94.07 ± 0.99**
GIST - LBP - HOG - CH - SIFT	30.38 ± 3.80	84.90 ± 2.00	92.30 ± 2.18	86.77 ± 2.81	92.01 ± 1.07	93.22 ± 0.78	**94.00 ± 0.60**
CENT - GIST - LBP - HOG - CH - SIFT	31.09 ± 5.04	86.48 ± 1.84	91.56 ± 2.53	90.35 ± 0.98	93.31 ± 1.04	93.26 ± 0.66	**94.83 ± 0.28**

TABLE 6
Accuracy of two methods with layers ranging from two to seven on Caltech101 with 6 views. '-' is for the failure of the training process.

Layers	2	3	4	5	6	7
DGCCA (sigmoid)	93.46	92.63	93.31	90.00	56.39	-
DTCCA (sigmoid)	95.64	95.04	95.41	92.11	81.58	-
DGCCA (tanh)	92.33	93.01	92.63	92.11	91.88	92.48
DTCCA (tanh)	94.29	94.44	94.51	94.74	95.11	95.11

Fig. 3, Table 9 respectively. These observations are consistent with what we have observed for Caltech101.

4.4 Scene classification

The 15 class scene data set was gradually built. The initial 8 classes were collected by Oliva and Torralba [48], and then 5 categories were added by Fei-Fei and Perona [53]; finally, 2 additional categories were introduced by Lazebnik et al. [50]. The 15 scene categories are office, kitchen, living room, bedroom, store, industrial, tall building, inside cite, street, highway, coast, open country, mountain, forest, and

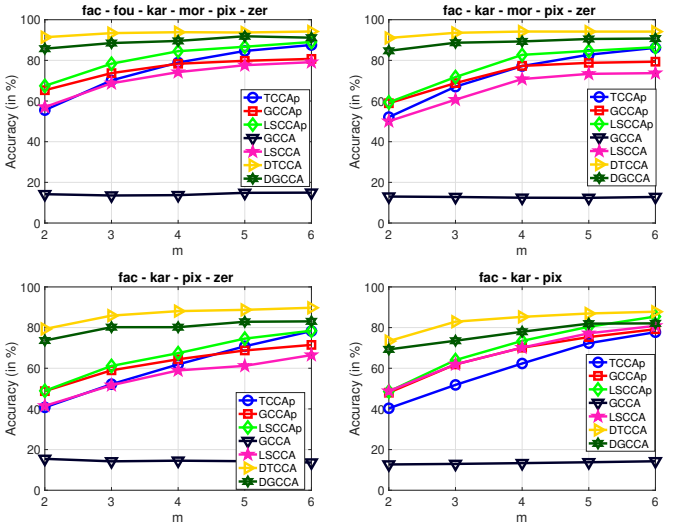

Fig. 3. The accuracy of seven compared methods on Mfeat data by varying the dimension of the reduced space $m \in \{2, 3, \ldots, 10\}$.

TABLE 7
Mean accuracy and standard deviation of 7 compared methods on 42 data sets generated from Mfeat data by choosing all combinations of more than two views over 10 folders.

View combinations	GCCA	LSCCA	TCCA$_p$	GCCA$_p$	LSCCA$_p$	DGCCA	DTCCA
fac - fou - kar	26.18 ± 4.01	62.80 ± 3.02	83.34 ± 2.83	79.60 ± 2.33	84.73 ± 2.14	**85.82 ± 1.83**	85.61 ± 1.10
fac - fou - mor	39.72 ± 3.63	62.05 ± 4.01	86.96 ± 2.81	83.23 ± 1.68	85.39 ± 1.43	92.51 ± 1.17	**92.62 ± 0.99**
fac - fou - pix	13.58 ± 1.63	69.15 ± 3.44	83.45 ± 1.58	81.84 ± 2.01	85.73 ± 1.98	86.75 ± 1.47	**88.91 ± 1.08**
fac - fou - zer	30.82 ± 5.15	64.69 ± 2.58	79.61 ± 3.21	78.92 ± 2.17	82.89 ± 1.66	86.59 ± 1.70	**87.67 ± 0.94**
fac - kar - mor	37.52 ± 6.46	69.84 ± 5.68	87.05 ± 1.48	84.90 ± 2.62	88.30 ± 2.62	91.39 ± 0.67	**92.69 ± 1.02**
fac - kar - pix	14.26 ± 1.09	80.84 ± 2.34	77.73 ± 1.86	79.18 ± 2.35	85.43 ± 1.34	82.08 ± 1.70	**87.82 ± 1.70**
fac - kar - zer	31.72 ± 2.74	65.68 ± 2.48	79.03 ± 2.57	73.70 ± 2.77	77.35 ± 3.03	83.12 ± 1.56	**86.96 ± 0.79**
fac - mor - pix	12.52 ± 2.05	55.00 ± 6.94	88.89 ± 2.11	83.78 ± 2.68	86.84 ± 2.64	92.08 ± 0.61	**94.14 ± 0.53**
fac - mor - zer	39.04 ± 2.79	67.52 ± 2.59	84.07 ± 2.25	80.61 ± 1.90	82.59 ± 1.93	91.32 ± 0.67	**93.21 ± 0.89**
fac - pix - zer	13.48 ± 1.82	61.83 ± 2.26	80.76 ± 3.92	71.18 ± 2.68	75.17 ± 2.37	85.64 ± 1.42	**89.47 ± 1.05**
fou - kar - mor	69.55 ± 2.25	73.95 ± 2.22	85.77 ± 1.34	80.02 ± 1.62	83.03 ± 1.70	**87.61 ± 1.62**	87.51 ± 1.24
fou - kar - pix	13.61 ± 1.77	60.19 ± 3.85	82.46 ± 2.85	73.81 ± 3.39	80.16 ± 3.11	83.52 ± 1.23	**84.18 ± 1.63**
fou - kar - zer	63.83 ± 1.70	74.33 ± 1.70	77.96 ± 3.50	75.72 ± 3.37	80.24 ± 2.90	**81.23 ± 1.15**	80.28 ± 1.83
fou - mor - pix	11.89 ± 1.52	62.20 ± 3.71	86.96 ± 2.36	80.19 ± 2.16	82.81 ± 1.56	90.98 ± 2.05	**92.43 ± 0.72**
fou - mor - zer	71.16 ± 3.02	74.58 ± 2.33	79.64 ± 1.25	78.03 ± 1.74	80.29 ± 1.33	**81.18 ± 0.77**	79.75 ± 0.70
fou - pix - zer	12.30 ± 1.53	61.30 ± 2.34	78.88 ± 2.56	76.77 ± 1.68	80.59 ± 1.60	85.26 ± 1.59	**85.58 ± 0.98**
kar - mor - pix	13.37 ± 1.43	69.11 ± 5.24	85.67 ± 2.16	79.41 ± 2.89	82.88 ± 2.79	89.68 ± 1.34	**92.80 ± 0.72**
kar - mor - zer	72.71 ± 2.77	76.68 ± 2.67	83.95 ± 2.12	79.94 ± 2.01	82.87 ± 1.83	86.91 ± 1.85	**89.21 ± 1.10**
kar - pix - zer	15.10 ± 2.29	65.11 ± 2.81	75.29 ± 2.12	65.22 ± 1.88	71.21 ± 2.60	83.15 ± 1.58	**86.30 ± 1.76**
mor - pix - zer	13.87 ± 2.51	63.39 ± 2.06	85.01 ± 1.47	79.08 ± 1.55	81.79 ± 1.75	89.73 ± 1.00	**92.81 ± 0.65**
fac - fou - kar - mor	41.02 ± 5.67	73.00 ± 1.91	86.87 ± 2.25	83.64 ± 2.44	88.29 ± 1.78	91.88 ± 0.64	**92.53 ± 0.59**
fac - fou - kar - pix	13.06 ± 1.69	64.89 ± 2.95	79.75 ± 2.39	77.73 ± 2.90	86.21 ± 2.07	84.59 ± 1.79	**88.62 ± 0.77**
fac - fou - kar - zer	32.44 ± 3.98	74.89 ± 1.14	77.94 ± 3.10	79.57 ± 1.88	85.60 ± 1.90	85.85 ± 1.13	**87.11 ± 1.20**
fac - fou - mor - pix	11.77 ± 2.02	60.92 ± 3.33	87.84 ± 2.87	83.77 ± 2.40	88.58 ± 1.64	92.41 ± 1.33	**93.87 ± 0.44**
fac - fou - mor - zer	39.43 ± 5.12	72.76 ± 0.91	83.62 ± 4.00	81.29 ± 1.46	85.07 ± 1.43	91.60 ± 0.99	**93.28 ± 0.80**
fac - fou - pix - zer	12.11 ± 1.53	64.72 ± 2.35	79.04 ± 3.27	79.40 ± 1.86	85.56 ± 1.66	86.59 ± 1.03	**90.60 ± 1.00**
fac - kar - mor - pix	14.02 ± 0.75	69.27 ± 5.43	86.46 ± 1.50	84.68 ± 2.95	90.72 ± 1.69	91.08 ± 1.15	**94.50 ± 0.66**
fac - kar - mor - zer	39.37 ± 3.92	74.68 ± 2.72	86.16 ± 2.33	81.16 ± 1.58	85.82 ± 1.91	90.22 ± 1.24	**93.62 ± 0.86**
fac - kar - pix - zer	15.45 ± 2.49	66.48 ± 2.36	78.28 ± 2.46	71.48 ± 2.87	78.47 ± 2.91	83.07 ± 1.80	**89.75 ± 0.53**
fac - mor - pix - zer	12.87 ± 2.73	64.54 ± 2.20	85.23 ± 2.72	80.33 ± 1.84	84.61 ± 1.81	91.82 ± 1.20	**94.59 ± 0.59**
fou - kar - mor - pix	14.31 ± 0.73	71.92 ± 1.84	85.73 ± 1.84	79.48 ± 2.17	85.25 ± 1.64	90.31 ± 1.05	**92.19 ± 1.05**
fou - kar - mor - zer	71.01 ± 2.79	79.72 ± 1.60	84.27 ± 2.80	80.73 ± 2.43	85.03 ± 2.16	87.11 ± 1.54	**88.62 ± 1.60**
fou - kar - pix - zer	15.40 ± 2.21	74.01 ± 1.72	77.18 ± 3.40	76.07 ± 1.89	82.77 ± 1.91	84.28 ± 1.59	**86.73 ± 1.72**
fou - mor - pix - zer	13.12 ± 1.75	70.47 ± 2.37	85.17 ± 2.45	79.86 ± 1.81	84.96 ± 1.58	90.36 ± 1.68	**92.48 ± 0.56**
kar - mor - pix - zer	13.86 ± 2.49	73.28 ± 2.26	85.22 ± 2.40	77.94 ± 2.27	83.24 ± 1.58	89.97 ± 1.69	**93.39 ± 0.86**
fac - fou - kar - mor - pix	14.38 ± 1.34	72.72 ± 1.90	88.02 ± 2.76	81.93 ± 3.16	89.27 ± 1.99	91.76 ± 0.62	**93.87 ± 0.63**
fac - fou - kar - mor - zer	39.92 ± 5.13	79.48 ± 1.11	85.76 ± 1.85	81.93 ± 2.17	88.03 ± 1.87	91.72 ± 0.90	**93.23 ± 0.68**
fac - fou - kar - pix - zer	15.30 ± 2.36	75.42 ± 1.18	79.94 ± 2.41	78.84 ± 1.30	86.75 ± 1.82	86.34 ± 1.75	**89.72 ± 1.19**
fac - fou - mor - pix - zer	13.14 ± 2.35	71.19 ± 1.01	86.69 ± 3.08	81.73 ± 2.70	87.39 ± 1.81	91.99 ± 1.00	**94.09 ± 0.98**
fac - kar - mor - pix - zer	13.02 ± 1.98	73.73 ± 2.06	86.17 ± 1.89	79.34 ± 2.03	86.46 ± 1.60	90.73 ± 0.60	**94.18 ± 0.76**
fou - kar - mor - pix - zer	15.36 ± 2.22	78.29 ± 1.45	85.11 ± 2.67	80.42 ± 2.15	86.49 ± 2.07	90.14 ± 1.24	**92.92 ± 0.88**
fac - fou - kar - mor - pix - zer	14.94 ± 2.33	79.12 ± 1.39	87.56 ± 2.56	80.73 ± 2.59	89.03 ± 1.79	91.86 ± 1.03	**94.21 ± 0.61**

TABLE 8
The accuracy of 7 compared methods over Mfeat data set by varying the number of views.

	3 views	4 views	5 views	6 views
GCCA	30.81 ± 22.04	23.95 ± 17.20	18.52 ± 10.53	14.94
LSCCA	67.01 ± 6.49	70.37 ± 5.10	75.14 ± 3.23	79.12
TCCA$_p$	82.62 ± 3.80	83.25 ± 3.70	85.28 ± 2.80	87.56
GCCA$_p$	78.26 ± 4.60	79.81 ± 3.28	80.70 ± 1.37	80.73
LSCCA$_p$	82.01 ± 4.03	85.35 ± 2.78	87.40 ± 1.10	89.03
DGCCA	86.83 ± 3.73	88.74 ± 3.16	90.45 ± 2.13	91.86
DTCCA	**88.50 ± 4.18**	**91.46 ± 2.68**	**93.00 ± 1.68**	**94.21**

TABLE 9
The accuracy of seven compared methods on Mfeat data by varying the number of training data with 6 views.

	10%	20%	30%	40%	50%	60%	70%
GCCA	14.94	75.68	84.96	87.69	88.74	89.25	89.31
LSCCA	79.12	86.88	88.59	89.69	90.12	89.92	90.21
TCCA$_p$	87.56	91.13	92.97	94.57	94.56	95.15	95.21
GCCA$_p$	80.73	86.74	89.97	90.58	91.61	92.53	92.39
LSCCA$_p$	89.03	90.00	91.46	91.70	92.01	92.58	92.79
DGCCA	91.86	94.09	95.69	96.47	96.89	97.23	**97.72**
DTCCA	**94.21**	**95.69**	**96.54**	**96.78**	**97.14**	**97.56**	97.59

suburb. Images in the data set are about 250×300 resolution, with 210 to 410 images per class. This data set contains a wide range of outdoor and indoor scene environments. 4310 images are used in this experiment. Five descriptors are used to generate the features of views including 254-d CENTRIST [47], 512-d GIST [48], 531-d LBP [49], 360-d histogram of oriented gradient (HOG), and 1000-d SIFT [50]. The same experiments are also conducted for Scene15 data. The classification accuracy in terms of the varied views, the dimension of common space, and view combinations are shown in Table 11, Fig. 4 and Table 10, respectively. The similar observations as above two data sets can also be obtained on this data.

4.5 Wikipedia classification

To further validate the effectiveness of the proposed methods on multi-modal data, we conduct the experiments on data Wikipedia [54], which consists of 2866 image and text pairs. Each featured article is categorized by Wikipedia into one of 10 categories. The representation of text is derived from a latent Dirichlet allocation (LDA) model, where each

TABLE 10
Mean accuracy and standard deviation of 7 compared methods on 16 data sets generated from Scene15 data by choosing all combinations of more than two views over 10 folders.

View combinations	GCCA	LSCCA	$TCCA_p$	$GCCA_p$	$LSCCA_p$	DGCCA	DTCCA
CENTRIST - GIST - LBP	8.03 ± 0.73	50.83 ± 1.86	60.13 ± 1.90	58.39 ± 1.86	62.34 ± 1.45	**64.79 ± 0.95**	64.49 ± 1.08
CENTRIST - GIST - HOG	8.81 ± 0.69	35.26 ± 1.92	56.76 ± 2.75	56.05 ± 1.43	59.18 ± 1.28	**64.57 ± 1.94**	64.11 ± 0.42
CENTRIST - GIST - SIFT	9.64 ± 0.58	50.48 ± 2.19	68.70 ± 1.87	56.34 ± 2.34	61.23 ± 1.96	69.64 ± 1.35	**70.97 ± 1.46**
CENTRIST - LBP - HOG	8.15 ± 0.64	42.62 ± 1.78	56.82 ± 1.42	54.06 ± 1.49	57.20 ± 1.43	61.35 ± 1.14	**62.55 ± 1.37**
CENTRIST - LBP - SIFT	8.46 ± 1.09	58.72 ± 2.08	64.87 ± 2.96	50.35 ± 1.93	56.50 ± 1.78	65.22 ± 1.46	**69.87 ± 1.32**
CENTRIST - HOG - SIFT	8.70 ± 0.87	43.50 ± 2.03	65.21 ± 2.79	51.32 ± 2.28	55.64 ± 1.58	66.66 ± 2.09	**69.64 ± 1.56**
GIST - LBP - HOG	8.31 ± 0.65	43.23 ± 1.22	54.60 ± 1.78	46.08 ± 1.54	48.76 ± 1.66	**59.92 ± 0.74**	59.58 ± 0.99
GIST - LBP - SIFT	8.72 ± 0.57	55.43 ± 1.64	67.37 ± 3.24	50.37 ± 1.83	55.56 ± 1.91	63.59 ± 2.14	**66.45 ± 1.51**
GIST - HOG - SIFT	9.79 ± 0.62	45.03 ± 1.07	64.87 ± 3.67	44.97 ± 2.19	49.21 ± 2.01	62.34 ± 1.10	**65.19 ± 1.08**
LBP - HOG - SIFT	8.28 ± 0.98	50.91 ± 0.83	64.25 ± 2.31	44.76 ± 1.21	49.58 ± 1.73	61.50 ± 1.56	**65.18 ± 1.62**
CENTRIST - GIST - LBP - HOG	8.56 ± 0.96	37.90 ± 1.71	55.45 ± 1.33	55.28 ± 1.90	61.37 ± 1.09	**63.60 ± 1.53**	62.92 ± 0.79
CENTRIST - GIST - LBP - SIFT	9.20 ± 0.70	53.61 ± 2.00	66.24 ± 2.32	54.25 ± 2.85	63.14 ± 2.01	65.76 ± 1.67	**68.01 ± 1.98**
CENTRIST - GIST - HOG - SIFT	9.30 ± 0.81	38.93 ± 2.75	64.08 ± 2.38	53.60 ± 2.13	60.38 ± 1.71	66.73 ± 1.45	**66.96 ± 1.18**
CENTRIST - LBP - HOG - SIFT	8.79 ± 0.99	46.26 ± 1.61	64.94 ± 1.92	50.33 ± 1.85	59.04 ± 1.57	63.46 ± 1.79	**66.24 ± 1.85**
GIST - LBP - HOG - SIFT	9.24 ± 0.76	46.91 ± 1.07	62.65 ± 2.57	45.24 ± 1.44	51.47 ± 1.23	62.88 ± 1.82	**64.73 ± 1.49**
CENTRIST - GIST - LBP - HOG - SIFT	9.44 ± 1.00	41.07 ± 2.08	64.45 ± 1.75	51.27 ± 1.72	62.72 ± 1.41	64.71 ± 1.08	**67.33 ± 1.43**

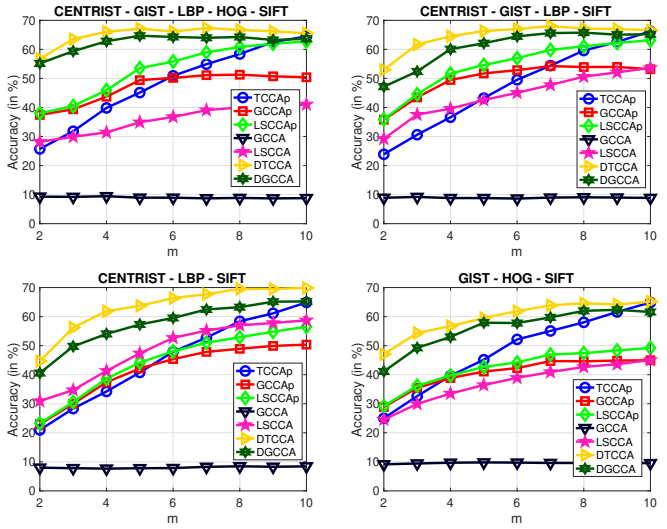

Fig. 4. The accuracy of seven compared methods on Scene15 data by varying the dimension of the reduced space $m \in \{2, 3, \ldots, 10\}$.

TABLE 11
The accuracy of 7 compared methods over Scene15 data set by varying the number of views.

	3 views	4 views	5 views
GCCA	8.69 ± 0.60	9.02 ± 0.32	9.44
LSCCA	47.60 ± 6.95	44.72 ± 6.44	41.07
$TCCA_p$	62.36 ± 4.91	62.67 ± 4.24	64.45
$GCCA_p$	51.27 ± 4.92	51.74 ± 4.08	51.27
$LSCCA_p$	55.52 ± 4.91	59.08 ± 4.51	62.72
DGCCA	63.96 ± 2.87	64.48 ± 1.66	64.71
DTCCA	**65.80 ± 3.54**	**65.77 ± 1.99**	**67.33**

document is represented by their topic assignment probabilistic distribution with top 10 topics selected. The representation of each image is constructed by four descriptors: 254-d CENTRIST, 512-d GIST, 1180-d LBP and 1000-d SIFT as used for Caltech101 in Section 4.2.

By taking the similar experimental setting in 4.2, we randomly selected 10% pairs of image and text as the training set and the rest as the testing set. Two hidden layers with 500 latent neurons are employed for both DGCCA and DTCCA with dropout ratio 0.1. The reduced dimension m is tuned in the range of $[2, 10]$ as the maximum dimension of text features is 10. Table 12 shows the best results of 7 methods for all 10 folders over different $m \in [2, 10]$. And the sensitivity of DTCCA in terms of classification accuracy by varying the number of views from 3 to 5 is shown in Table 13. The similar observations as shown above can also be obtained on multi-modal Wikipedia data.

5 Conclusion

We propose DTCCA for dealing with multi-view extensions of TCCA by capturing the high-order statistics among all feature views and simultaneously learning the nonlinear transformations of each view in a unified model. Based on the experiments on various multi-view data sets, we have shown that DTCCA can obtain significant improvement comparing with TCCA and better or competitive results comparing with others with respect to the classification performance on test data. In addition, DTCCA does not need to apply preprocessing step to avoid the high computational complexity of TCCA on the high-dimensional input data in the case that the dimensionality of the latent subspace is relatively small. We observed that DTCCA can achieve consistently better results especially when the amount of training data is small. As a result, the combination of nonlinear transformation and maximizing high-order canonical correlations are important to improve the learning performance of multi-view data sets.

References

[1] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.
[2] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.
[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.
[4] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *Proceedings of the ACM international conference on image and video retrieval*, 2009, pp. 1–9.

TABLE 12
Mean accuracy and standard deviation of 7 compared methods on 11 data sets generated from Wikipedia data by choosing all combinations of more than two views including LDA over 10 folders.

View combinations	GCCA	LSCCA	TCCA$_p$	GCCA$_p$	LSCCA$_p$	DGCCA	DTCCA
CENTRIST - GIST - LDA	11.43 ± 1.25	27.82 ± 2.14	64.17 ± 1.04	62.21 ± 1.62	63.60 ± 0.96	64.61 ± 1.15	**66.70 ± 0.76**
CENTRIST - LBP - LDA	13.05 ± 1.81	29.71 ± 1.35	64.47 ± 1.26	61.55 ± 1.60	64.20 ± 1.22	64.90 ± 0.63	**67.03 ± 0.86**
CENTRIST - SIFT - LDA	12.46 ± 1.17	36.39 ± 2.02	71.84 ± 1.58	63.28 ± 2.58	67.77 ± 1.67	70.90 ± 1.29	**74.10 ± 1.12**
GIST - LBP - LDA	11.85 ± 2.13	34.66 ± 1.40	64.27 ± 1.13	60.24 ± 1.95	62.45 ± 1.63	64.57 ± 1.01	**66.56 ± 0.65**
GIST - SIFT - LDA	12.30 ± 2.39	47.77 ± 1.37	72.16 ± 1.38	62.66 ± 1.77	66.53 ± 1.74	70.04 ± 1.03	**73.94 ± 1.23**
LBP - SIFT - LDA	12.02 ± 1.55	58.56 ± 1.43	73.03 ± 2.24	62.59 ± 2.63	67.28 ± 1.87	70.74 ± 1.30	**75.22 ± 2.03**
CENTRIST - GIST - LBP - LDA	10.91 ± 1.48	26.74 ± 1.94	60.80 ± 1.80	57.82 ± 1.61	61.23 ± 0.67	61.81 ± 1.05	**64.19 ± 0.97**
CENTRIST - GIST - SIFT - LDA	11.42 ± 1.85	32.04 ± 2.07	67.91 ± 1.64	59.16 ± 1.81	64.25 ± 1.37	67.00 ± 1.90	**69.56 ± 2.18**
CENTRIST - LBP - SIFT - LDA	12.40 ± 1.46	33.59 ± 1.67	68.51 ± 1.74	58.66 ± 1.79	64.57 ± 1.58	66.59 ± 1.97	**69.74 ± 0.72**
GIST - LBP - SIFT - LDA	11.96 ± 2.13	44.51 ± 1.08	68.53 ± 1.41	57.72 ± 1.54	63.75 ± 1.65	66.71 ± 1.46	**69.60 ± 1.80**
CENTRIST - GIST - LBP - SIFT - LDA	11.61 ± 1.69	30.74 ± 1.57	65.22 ± 1.74	55.51 ± 1.11	61.73 ± 0.75	63.71 ± 2.01	**67.56 ± 1.34**

TABLE 13
The accuracy of 7 compared methods over Wikipedia data set by varying the number of views.

	3 views	4 views	5 views
GCCA	12.18 ± 0.55	11.67 ± 0.65	11.61
LSCCA	39.15 ± 11.80	34.22 ± 7.46	30.74
TCCA+PCA	68.32 ± 4.42	66.44 ± 3.77	65.22
GCCA+PCA	62.09 ± 1.07	58.34 ± 0.69	55.51
LSCCA+PCA	65.31 ± 2.18	63.45 ± 1.52	61.73
DGCCA	67.63 ± 3.23	65.53 ± 2.48	63.71
DTCCA	**70.59 ± 4.22**	**68.27 ± 2.72**	**67.56**

[5] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," 2005.

[6] H. Harold, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[7] D. P. Foster, S. M. Kakade, and T. Zhang, "Multi-view dimensionality reduction via canonical correlation analysis," 2008.

[8] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *International conference on machine learning*, 2013, pp. 1247–1255.

[9] D. Chu, L.-Z. Liao, M. K. Ng, and X. Zhang, "Sparse canonical correlation analysis: New formulation and algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 3050–3065, 2013.

[10] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[11] A. A. Nielsen, "Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data," *IEEE transactions on image processing*, vol. 11, no. 3, pp. 293–305, 2002.

[12] V. Uurtio, J. M. Monteiro, J. Kandola, J. Shawe-Taylor, D. Fernandez-Reyes, and J. Rousu, "A tutorial on canonical correlation methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.

[13] Y. Song, Y. Li, L. Jia, and M. Qiu, "Retraining strategy-based domain adaption network for intelligent fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6163–6171, 2019.

[14] K. Gai, M. Qiu, H. Zhao, and X. Sun, "Resource management in sustainable cyber-physical systems using heterogeneous cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 2, pp. 60–72, 2017.

[15] K. Gai and M. Qiu, "Reinforcement learning-based content-centric services in mobile sensing," *IEEE Network*, vol. 32, no. 4, pp. 34–39, 2018.

[16] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *International Conference on Machine Learning*, 2015, pp. 1083–1092.

[19] W. Wang, X. Yan, H. Lee, and K. Livescu, "Deep variational canonical correlation analysis," *arXiv preprint arXiv:1610.03454*, 2016.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[21] N. E. D. Elmadany, Y. He, and L. Guan, "Multiview learning via deep discriminative canonical correlation analysis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2409–2413.

[22] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[23] J. Vía, I. Santamaría, and J. Pérez, "A learning algorithm for adaptive canonical correlation analysis of several data sets," *Neural Networks*, vol. 20, no. 1, pp. 139–152, 2007.

[24] P. Horst, *Generalized canonical correlations and their application to experimental data*. Journal of clinical psychology, 1961, no. 14.

[25] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen, "Tensor canonical correlation analysis for multi-view dimension reduction," *IEEE transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3111–3124, 2015.

[26] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 10, pp. 1700–1715, 2007.

[27] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5910–5922, 2019.

[28] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *International Journal of Computer Vision*, pp. 1–18, 2020.

[29] K. Somandepalli, N. Kumar, R. Travadi, and S. Narayanan, "Multimodal representation learning using deep multiset canonical correlation," *arXiv preprint arXiv:1904.01775*, 2019.

[30] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," *Proceedings of the 4th Workshop on Representation Learning for NLP*, pp. 1–6, 2019.

[31] S. Sun, X. Xie, and M. Yang, "Multiview uncorrelated discriminant analysis," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 3272–3284, 2015.

[32] G. Cao, A. Iosifidis, K. Chen, and M. Gabbouj, "Generalized multi-view embedding for visual recognition and cross-modal retrieval," *IEEE transactions on cybernetics*, vol. 48, no. 9, pp. 2542–2555, 2017.

[33] M. Xu, Z. Zhu, X. Zhang, Y. Zhao, and X. Li, "Canonical correlation analysis with l2, 1-norm for multiview data representation," *IEEE transactions on cybernetics*, 2019.

[34] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[35] P. M. Kroonenberg and J. De Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, no. 1, pp. 69–97, 1980.

[36] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 23, no. 7-8, pp. 393–405, 2009.

[37] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of machine learning research*, vol. 12, no. 64, pp. 2211–2268, 2011.
[38] R. Arora and K. Livescu, "Kernel cca for multi-view learning of acoustic features using articulatory measurements," in *Symposium on Machine Learning in Speech and Language Processing*, 2012.
[39] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
[40] Y. Guan and D. Chu, "Numerical computation for orthogonal low-rank approximation of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 3, pp. 1047–1065, 2019.
[41] Y. Guan, M. T. Chu, and D. Chu, "Convergence analysis of an svd-based algorithm for the best rank-1 tensor approximation," *Linear Algebra and its Applications*, vol. 555, pp. 53–69, 2018.
[42] A. Paszke, S. Gross, F. Massa, A. Lerer, and etc., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
[43] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "Tensorly: Tensor learning in python," *Journal of Machine Learning Research*, vol. 20, no. 26, pp. 1–6, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-277.html
[44] W. Fan, Y. Ma, H. Xu, X. Liu, J. Wang, Q. Li, and J. Tang, "Deep adversarial canonical correlation analysis," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 352–360.
[45] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.
[46] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer vision and Image understanding*, vol. 106, no. 1, pp. 59–70, 2007.
[47] J. Wu and J. M. Rehg, "Where am i: Place instance and category recognition using spatial pact," in *2008 Ieee Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.
[48] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.
[49] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.
[50] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 2169–2178.
[51] C. Deng, Z. Lv, W. Liu, J. Huang, D. Tao, and X. Gao, "Multi-view matrix decomposition: a new scheme for exploring discriminative information," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
[52] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[53] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 524–531.
[54] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 251–260.