A Novel Multi-task Tensor Correlation Neural Network for Facial Attribute Prediction

MINGXING DUAN and KENLI LI, Hunan University, China
KEQIN LI, State University of New York, USA
QI TIAN, Huawei, China

Multi-task learning plays an important role in face multi-attribute prediction. At present, most researches excavate the shared information between attributes by sharing all convolutional layers. However, it is not appropriate to treat the low-level and high-level features of the face multi-attribute equally, because the high-level features are more biased toward the specific content of the category. In this article, a novel multi-attribute tensor correlation neural network (MTCN) is used to predict face attributes. MTCN shares all attribute features at the low-level layers, and then distinguishes each attribute feature at the high-level layers. To better excavate the correlations among high-level attribute features, each sub-network explores useful information from other networks to enhance its original information. Then a tensor canonical correlation analysis method is used to seek the correlations among the highest-level attributes, which enhances the original information of each attribute. After that, these features are mapped into a highly correlated space through the correlation matrix. Finally, we use sufficient experiments to verify the performance of MTCN on the CelebA and LFWA datasets and our MTCN achieves the best performance compared with the latest multi-attribute recognition algorithms under the same settings.

CCS Concepts: • **General and reference** → **Reference works**; • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: Attribute prediction, correlation, multi-task learning, tensor correlation analysis algorithm

ACM Reference format:
Mingxing Duan, Kenli Li, Keqin Li, and Qi Tian. 2020. A Novel Multi-task Tensor Correlation Neural Network for Facial Attribute Prediction. *ACM Trans. Intell. Syst. Technol.* 12, 1, Article 3 (November 2020), 22 pages.
https://doi.org/10.1145/3418285

This work was supported in part by the National Outstanding Youth Science Program of National Natural Science Foundation of China under Grant No. 61625202, in part by the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China under Grant No. 61661146006, in part by the National Key R&D Program of China under Grant No. 2016YT80201900, in part by the National Youth Science Program of National Natural Science Foundation of China under Grant No. 61902119, and in part by the Project funded by China Postdoctoral Science Foundation under Grants No. 2019M652758 and No. 2019TQ0087. This article is funded by the Hong Kong Scholars Program under Grant No. XJ2020032.

Authors' addresses: M. Duan and K. Li (corresponding author), Hunan University, School of Information Science and Engineering, Lushang Road, Changsha, Hunan, 410000, China; emails: duanmingxing@hnu.edu.cn, lkl@hnu.edu.cn; K. Li, State University of New York, Computer Science, New Paltz, New York, 12561, USA; email: lik@newpaltz.edu; Q. Tian, Huawei, Cloud BU, Shenzhen, China; email: tian.qi1@huawei.com.

2157-6904/2020/11-ART3 $15.00
https://doi.org/10.1145/3418285

1 INTRODUCTION

1.1 Motivation

Face attribute recognition is widely used in our lives, such as (i) target tracking, surveillance, and salfeld child control [15, 35, 36, 38, 39, 55], e.g., prisoner tracking, pedestrian identification; (ii) face retrieval [14, 45, 48, 60]; and (iii) social media [46, 47], e.g., face aging, face different attributes transfer learning.

In spite of the recent progress in face attribute estimation [10, 32, 49, 53], most work still uses a specify algorithm to predict a single attribute, but they ignore the strong correlation between attributes, such as beard and male, lipstick and female, apple knot and male, and so on. When some attributes are unknown, we can perform correlation analysis on other closely related attributes to accurately predict those attributes and a joint attribute estimation algorithm is one of the typical applications [2, 12, 19, 40, 48]. Multi-task learning is generally used in the joint attribute estimation algorithm and Figure 1 shows two commonly used multi-task attribute learning models. In Figure 1(a), the algorithms of this type share all information from the input layer to the fully connected layer [19, 48]. Although these face attributes have more commonalities in the low-level features, they are more inclined to their own characteristics. If they completely share the feature information from the low-level to the high-level, then most attributes will lose their due characteristics in the high-level features. In Figure 1(b), these algorithms usually utilize multiple single networks to extract the corresponding attribute features, and then fuse these information to make the final prediction [2, 9–11, 65]. The model of this type is expensive to train and ignores the correlations among attributes.

To share the low-level features and distinguish their high-level features for all attributes, Ref. [20] designed a multi-task learning model (MCNN) for multi-attribute recognition. The model uses the first two convolutional layers to share the low-level features of each attribute, and then the third convolutional layer starts to distinguish the information of each attribute. After that the authors design an additional network to count all attribute scores, thereby improving the classification accuracies [27, 54, 66, 67]. Although the high-level characteristics of each attribute are different, there is a strong correlation between each other. MCNN does not make full use of these correlations, and in addition, three convolutional layers cannot fully extract certain attribute-specific information. Therefore, we build a multi-task CNN including five convolutional layers to extract the characteristic information of each attribute, and explore the correlations among subnetworks, which improves the original information of a single attribute.

We have already discussed that there are strong correlations between face attributes and making full use of these correlations can improve the final prediction performance of a single attribute. For example, we can predict the gender or age attributes of the face through the dynamic changes of smile attribute [7, 8], and we can also utilize gender and race attribute to enhance age attributes [17]. However, the strengths of the correlations between face attributes are not the same, and using a unified correlation learning algorithm to enhance a single attribute feature may cause the performance of some attributes to decrease. To solve this kind of problem, we use a tensor canonical correlation analysis (TCCA) to excavate the correlation among high-level features. At the same time, a global generalization matrix projects the features learned by TCCA into a highly correlated space, so that both the correlations among attributes and its different degrees of influence between the attributes are taken into account.

1.2 Our Contributions

In this article, we propose a multi-task learning model (MTCN) for the prediction of face attributes. MTCN mainly consists of three parts: low-level feature sharing, high-level feature differentiation

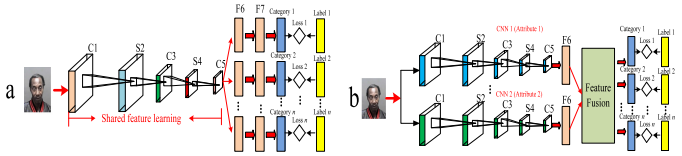

Fig. 1. The methods used for attribute estimation. In Fig. (a), the type of the method shares information in all convolutional layers, while in Fig. (b), the approach fuses the features of multiple CNN models to achieve robust features.

and correlation excavation, and correlations of global features learning via TCCA and generalization matrix. As shown in Figure 3, MTCN first uses two convolutional layers to learn the common features of all attributes, and then learns all attribute-specific information separately from the third convolutional layer. At this time, the networks after C5 layer are called subnetworks, such as (Age-Net, Gender-Net, Hair-Net, etc). To explore the correlations between the high-level features of these different attributes, each subnetwork will extract features from other subnetworks to enhance the useful information of the original attributes. After that, we utilize TCCA to learn different degrees of correlations from global features to enhance features of each feature, and then transform these features into a highly correlated space through the generalization matrix to make the final attribute prediction. The neural network structure adopts multi-task joint learning method for learning. During experiments, we analyze the performance without TCCA and conclude that the degree of correlation between attributes is different. We use sufficient experiments to verify the performance of MTCN on the CelebA [40] and LFWA [40] datasets, and compared it with state-of-the-art methods. Experimental results show that MTCN achieves the best performance of all compared algorithms on CelebA and LFWA with 92.97% and 87.86% accuracy, respectively.

The rest of this article is organized as follows. Section 2 presents closely related works. Section 3 shows our method. The final results are analyzed in Section 4, and the article is concluded in Section 5.

2 RELATED WORK

2.1 Multi-attribute Prediction from Faces

Researches on face multi-attribute prediction started in the 1990s [6]. Since then, a large amount of work has been proposed and the early approaches mainly used handcrafted features to estimate attributes. Neeraj et al. [31] utilized a commercial face detector to preprocess images and then a separate SVM classifier learnt by extracted features are used to predict the face attributes. Based on biologically inspired feature (BIF), Guo et al. [18] used the canonical correlation analysis (CCA) and the partial least squares (PLS) methods for multi-attribute prediction.

Except for Reference [6], which used autoencoders to learn face attributes, all of the approaches put forward above utilized the handcrafted features. Lately, convolutional neural networks have made great success in computer vision applications. Dong et al. [65] designed a deep learning model to extract features from multi-scale patches and the CNN model was used to jointly predict face attributes via concatenating. Levi et al. [34] proposed two independent CNN models. Liu et al. [40] first located the face, and then used SVM to classify the extracted features. Uricar et al. [57] also utilized SVM to classify the face features extracted by CNN model. Yang et al. [68] used off-the-shelf CNN features to estimate 40 face attributes on CeleA and LFWA [40]. In Reference [23], a margin local embedding kNN (LMLE-kNN) approach was utilized to classify large-scale imbalanced attributes. Ehrlich et al. [12] proposed a new multi-task learning method to learn a shared feature representation and three multi-task public datasets were used to evaluate the proposed

model. Xie [61] proposed an online mulit-instance learning method for early expression detection and acquired better performance on video-based expression datasets. Kalayeh et al. [28] improved facial attribute prediction using semantic segmentation. In Reference [19], to excavate correlation and heterogeneity among face attributes, a Deep Multi-Task Learning (DMTL) approach was designed to predict multiple heterogeneous attributes of an image. An ensemble structure was proposed by Duan et al. [10] for age prediction that fully utilized feature enhancement and age grouping strategies. Cao [3] proposed a partially shared multi-task Convolutional Neural Network (PS-MCNN) to learn face attributes, which considered the identity information and attribute relationships simultaneously. Huang et al. [24] used a deep imbalanced learning method for face attribute estimation and achieved significant improvements in accuracy over compared approaches.

2.2 Canonical Correlation Analysis in Deep Learning

CCA was first proposed by Hotelling (1936) to find bases for two sets of variables so that the projections of the variables on these bases are maximally correlated [41]. The CCA tries to maximally preserve the useful/positive information [43] and it has been widely used in deep learning, mainly multi/cross-view learning/analysis [37, 51, 56, 59, 63], image annotation [44], the matching of (audio and articulation, audio and video, images and text, or text in two languages) [22, 52, 58], and so on.

Murthy et al. [44] proposed an effective method for image annotation, in which CNN is utilized to extract features from an image and word embedding vectors, and CCA-KNN explores the correlations among the image and vectors. [59] utilized deep canonically correlated autoencoders to learn multi-view features and achieved good performance. [62] used deep canonical correlation analysis (DCCA) to match images and captions in a joint latent space and addressed the high dimensionality features cased by DCCA. Yao et al. [64] proposed a novel Ranking CCA (RCCA) to learn query and image similarities and the satisfactory performance of the method are achieved via verifying with 11.7 million queries and one million images. [13] introduced deep discriminative canonical correlation analysis (DDCCA) to learn the nonlinear transformation among two datasets, which maximizes correlation within-class while minimizes correlation among inter-class. [63] proposed a CCA network (CCANet) to classify images and this method achieved good performance on several public datasets. Gao [16] utilized a labeled multiple CCA (LMCCA) to fuse and represent multimodal information, and datasets including from both audio and visual domains are used to verify its performance. Kim [29] proposed a TCCA to classify action/gesture in video and it significantly achieved better detection accuracy. Luo [41] used TCCA for multi-view dimension reduction and it aimed to maximize the canonical correlation of multiple views. The proposed algorithm performs well on various challenging tasks. This article attempts to apply TCCA to explore the correlations among face attributes and then utilizes the correlations to enhance the performance of final face attribute predictions.

3 PROPOSED METHOD

Our goal is to predict multiple face attributes through a joint estimation model. To learn features effectively, we try to leverage the attribute inter-correlations to enhance origin information. Figure 2 presents an example of strong pair-wise correlations among face attributes in the CelebA dataset.[1] We can see that the correlations among face attributes are different, some attributes are strongly related, and some attributes are generally related. Therefore, we propose a MTCN model to explore these correlations and use them to enhance the feature information of face attributes. Figure 3 presents the overall system framework. MTCN mainly consists of three parts: (1) shallow

[1] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

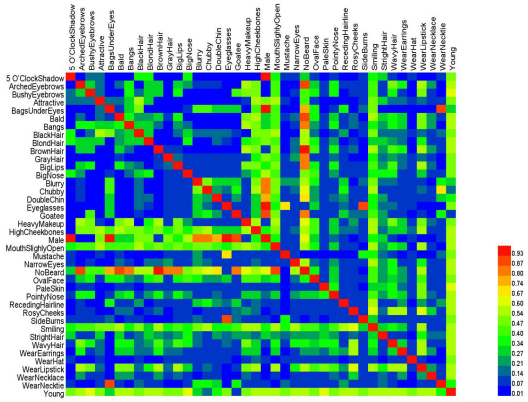

Fig. 2. Heat map of pair-wise correlations matrix of the 40 face attributes from the CelebA dataset.

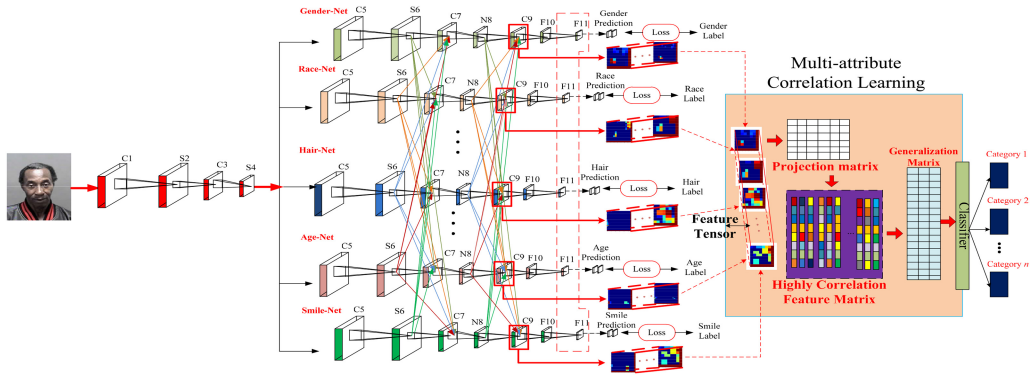

Fig. 3. Our system framework. (C1, . . . , C9) represent the convolutional layers, (S2, . . . , S6) denotes pooling and normalization operations, N8 is the normalization operation, and (F10 and F11) signifies the fully connected layers. MTCN shares all attribute features at the low-level layers, and then distinguishes each attribute feature at the high-level layers. To better excavate the correlations among high-level attribute features, each subnetwork explores useful information from other networks to enhance its original information. Then a tensor canonical correlation analysis (TCCA) method is used to seek the correlations among the features of the C9 layers, which enhances the original information of each attribute. After that, these features are mapped into a highly correlated space through the generalization matrix and final estimations are made based on these features.

feature sharing; (2) high-level feature differentiation and correlation; (3) excavation and utilization of global correlation. We will explain the three parts in detail later.

3.1 Network Structure Design

Many studies have successfully used shallow networks to predict face attributes and achieved good performance [3, 19, 20], and we utilize a five-layer convolutional network to identify face attributes. Table 4 presents the detailed parameters. Multi-CNN (MCNN) [20] has proven that the characteristics of different attributes of the same face image at the lower levels are basically the same, and it proposes a multi-task learning model for predicting multiple attributes of face images. In MCNN, the first two layers share attribute features, and the third layer of convolution starts to distinguish each attribute feature information. The MTCN we propose is based on the

Table 1. Network Structure Analysis

	Accuracy on CelebA	Accuracy on LFWA	Computational time on CelebA	Computational time on LFWA	No. Param. ($\times 10^9$)	No. FLOPs ($\times 10^9$)
Share-0	91.27%	85.93%	66 hours	40 hours	7.389	78.4
Share-1	90.21%	82.73%	10.57 hours	5.63 hours	7.389	67.55
Share-2	91.95%	86.67%	7.33 hours	3.08 hours	7.380	40.75
Share-3	87.43%	79.88%	4.31 hours	2.12 hours	7.346	14.95

"Share-0" → No shared layers, "Share-1" → C1 and S2 are shared, "Share-02" → C1, S2, C3, S4 are shared, "Share-03" → C1, S2, C3, S4, C5 are shared.

MCNN model while our MTCN will cross-extract high-level features of different networks to fully excavate the correlations among attributes. To make our proposed model more convincing, we give the recognition accuracy, system parameters, floating-point operations (FLOPs), and training and testing time of the same network structure in the case of different shared layers. In this time, TCCA is not used to excavate correlations, and other settings are the same. From Table 1, as the number of shared network layers increases, computational time, system parameters, and FLOPs decrease while the accuracy increases first and then decreases. We can conclude that if we share the first two convolutional layers, the system achieves the best performance. It should be emphasized that when there is no shared layer, there is no cross-extraction feature among the 40 subnetworks, mainly because each network processing process is not synchronized, and the final prediction is the average of the prediction results of all subnetworks. At this time, the calculation time of the system is the sum of the calculation times of all subnetworks.

3.2 Low-level Feature Sharing for Face Attributes

Different convolutional layers extract different abstract feature information, and the shallow convolutional layer extracts more spatial information [42]. Therefore, when using convolutional layers extract the features of face images, the spatial information are basically similar, for example, the corners of the face, the convexity of the nose, the positions of the eyes and the mouth, and so on. The high-level convolutional layer mainly extracts more semantic information, which is generally a unique feature of the input image, and it is also the most critical part for judging the category of the input image. At this time, high-level features involve fewer spatial features. According to the latest two works [20, 42], we can conclude that the first two convolutional layers contain more spatial information, so the information in this part of the different networks can be shared, reducing the computational overhead of multiple single-networks.

3.3 Differentiation and Correlation in High-level Layers

Since CNN can learn different abstract features during the process of training with different targets, we split the network into multi-subnetworks from the third convolutional layer. At this time, these abstract features are unique to attributes. MTCN uses multiple identical subnetworks to learn these feature information, and the same network structure of subnetworks is conducive to the learning and convergence of MTCN.

Meanwhile, according to References [7, 8, 10, 17], there are a lot of positive correlations among face attributes, and by seeking which attributes can enhance their original features, thereby improving the final prediction performance. As can be seen from Figure 3, each subnetwork excavates positive information from other subnetworks, and because the convolutional layer contains more semantic information, this operation occurs twice on the two convolutional layers C7 and C9.

In MTCN, the features extracted from the C5 layer to the F11 layer can fully represent the uniqueness of different attributes. To fully excavate the correlations between different attributes, we utilize the C7 and C9 layers of each subnetwork to extract the feature information of the S6 and N8 layers of other subnetworks. The S6 and N8 layers are not used to extract the C5 and C7 layer feature information, because the convolution operation can extract valuable information more effectively. Each convolution in the C5 and C7 layers of any subnetwork not only extracts feature information from its S6 and N7 layers but also extracts those feature information of other subnetworks. It is particularly emphasized that during this process, this type of convolution operation does not change the size and parameters of the convolution kernel.

Because MTCN utilizes multi-task learning for feature extraction, how to update the parameters of neural network model is an extremely important part of the entire learning process. Through theoretical analysis, we can conclude that the parameter learning of the convolutional layers is the most complicated part in the whole process of gradient learning. So in the following sections, we show the derivations and the implementation of this part in detail. Because the labels of each attribute are discrete, we use the cross-entropy loss function to calculate the system loss, which can be written as

$$\phi = -\frac{1}{N}\sum_{i=1}^{N}(y_i \ln p_i + (1-y_i)\ln(1-p_i)), \tag{1}$$

where p_i expresses the final result of an attribute, y_i denotes the label, and N is the number of training instances.

3.3.1 Gradients Transferred from the C9 Layer to the N8 Layer. During the process of feature extraction, MTCN extracts features not only from the feature map of its low-level layer but also from the feature map of the same layer of other subnetworks. Since the structure, the input dataset, and the learning process of the subnetwork are the same, we only introduce the derivations and the implementation process of Gender-Net in detail. Here, $\mathbf{w}_{nc}$ and $\mathbf{b}_{nc}$, and $\mathbf{w}_{cf}$ and $\mathbf{b}_{cf}$ denote the weights and biases of the C9 layer and the fully connected layer, respectively. K represents the number of subnetworks. At the same time, we assume that the outputs of the C9 and N8 layers of ith sample are X_c^i and X_n^i. To make the article easier to understand, we will treat the feature maps of each subnetwork as a whole, such as $(X_1^i, X_2^i, \ldots, X_K^i)$ represents the feature maps of the K subnetworks, so the feature extraction results of the C9 layer of Gender-Net can be computed as

$$X_c^i = f\left(X_1^i\mathbf{w}_{nc} + X_2^i\mathbf{w}_{nc} + \cdots + X_K^i\mathbf{w}_{nc} + \mathbf{b}_{nc}\right). \tag{2}$$

Based on the cross-entropy loss function, we can achieve the partial derivative of the weights and biases (For reader's convenience, the detailed derivation process can be seen in the supplementary file). We use η to denote the learning rate and the corresponding weights and biases can be updated as

$$w_{nc} = w_{nc} - \eta\frac{\partial\phi}{\partial w_{nc}}, \tag{3}$$

$$b_{nc} = b_{nc} - \eta\frac{\partial\phi}{\partial b_{nc}}. \tag{4}$$

3.3.2 Gradients Transferred from the N8 Layer to the S6 Layer. The parameter learning process from the C7 to S6 layers is the same as that of the upper layers. At this time, because the C9 layer extracts features not only from its own low-level layer network but also from other subnetworks, how the C7 layer learns the gradient of the upper layer is full of challenges. We use $\mathbf{w}'_{sc}$ and $\mathbf{b}'_{sc}$ to denote the weights and biases of the C7 layer, respectively, and $X_c^{i'}$ expresses the extracted

features of the C7 layer (for reader's convenience, the detailed derivation process can be seen in the supplementary file). After that, the weights and biases of the C7 layer can be updated as

$$w'_{sc} = w'_{sc} - \eta \frac{\partial \phi}{\partial w'_{sc}}, \tag{5}$$

$$b'_{sc} = b'_{sc} - \eta \frac{\partial \phi}{\partial b'_{sc}}. \tag{6}$$

3.3.3 Gradients Transferred from Subnetworks to a Shared Single Network. The parameter update process of the C5 layer is the same as the general gradient learning process and how to pass the gradients from multiple subnetworks to the shared network is the most critical thing. We propose a joint gradient transfer method to pass these gradients. We utilize $\phi_1, \phi_2, \ldots, \phi_K$ to represent the losses of the subnetworks. In addition, $\mathbf{w}'''_{sc}$ and $\mathbf{b}'''_{sc}$, and $\mathbf{w}''_{sc}$ and $\mathbf{b}''_{sc}$ represent the weights and biases of the C3 and C5 layers, respectively. The joint gradient transferred method is computed as

$$\Delta w = \frac{\partial \phi_1}{\partial w'''_{sc}} + \frac{\partial \phi_2}{\partial w'''_{sc}} + \cdots + \frac{\partial \phi_K}{\partial w'''_{sc}}, \tag{7}$$

$$\Delta b = \frac{\partial \phi_1}{\partial b'''_{sc}} + \frac{\partial \phi_2}{\partial b'''_{sc}} + \cdots + \frac{\partial \phi_K}{\partial b'''_{sc}}, \tag{8}$$

where $\frac{\partial \phi_1}{\partial w'''_{sc}}, \ldots, \frac{\partial \phi_K}{\partial w'''_{sc}}$ and $\frac{\partial \phi_1}{\partial b'''_{sc}}, \ldots, \frac{\partial \phi_K}{\partial b'''_{sc}}$ are computed using the chain rule.

As a consequence, the parameters of the C3 layer are updated as

$$w'''_{sc} = w'''_{sc} - \eta \Delta w, \tag{9}$$

$$b'''_{sc} = b'''_{sc} - \eta \Delta b. \tag{10}$$

3.4 Multi-attribute Tensor Correlation Learning Framework

3.4.1 Short Review of TCCA. The n-mode product of $\mathcal{X}$ with the matrix $U \in R^{J_n \times I_n}$ can be expressed as $\mathcal{M} = \mathcal{X} \times {}_nU$, which is a tensor with $I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \cdots \times I_N$, and the n-mode product is written as

$$M(i_1, \ldots, i_{n-1}, j_p, i_{n+1}, \ldots, i_N) = \sum_{i_n=1}^{I_n} \mathcal{X}(i_1, \ldots, i_N) U(j_n, i_n). \tag{11}$$

The product of $\mathcal{X}$ and a sequence of matrices $\{U_n \in R^{J_n \times I_n}\}_{n=1}^{N}$ is calculated as

$$\mathcal{M} = \mathcal{X} \times {}_1U_1 \times {}_2U_2 \times \cdots \times {}_NU_N. \tag{12}$$

The CANDECOMP/PARAFAC (CP) decomposition [4] decomposes an Nth-order tensor, $\mathcal{X} \in R^{I_1 \times I_2 \times \cdots \times I_N}$, into a linear combination of terms, $\mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}$, which are rank-1 tensors, and the process of decomposition is

$$\begin{aligned} \mathcal{X} &\cong \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} \\ &= \Lambda \times {}_1A^{(1)} \times {}_2A^{(2)} \times \cdots \times {}_NA^{(N)}. \end{aligned} \tag{13}$$

The variance matrices of m views are $H_{pp} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_{pn} \mathbf{x}_{pn}^T$, where X_p expresses samples of a view and $X_p = \{\mathbf{x}_{\mathrm{p1}}, \mathbf{x}_{\mathrm{p2}}, \ldots, \mathbf{x}_{\mathrm{pN}}\} \in R^{d_p \times N}$. After that, the covariance tensor of all views can be computed as

$$\mathcal{H}_{1,2,\ldots,m} = \frac{1}{N} \sum\nolimits_{n=1}^{N} \mathbf{x}_{1n} \circ \mathbf{x}_{2n} \circ \cdots \circ \mathbf{x}_{mn}, \tag{14}$$

where $\mathcal{H}$ is a $d_1 \times d_2 \times \cdots \times d_m$ tensor. Based on the classic CCA [21], the canonical variables and canonical vectors are $\mathbf{z}_p = X_p^T \mathbf{h}_p$ and $\{\mathbf{h}_p \in R^{d_p \times 1}\}_{p=1}^m$, respectively, where $p = 1, 2, \ldots, m$. As a consequence, the optimization function can be denoted as

$$\begin{aligned} \arg\max_{\{\mathbf{h}_p\}} &= \text{corr}(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m), \\ s.t.\ \mathbf{z}_p^T \mathbf{z}_p &= 1,\ p = 1, \ldots, m, \end{aligned} \tag{15}$$

where $(\mathbf{z}_1 \odot \mathbf{z}_2 \odot \ldots, \odot \mathbf{z}_m)^T \mathbf{e} = \text{corr}(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m)$ denotes the canonical correlation, $\odot$ expresses the element-wise product, and $\mathbf{e} \in R^N$. Based on TCCA [41], Equation (15) is equivalent to

$$\begin{aligned} \arg\max_{\{\mathbf{h}_p\}} &\ \rho = \mathcal{H}_{1,2,\ldots,m} \bar{\times}_1 \mathbf{h}_1^T \bar{\times}_2 \mathbf{h}_2^T \ldots \bar{\times}_m \mathbf{h}_m^T, \\ s.t.&\ \mathbf{h}_p^T H_{pp} \mathbf{h}_p = 1,\ p = 1, 2, \ldots, m, \end{aligned} \tag{16}$$

where $\bar{\times}_p$ denotes the p-mode contracted tensor-vector product. Let $\mathrm{u}_p = \widetilde{H}_{pp}^{1/2} \mathbf{h}$ and $\mathcal{M} = \mathcal{H}_{1,2,\ldots,m} \bar{\times}_1 \widetilde{H}_{11}^{1/2} \mathbf{h} \bar{\times}_2 \widetilde{H}_{22}^{1/2} \mathbf{h} \bar{\times}_3 \ldots \bar{\times}_m \widetilde{H}_{mm}^{1/2} \mathbf{h}$. Then, the optimization problem in Equation (16) is described as

$$\begin{aligned} \arg\max_{\{\mathbf{h}_p\}} &\ \rho = \mathcal{M} \bar{\times}_1 \mathbf{u}_1^T \bar{\times}_2 \mathbf{u}_2^T \bar{\times}_3 \ldots \bar{\times}_m \mathbf{u}_m^T, \\ s.t.&\ \mathbf{u}_p^T \mathbf{u}_p = 1,\ p = 1, 2, \ldots, m, \end{aligned} \tag{17}$$

where $\widetilde{H}_{pp} = H_{pp} + \epsilon I$, I expresses the identity matrix, and ϵ denotes a nonnegative trade-off parameter.

According to Reference [33], Equation (17) is equivalent to following formula:

$$\mathcal{M} \approx \sum_{k=1}^{r} \rho_k \mathbf{u}_1^{(k)} \circ \mathbf{u}_2^{(k)} \circ \cdots \circ \mathbf{u}_p^{(k)}. \tag{18}$$

The alternating least squares (ALS) method [5, 30] is utilized to find approximate solutions. Let $U_p = [\mathbf{u}_p^{(1)}, \ldots, \mathbf{u}_p^{(r)}]$, the mapped features for the pth view are computed as

$$Z_p = X_p^T \widetilde{H}_{pp}^{-1/2} U_p. \tag{19}$$

After that, TCCA concatenates the whole $Z_p{}_{p=1}^m$ to achieve the final representation $Z \in R^{(mr) \times N}$, which is used as input for subsequent operations.

3.4.2 Multi-attribute Tensor Correlation Learning. The neural networks of MTCN mainly seek the correlations among attributes from abstract features, thereby enhancing the original information of each attribute and improving the accuracy of attribute estimation. However, the degrees of correlation between attributes are different, and the subnetworks do not fully consider this factor. Therefore, we use TCCA [41] to further explore the different degrees of correlation among high-level features.

To fully utilize the correlation between the different attributes of the C9 layer, we assume that $\mathbf{X}_l^i = \{\{X_1^1, X_2^1, \ldots, X_L^1\}, \{X_1^2, X_2^2, \ldots, X_L^2\}, \ldots, \{X_1^K, X_2^K, \ldots, X_L^K\}\}$, where $l = 1, 2, \ldots, L$, L expresses the number of feature maps of C9 layer, and $i = 1, 2, \ldots, K$. X_l^i signifies a 3-D tensor, and $\kappa \times \kappa$ represents the size of the feature map in C9 layer. $\mathcal{X} \in R^{\kappa \times \kappa \times KL}$, where KL expresses the number of all feature maps. According to TCCA, the feature maps in C9 layer can be expressed as $\{X_p\}_{p=1}^{KL}$ and $X_p = \{\mathbf{x}_{p1}, \mathbf{x}_{p2}, \ldots, \mathbf{x}_{p\kappa}\} \in R^{\kappa \times \kappa}$, and we calculate the variance matrices as $H_{pp} = \frac{1}{\kappa} \sum_{j=1}^{k} \mathbf{x}_{pj} \mathbf{x}_{pj}^T$. The covariance tensor among $X_1, X_2, \ldots, X_{KL}$ is computed as

$$\mathcal{H}_{1,2,\ldots,(KL)} = \frac{1}{\kappa} \sum_{j=1}^{\kappa} \mathbf{x}_{1j} \circ \mathbf{x}_{2j} \circ \cdots \circ \mathbf{x}_{(KL)j}, \tag{20}$$

where $\circ$ denotes the outer product and $\mathcal{H}$ expresses a $\kappa \times \kappa \times \cdots \times \kappa$ tensor.

The canonical correlation is denoted as

$$\begin{aligned} &\arg\max \rho = corr(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{KL}), \\ &s.t.\ \mathbf{z}_p^T \mathbf{z}_p = 1,\ p = 1,\ 2, \ldots, (KL), \end{aligned} \tag{21}$$

where the canonical variables $\mathbf{z}_p = X_p^T \mathbf{h}_p$.

According to TCCA, Equation (21) is equivalent to

$$\begin{aligned} \arg\max \rho &= \sum_{j=1}^{\kappa} z_1(j) z_2(j) \cdots z_{KL}(j), \\ &= \sum_{j=1}^{\kappa} \prod_{p=1}^{KL} z_p(j) = \sum_{j=1}^{\kappa} \prod_{d=1}^{KL} \left(\sum_{k_p=1}^{\kappa_p} x_{p,j}(k_p) h(k_p) \right), \end{aligned} \tag{22}$$

where $z_p(j)$ denotes the jth element of $\mathbf{z}_p$, and $x_{p,j}(k_p)$ and $h(k_p)$ denote the k_pth elements of $\mathbf{x}_{p,j}$ and $\mathbf{z}_p$.

Furthermore,

$$\begin{aligned} &(\mathcal{H} \bar{\times}_p \mathbf{h}_p^T)(k_1,\ k_2, \ldots, k_{p-1}, k_{p+1}, \ldots, k_{KL}) \\ &= \sum_{k_p=1}^{\kappa_p} \mathcal{H}(k_1,\ k_2, \ldots, k_{KL}) \mathbf{h}(k_p) \\ &= \sum_{j=1}^{\kappa} \sum_{k_p=1}^{\kappa_p} \left(\prod_{p=1}^{KL} \mathbf{x}_{p,j}(k_p) \right) \mathbf{h}(k_p), \end{aligned} \tag{23}$$

where $\bar{\times}_p$ expresses the p-mode contracted tensor-vector product. We can learn that Equation (29) is equivalent to Equation (23), that is,

$$corr(z_1,\ z_2, \ldots, z_{KL}) \ = \ \mathcal{H}_{1,2,\ldots,KL} \bar{\times}_1 h_1^T \bar{\times} \cdots \bar{\times}_{KL} h_{KL}^T. \tag{24}$$

According to the TCCA, Equation (24) is further denoted as

$$\begin{aligned} &\mathcal{H}_{1,\ldots,KL} \bar{\times}_1 \mathbf{h}_1^T \bar{\times}_2 \mathbf{h}_2^T \times \cdots \bar{\times}_{KL} \mathbf{h}_{KL}^T \\ &= \mathbf{h}_{KL}^T C_{(KL)} (\mathbf{h}_{KL-1} \otimes \cdots \otimes \mathbf{h}_2 \otimes \mathbf{h}_1) \\ &= \mathbf{u}_{KL}^T \widetilde{H}_{(KL)(KL)}^{-1/2} H_{(KL)} \\ &\quad \cdot \left(\left(\widetilde{H}_{KL-1,KL-1}^{-1/2} \mathbf{u}_{KL-1} \right) \otimes \cdots \otimes \left(\widetilde{H}_{1,1}^{-1/2} \mathbf{u}_1 \right) \right) \\ &= \mathbf{u}_{KL}^T \widetilde{H}_{(KL)(KL)}^{-1/2} H_{(KL)} \\ &\quad \cdot \left(\widetilde{H}_{KL-1,KL-1}^{-1/2} \otimes \cdots \otimes \widetilde{H}_{1,1}^{-1/2} \right) \quad (\mathbf{u}_{KL-1} \otimes \cdots \otimes \mathbf{u}_1), \end{aligned} \tag{25}$$

where $\mathbf{h}_p^T H_{pp} \mathbf{h}_p = 1$, $\mathbf{u}_p = \widetilde{H}_{pp}^{1/2} \mathbf{h}_p$, and $\mathcal{F} = \mathcal{H}_{1,2,\ldots,KL} \bar{\times}_1 \widetilde{H}_{11}^{1/2} \bar{\times}_2 \widetilde{H}_{22}^{1/2} \bar{\times} \cdots \bar{\times}_{KL} \widetilde{H}_{(KL)(KL)}^{1/2}$.

Therefore,

$$\mathcal{H}_{1,\ldots,KL} \bar{\times}_1 \mathbf{h}_1^T \bar{\times} \cdots \bar{\times}_{KL} \mathbf{h}_{KL}^T = \mathcal{F} \bar{\times}_1 \mathbf{u}_1^T \bar{\times} \cdots \bar{\times}_{KL} \mathbf{u}_{KL}^T. \tag{26}$$

Based on the analysis of Reference [33], Equation (26) is aimed to find the best rank-1 approximation of $\mathcal{F}$. Let $\hat{\mathcal{F}} = \rho \mathbf{u}_1 \circ \mathbf{u}_2 \circ \cdots \circ \mathbf{u}_{KL}$. Our aim is

$$\arg\max_{\mathbf{u}_p} ||\mathcal{F} - \hat{\mathcal{F}}||. \tag{27}$$

We use ALS method to optimize the problem, and we can obtain the solution $\mathbf{u}_p$. After that, the canonical variables are calculated as $\mathbf{z}_p = X_p^T \mathbf{h}_p = X_p^T \tilde{H}_{pp}^{-1/2} \mathbf{u}_p$. Let $U_p = [\mathbf{u}_p^{(1)}, \ldots, \mathbf{u}_p^{(r)}]$ and

Table 2. Complexity Analysis

Method	Input Size	No. Param. $\times 10^6$	No. FLOPs $\times 10^9$	Trunk Depth	Method	Input Size	No. Param. $\times 10^6$	No. FLOPs $\times 10^9$	Trunk Depth	Method	Input Size	No. Param. $\times 10^6$	No. FLOPs $\times 10^9$	Trunk Depth
MCNN	227 × 227	320	8.6	5	PS-MCNN	192 × 160	16	6.7	7	MTCN	224 × 224	7380	40.75	7

$\mathbf{z}_p^{(1)}, \ldots, \mathbf{z}_p^{(r)}$ express the column vectors of Z_p. U signifies the transformation matrix. As a consequence, the mapped features for the p'th attribute can be computed as

$$Z_p = X_p^T \widetilde{H}_{pp}^{-1/2} U_p. \tag{28}$$

According to TCCA, Equation (21) is equivalent to $H_{1,2,\ldots,(KL)} \bar{\times}_1 \mathbf{h}_1^T \bar{\times}_2 \mathbf{h}_2^T \bar{\times} \cdots \bar{\times}_{(KL)} \mathbf{h}_{(KL)}^T$, and Equation (22) can be denoted as

$$\begin{aligned} \arg\max \rho &= H_{1,2,\ldots,(KL)} \bar{\times}_1 \mathbf{h}_1^T \bar{\times} \cdots \bar{\times}_{(KL)} \mathbf{h}_{(KL)}^T, \\ s.t.\ \mathbf{h}_p^T H_{pp} \mathbf{h}_p &= 1, \end{aligned} \tag{29}$$

where $H_{pp} = X_p X_p^T$.

Based on our previous analysis, the ALS method is utilized to solve approximate solutions. We set $U_p = [\mathbf{u}_p^{(1)}, \ldots, \mathbf{u}_p^{(r)}]$, and the mapped feature for the p'th view is computed as

$$Z_p = X_p^T \widetilde{H}_{pp}^{-1/2} U_p. \tag{30}$$

After that, the final representation can be denoted as $Z \in R^{(KLr)\times\kappa}$ via concatenating the different $\{Z_p\}_{p=1}^{(KL)}$. Since the above method only achieves the correlation among different attributes for one face image, to ensure MTCN have better generalization performance for whole dataset, we construct a generalization matrix to ensure that the mapped features of each image are highly correlation. During the parameter learning of the generalization matrix, the previous neural network is not updated. Because MTCN adopts a multi-task learning model to recognize face multi-attributes, we use a joint attribute estimation method to calculate the whole loss of MTCN to update its relevant parameters. If an image has ω attributes, then a joint attribute prediction system is

$$\epsilon = \arg\ \min \sum_{i=1}^{\omega} H_i + \gamma \Phi(W), \tag{31}$$

where H_i expresses the loss of the ith attribute, W expresses the weights of generalization matrix, $\Phi(\cdot)$ is used to penalize the complexity of the weights, and $\gamma > 0$ denotes a regularization parameter.

We utilize CelebA and LFWA datasets [40] to test the performance of MTCN. The whole process of MTCN is roughly summarized as the follows:

Step 1: Utilize CelebA or LFWA dataset to train MTCN without TCCA and then a fine-tuned model is utilized to estimate face attributes, and Equation (1) is as the loss function in this process; Step 2: Use one third of the training datasets to train the generalization matrix with TCCA and Equation (31) is as the loss function in this time; Step 3: Utilize the testing datasets to validate the performance of MTCN.

3.5 Complexity Analysis

We analyze the time and memory complexity of MCNN [20], PS-MCNN-LC [3], and MTCN via FLOPs. As can be seen from Table 2, MTCN consumes about 55% more FLOPs and has 10 times

Table 3. Summary of the 40 Face Attributes Provided in the CelebA Dataset

Attr. Idx.	Attr. Def	Attr. Idx.	Attr. Def	Attr. Idx	Attr. Def	Attr. Idx.	Attr. Def	Attr. Idx.	Attr. Def	Attr. Idx.	Attr. Def	Attr. Idx	Attr. Def	Attr. Idx.	Attr. Def
1	5 O'ClockShadow	6	Bald	11	GrayHair	16	DoubleChin	21	Male	26	OvalFace	31	SideBurns	36	WearHat
2	ArchedEyebrows	7	Bangs	12	BigLips	17	Eyeglasses	22	MouthSlighlyOpen	27	PaleSkin	32	Smiling	37	WearLipstick
3	BushyEyebrows	8	BlackHair	13	BigNose	18	Goatee	23	Mustache	28	PointyNose	33	StrightHair	38	WearNecklace
4	Attractive	9	BlondHair	14	Blurry	19	HeavyMakeup	24	NarrowEyes	29	RecedingHairline	34	WavyHair	39	WearNecktie
5	BagsUnderEyes	10	BrownHair	15	Chubby	20	HighCheekbones	25	NoBeard	30	RosyCheeks	35	WearEarrings	40	Young

more parameters than PS-MCNN-LC. MCNN uses a MTL structure to recognize face attributes, which leads to its most parameters, which is also the main reason why MTCN has so many parameters. PS-MCNN-LC adopts a partially shared multi-task learning model, which is why it has fewer parameters. MTCN consumes the most FLOPs and needs the most memory, because it needs to excavate the correlation information from different subnetworks twice. However, these operations can enhance the original information of each attribute. From Table 5, we also see that each prediction result of face attributes on CelebA is relatively stable, and there is no large fluctuation prediction result. These operations also ensure MTCN gets the best results on the LFWA dataset. The corresponding computational time is analyzed in detail in Sections 4.3.4 and 4.3.5.

4 EXPERIMENTS

4.1 Datasets

4.1.1 CelebA. CelebA [40] contains 200,000 images and each image has 40 attributes (see Table 3): 160,000, 20,000, and 20,000 are utilized for training, validation, and testing, respectively. We did not augment the CelebA dataset, because it is large enough.

4.1.2 LFWA. LFWA [40] is another dataset containing 40 face attributes and its face images are from the LFW dataset [25]. Its attribute annotations are the same as these in CelebA. The LFWA contains only 13,143 photos. Most studies adopt 6,263 photos as the training dataset, and 6,880 photos are utilized for testing. However, due to the complexity of MTCN, these dataset are difficult to train a model well, so we adopt data augmentation method proposed in Reference [20] to augment the original photos, and then we achieve more than 75,000 photos as the training dataset.

4.2 Implementation Details

We implement MTCN with Tensorflow [1] on NVIDIA Tesla P100. The selection method for crop is the same as that used in ref. [11] and dropout methods are used to reduce overfitting. We initialize the weights of MTCN with Gaussian distribution. We set the mean, standard deviation, and base learning rate as 0, 0.01, and 10^{-4}, respectively. Every 100,000 iterations, the learning rate is reduced by 10%. During the MTCN training, a batch size of 100 is utilized and MTCN is trained for 30 epochs on CelebA or LFWA. As a whole, it takes approximately 10 and 4 h to train MTCN with TCCA on CelebA and LFWA, respectively, and the training of the generalization matrix takes nearly 1.5 h. We conduct each experiment ten times, after that we achieve the corresponding average results. We just present the results of the baseline methods, because codes of these methods used in corresponding publications are not available in the public domain.

4.2.1 Network Structure. The neural network structure of MTCN consists of a shared network and 40 subnetworks, and all subnetworks have the same structures. The detailed subnetwork configurations are shown in Table 4. A max pooling layer and a local response normalization layer

Table 4. Subnetwork Parameters

Layers	Parameters	Layers	Parameters	Layers	Parameters	Layers	Parameters	Layers	Parameters	Layers	Parameters
Conv1	Num_output: 96 Kernel_size: 5 Stride: 2	Pool1	Num_output: 96 Kernel_size: 3 Stride: 2	Norm1	Local_size: 5 alpha: 1e-1 beta: 0.75	Conv3	Num_output: 384 Kernel_size: 3 pad: 1	Pool3	Num_output: 384 Kernel_size: 3 Stride: 2	Norm3	Local_size: 5 alpha: 1e-1 beta: 0.75
Conv2	Num_output: 256 Kernel_size: 3 Stride: 1	Pool2	Num_output: 256 Kernel_size: 3 Stride: 2	Norm2	Local_size: 5 alpha: 1e-1 beta: 0.75	Conv4	Num_output: 384 Kernel_size: 3 Stride: 1	Norm4	Local_size: 5 alpha: 0.01 beta: 0.75	Conv5	Num_output: 256 Kernel_size: 3 Stride: 1

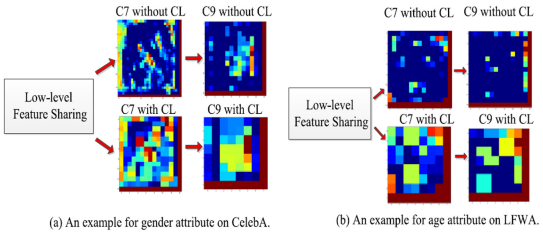

Fig. 4. Examples of positive information learnt by the correlation explored process. C1/C3/C3/C7/C9 denote the corresponding convolutional layers of MTCN. CL expresses correlation learning.

follow the convolutional layer. Both F10 and F11 layers have 4,098 units and are followed by a ReLU. We use 50% dropout method to reduce overfitting and the final layer is with 1,000 units.

4.2.2 Examples of Feature Learning via MTCN. Our MTCN tries to fully excavate the correlations among face attributes to enhance the useful information of origin features. We want to know what has learnt via correlation learning that ensures our MTCN achieving best performance among compared methods instead of using deep network, such as DenseNet and ResNet. Due to that the correlation learning process of each attribute is same, we just present each convolutional layer features of an image via fine-tuned Age-Net and Gender-Net whether using correlation learning. Figure 4 presents examples of positive information learnt by the correlation explored process for gender/age attributes.

Although we use a small network rather than a DenseNet or ResNet, it achieves better performance than DMTL [19] with deep network. By capturing the correlation among these attributes to enhance the original features, our work pursues better performance by applying width of the network instead of "deep" network, and the "width" networks can be easily fine-tuned, which is the reason why we have not adopted the DenseNet and ResNet. In other words, it is just because of without using the deep models that the correlation learning in our model plays an important role in the whole predictions. Figure 4 presents the positive information learnt by correlation explored process. Under the situation of the same prediction system, a feature with more gender/age-related information may achieve the best performance. From Figure 4, we can learn that Age-Net and Gender-Net without the correlation learning, it just learns part of age-related information; however, with the correlation learning, positive effects of other facial attributes on gender/age are learnt to enhance the original feature and the new features are with more age-related information. The same positive information are learnt by the decompositions.

Generalization matrices for CelebA and LFWA are visualized in Figures 5 and 6. The goal of the matrix is to make the correlation feature matrix more robust and smoothness for each image in dataset. The size of matrix is 40×40 and red and yellow colors indicate high values. Through the TCCA process, the correlations among face attributes are fully exploited and utilized. The

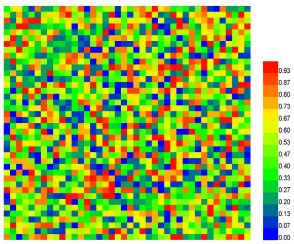

Fig. 5. Heat map of generalization matrix for CelebA.

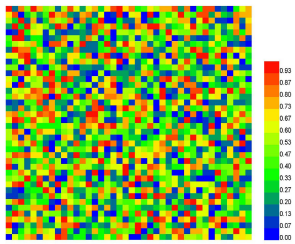

Fig. 6. Heat map of generalization matrix for LFWA.

generalization matrix projects the correlation feature matrix into a new space that ensures the outputs of TCCA for the whole image being not a lot of volatility.

4.3 Results

The prediction accuracies for face attributes on CelebA and LFWA by our method and several compared approaches [3, 9, 11, 19, 20, 26, 40, 50] are presented in Table 5. The MTCN with TCCA outperforms [9, 11, 19, 20, 26, 40, 50] for most of the 40 face attributes on both datasets. In terms of the average accuracies for CelebA, our MTCN with TCCA improves on Reference [40] by 5.67%, on Reference [50] by 2.03%, on Reference [20] by 1.68%, on Reference [19] by 0.37%, on Reference [9] by 1.74%, on Reference [26] by 1.34%, and on Reference [11] by 0.61%. Although the performance of the PS-MCNN-LC algorithm is similar to our proposed algorithm, the predictions of our algorithm for each attribute are relatively uniform while those of the PS-MCNN-LC algorithm fluctuate greatly, for example, the prediction result of attributes (♯ 12) is only 73.13% while that of our algorithm is 89.28%. In terms of the average accuracies for LFWA, our MTCN with TCCA improves on Reference [40] by 4.03%, on Reference [28] by 2.58%, on Reference [20] by 1.55%, on Reference [19] by 1.71%, on Reference [26] by 1.7%, and on Reference [3] by 0.5%. Although MTCN obtains the best performance among all the compared methods, we still do not know whether MTCN with TCCA is effective for the overall attributes or for some attributes, so we conduct further exploration and research.

4.3.1 Ablation Analyses on the CelebA Dataset. Since there is no strong correlation between some face attributes, MTCN will not enhance the features of all attributes, but most attributes have better prediction results. According to the results in Table 5, we divide the attributes into three categories: **(**I) attributes (1, 5, 6, 7, 10, 11, 14, 15, 16, 17, 18, 21, 23, 25, 27, 30, 31, 36, 39), most of their prediction accuracies exceed 95% via MTCN with TCCA, while these using compared methods are less than 95%. The main reason is that there are strong correlations among these attributes and by fully excavating which, MTCN improves the prediction performance of each attribute. The compared algorithms do not make full use of these correlations, resulting in poor prediction results for some attributes; **(**II) the predictions of attributes (26 and 28) are lower than 80%, and the main reason for those is that there is almost no correlation between these two attributes. The attributes in category **(**III) have a strong one-line correlation with those in category **(**I). In other words, the attributes of the former can well enhance the prediction performance of attributes in category **(**I). On the contrary, the attributes in category **(**I) cannot significantly enhance the attributes in category **(**III). For example, {20 (*HighCheekbones*) and (25 (*NoBeard*), 32 (*Smiling*))}, {25 (*NoBeard*) and 3 (*BushyEyebrows*)}, and {2 (*ArchedEyebrows*) and 25 (*NoBeard*)}.

Table 6 provides statistics on the prediction results of the three categories by different methods. First, the result for category **(**I) with MTCN without TCCA is 96.46%, which improves on

Table 5. The Accuracies (in %) of 40 Binary Attribute (See Table 2) Predictions on the CelebA and LFWA by the MTCN and the Latest Methods [3, 9, 11, 19, 20, 26, 40, 50]

	Approach	Attribute index																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CelebA	LENet+ANet [40]	84.00	82.00	83.00	83.00	88.00	88.00	75.00	81.00	90.00	97.00	74.00	77.00	82.00	73.00	78.00	95.00	78.00	84.00	95.00	88.00
	MOON [50]	94.03	82.26	81.67	84.92	98.77	95.80	71.48	84.00	89.40	95.86	95.67	89.38	92.62	95.44	96.32	99.47	97.04	98.10	90.99	87.01
	MCNN+AUX [20]	94.51	83.42	83.06	84.92	98.90	96.05	71.47	84.53	89.78	96.01	96.17	89.15	92.84	95.67	96.32	99.63	97.24	98.20	91.55	87.58
	DMTL [19]	95.00	86.00	85.00	85.00	99.00	99.00	96.00	85.00	91.00	96.00	96.00	88.00	92.00	96.00	97.00	99.00	99.00	98.00	92.00	88.00
	PaW [9]	94.64	83.01	82.86	84.58	98.93	95.93	71.46	83.63	89.84	95.85	96.11	88.50	92.62	95.46	96.26	99.59	97.38	98.21	91.53	87.44
	GNAS [26]	94.76	84.25	92.99	83.06	85.87	98.96	96.2	90.24	96.11	89.75	98.37	71.79	85.1	96.42	96.93	96.48	99.69	97.59	91.82	88.05
	TCFN [11]	94.81	84.53	85.94	84.13	99.01	98.42	95.8	85.81	90.63	96.11	96.67	88.93	92.31	95.73	96.43	96.38	98.32	98.17	91.93	88.29
	PS-MCNN-LC [3]	96.6	85.77	94.51	84.39	87.29	99.41	98	91.66	97.93	91.03	98.66	73.13	86.4	98	97.66	98.29	99.85	97.74	93.31	89.5
	MTCN without TCCA	94.68	84.92	84.71	85.11	98.05	97.73	86.04	84.18	90.42	95.47	95.13	88.48	91.37	95.49	96.18	99.03	98.42	98.10	91.47	87.19
	MTCN with TCCA	95.46	86.02	86.23	85.97	99.12	99.42	95.44	86.03	91.14	96.82	96.44	89.28	92.00	96.32	97.16	99.68	98.73	98.59	92.34	88.95
LFWA	LENet+ANet [40]	84.00	82.00	83.00	83.00	88.00	88.00	75.00	81.00	90.00	97.00	74.00	77.00	82.00	73.00	78.00	95.00	78.00	84.00	95.00	88.00
	MCNN+AUX [20]	77.06	81.78	80.31	83.48	91.94	90.08	79.24	84.98	92.63	97.41	85.23	80.85	84.97	76.86	81.52	91.30	82.97	88.93	95.85	88.38
	DMTL [19]	80.00	86.00	82.00	84.00	92.00	93.00	77.00	83.00	92.00	97.00	89.00	81.00	80.00	75.00	78.00	92.00	86.00	88.00	95.00	89.00
	PS-MCNN-LC [3]	78.17	83.53	85.72	81.84	86.74	92.6	91.45	92.96	98.51	81.87	91.04	82.7	86.48	87.2	78.11	86.7	92.78	84.11	96.6	88.77
	MTCN without TCCA	80.39	85.10	82.23	83.71	91.94	92.65	80.53	84.46	92.19	97.29	87.85	80.86	83.03	78.85	80.13	91.42	85.48	88.63	95.61	88.55
	MTCN with TCCA	81.68	86.13	83.01	84.29	92.13	93.27	84.33	85.06	93.18	97.89	89.26	81.61	84.43	83.19	81.94	92.84	87.03	89.66	96.25	89.57
	Approach	Attribute index																			
		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
CelebA	LENet+ANet [40]	94.00	82.00	92.00	81.00	79.00	74.00	84.00	80.00	85.00	78.00	77.00	91.00	76.00	76.00	94.00	88.00	95.00	88.00	79.00	86.00
	MOON [50]	98.10	93.54	96.82	86.52	95.58	75.73	97.00	76.46	93.56	94.82	97.59	92.60	82.26	82.47	89.60	98.95	93.93	87.04	96.63	88.08
	MCNN+AUX [20]	98.17	93.74	96.88	87.23	96.05	75.64	97.05	77.47	93.81	95.16	97.55	92.73	83.58	83.71	90.43	99.05	94.11	86.63	96.51	88.38
	DMTL [19]	98.00	94.00	97.00	90.00	97.00	78.00	97.00	78.00	94.00	96.00	98.00	94.00	85.00	87.00	91.00	99.00	93.00	89.00	97.00	90.00
	PaW [9]	98.39	94.05	96.90	87.56	96.22	75.03	97.08	77.35	93.44	95.07	97.64	92.73	83.52	84.07	89.93	99.02	94.24	87.70	96.85	88.59
	GNAS [26]	98.5	94.16	97.03	88.66	96.3	75.57	97.24	78.24	93.94	95.01	97.96	93.24	84.77	84.52	90.98	99.12	94.41	87.61	96.76	88.89
	TCFN [11]	98.43	94.23	97.02	88.95	96.83	77.49	97.03	77.93	94.14	95.33	98.07	93.41	84.93	85.73	90.74	99.04	94.28	88.83	96.83	86.92
	PS-MCNN-LC [3]	98.81	95.99	98.56	89.07	98.03	77.43	98.84	79.32	95.85	96.92	98.22	94.85	85.96	86.39	92.66	99.43	95.7	88.98	98.52	90.54
	MTCN without TCCA	98.43	93.89	96.59	88.97	96.71	76.35	97.04	77.81	93.92	95.78	97.91	93.07	84.98	86.54	90.17	98.91	93.18	88.76	97.00	89.95
	MTCN with TCCA	98.52	94.61	97.18	89.42	97.31	78.52	97.18	78.47	94.35	96.00	98.34	93.91	85.49	87.00	91.04	99.10	94.00	89.31	97.26	90.71
LFWA	LENet+ANet [40]	94.00	82.00	92.00	81.00	79.00	74.00	84.00	80.00	85.00	78.00	77.00	91.00	76.00	76.00	94.00	88.00	95.00	88.00	79.00	86.00
	MCNN+AUX [20]	94.02	83.51	93.43	82.86	82.15	77.39	93.32	84.14	86.25	87.92	83.13	91.83	78.53	81.61	94.95	90.07	95.04	89.94	80.66	85.84
	DMTL [19]	93.00	86.00	95.00	82.00	81.00	75.00	91.00	84.00	85.00	86.00	80.00	92.00	79.00	80.00	94.00	92.00	93.00	91.00	81.00	87.00
	PS-MCNN-LC [3]	95.18	84.6	94.47	83.51	82.01	77.9	94.97	87.52	87.5	88.81	84.42	92.7	79.65	83.35	95.54	91.21	95.7	90.92	82.18	86.88
	MTCN without TCCA	93.48	85.34	94.21	82.36	81.90	77.43	92.25	83.79	85.42	87.01	82.49	91.69	78.43	81.06	95.04	91.38	94.39	90.62	80.96	86.63
	MTCN with TCCA	94.16	85.61	95.46	83.42	82.39	78.71	93.59	84.91	87.06	88.41	84.21	92.67	80.00	81.45	95.76	92.34	95.59	91.74	82.03	88.04

The average accuracies of [3, 9, 11, 19, 20, 26, 40, 50], and the proposed approaches are 92.98%, 91.23%, 92.36%, 92.60%, 91.29%, 91.63%, 87.30%, 90.94%, 91.95% (Ours), and 92.97% (Ours), respectively, on CelebA, and these of References [3, 19, 20, 40], and the proposed approaches are 87.36%, 86.15%, 86.31%, 83.85%, 86.67% (Ours), and 87.86% (Ours), respectively, on LFWA.

Reference [20] by 0.99%, on Reference [26] by 0.2%, on Reference [40] by 13.04%, and on Reference [50] by 1%. With TCCA, MTCN improves the average accuracy by 1.12% compared to that without TCCA. Second, for category (II), for the average accuracies of References [20, 40, 50], [9, 19, 26], our MTCN without TCCA, and our MTCN with TCCA are 77%, 76.1%, 75.31%, 75.31%, 78%, and 76.9%, respectively. We can conclude that MTCN with TCCA obtains the best accuracy. Finally, the result for category (III) with TCCA is 89.88% and it performs better than most comparison algorithms.

Through the above detailed analysis, we can first conclude that there are correlations among face attributes. How to make good use of these correlations can improve the prediction performance of attributes. Second, the correlations among some attributes is weak. The prediction performance at this time is completely dependent on the training effect of the model on the corresponding attribute dataset. Finally, the degrees of the correlations between attributes are not the same. Some attributes have an enhanced effect on other attributes, on the contrary, other attributes do not have such an effect. Our MTCN fully explores the correlations between attributes from several aspects, and

Table 6. The Average Results of the Three Categories on CelebA

Methods	Category I	Category II	Category III	Methods	Category I	Category II	Category III
LENet+ANet [40]	83.42%	77%	85%	GNAS [26]	96.26%	76.9%	88.66%
MOON [50]	95.46%	76.1%	87.99%	TCFN [11]	97.08%	77.71%	89.19%
MCNN+AUX [20]	95.47%	75.31%	88.18%	PS-MCNN-LC [3]	97.36%	78.38%	90.66%
DMTL [19]	95.14%	75.56%	87.06%	MTCN without TCCA	96.46%	77.08%	89.01%
PaW [9]	97.31%	78%	89.42%	MTCN with TCCA	97.58%	78.49%	89.88%

Table 7. The Average Results of the Three Categories on LFWA

Methods	Category I	Category II	Category III	Methods	Category I	Category II	Category III
LENet+ANet [40]	89.17%	74%	79.76%	PS-MCNN-LC [3]	87.58%	82.71%	88.03%
MCNN+AUX [20]	91.38%	77.39%	82.39%	MTCN without TCCA	91.73%	77.43%	82.32%
DMTL [19]	91.67%	75%	81.95%	MTCN with TCCA	92.73%	78.71%	83.69%

considers the degrees of the correlations of high-level features through the TCCA algorithm, and finally uses a generalization matrix to ensure that MTCN has good generalization performance.

4.3.2 Ablation Analyses on the LFWA Dataset. The LFWA is a smaller dataset compared to the CelebA, so the overall prediction results are lower than those of CelebA. Although MTCN still obtains the best prediction results among all compared methods, the trends in the prediction results for some attributes on LFWA are different from those on CelebA. For example, *Bangs* (7) on LFWA belongs to category (II) and its prediction result is 84.33%, while *Bangs* belongs to category (I) on CelebA. Even though LFWA is a relatively small dataset, the prediction results of most attributes are close to those on CelebA. One important reason is that the data augmentation method is used before training. In fact, a more important reason is that MTCN fully excavates the correlations among attributes to obtain better prediction performance.

For better analysis and comparison, we divide the attributes into three categories according to the prediction results in Table 5. The three categories are: (I) for attributes (5, 6, 9, 10, 11, 16, 18, 19, 20, 21, 23, 27, 30, 32, 35, 36, 37, 38, 40), most of the average accuracies reach 90% while the average accuracies of the compared methods are less than 90%; (II) the result of attribute (26) is lower than 80%; and (III) for attributes (1, 2, 3, 4, 7, 8, 12, 13, 14, 15, 17, 22, 24, 25, 28, 29, 31, 33, 34, 39), all average accuracies exceed 80%. Table 7 presents statistics on the average accuracies of the three categories by different methods.

For category (I), the attributes on CelebA contain (1, 5, 6, 7, 10, 11, 14, 15, 16, 17, 18, 21, 23, 25, 27, 30, 31, 36, 39), while LFWA includes attributes (5, 6, 9, 10, 11, 16, 18, 19, 20, 21, 23, 27, 30, 32, 35, 36, 37, 38, 40). We can find that attributes (1, 7, 14, 15, 17, 25, 31, 39) in LFWA are not in category (I) in CelebA while that attributes (9, 19, 20, 32, 35, 37, 38, 40) in LFWA belong to category (III) in CelebA. For the above situation, the main reason is due to the insufficiency of the LFWA, resulting in MTCN not fully excavate the correlations among some attributes. We need to emphasize that attributes (9, 19, 20, 32, 35, 37, 38) are relatively difficult to estimate, which are not strongly affected by the insufficiency of LFWA.

From the above detailed analysis, we can conclude the following three points: (1) There is a strong correlation among face attributes, but the degrees of these correlations are different; (2) The size of the dataset will affect the model to learn the correlation. For example, when the dataset is sufficient, MTCN can well excavate the different degrees of correlations among

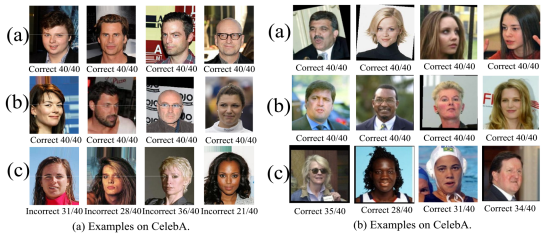

Fig. 7. Examples of 40 binary face attribute predictions. Rows (a) and (b) are good results and row (c) is poor estimation. "m/n" expresses (the number of correct predictions)/(40 face attributes) for each face.

Table 8. Results on Different Models and Datasets

Methods	Results on CelebA		Results on LFWA	
	Average accuracy	Computational time	Average accuracy	Computational time
Subnetworks without correlation excavating operation	91.21%	2.63 hours	86.04%	1.27 hours
Subnetwork with one correlation excavating operation	91.58%	5.61 hours	86.39%	2.31 hours
MTCN without TCCA	91.95%	7.33 hours	86.67%	3.08 hours
MTCN with TCCA	92.97%	11.5 hours	87.86%	5.5 hours

attributes; (3) When the dataset is small, the advantages of MTCN can be reflected. It is because of that advantage of MTCN, compared to the estimation performance on CelebA, those of most attributes on LFWA decrease slightly.

Examples of good and poor results by our MTCN are shown in Figure 7. Some poor results on CelebA are caused by the inconsistencies in the provided attributes and these on LFWA are due to too much volatility and interference. In our future work, we will tries to address those problems to boost the whole performance.

4.3.3 Ablation Analyses on the Correlations between Different Subnetworks. The MTCN enhances the original feature information of the attributes by excavating the correlations among different subnetworks. However, we do not know whether these operations work, whether the attribute feature enhancement comes from the correlation excavating of the subnetworks or the global correlation excavating from the TCCA. In addition, there are two correlation excavating operations between the S6 and C7 layers and between the N8 and C9 layers, and we still do not know which one is more important.

To figure out the above confusion, we show the test results (including the average accuracy and computational time) of the four different models on the CelebA and LFWA datasets, while the other experimental settings are unchanged. The four models are Subnetworks without correlation excavating operation, Subnetworks with one correlation excavating operation, MTCN without TCCA, and MTCN with TCCA. There is no correlation excavating operation between the C5 layer and the F11 layer of the first model, and that operation is used to excavate the correlation between the S6 and C7 layers in the second model. MTCN without TCCA not only excavates the correlations between the S6 and C7 layers but also excavates these between the N8 and C9 layers. The computational time includes training time and testing time.

As can be seen from the above Table 8, as the correlation excavating operations between subnetworks are gradually increasing, the final prediction performance continues to increase, and MTCN with TCCA achieves the highest accuracy. We can conclude that correlation excavating operations among subnetworks can enhance the original information of the face attributes, thus

Table 9. Results on Different Models and Datasets

Methods	Results on CelebA		Results on LFWA	
	Average accuracy	Computational time	Average accuracy	Computational time
MTCN without TCCA	91.95%	7.33 hours	86.67%	3.08 hours
Subnetworks with projection matrix	92.17%	8.14 hours	87.23%	3.59 hours
MTCN with TCCA	92.97%	11.5 hours	87.86%	5.5 hours

improving the final prediction performance of the model. At the same time, we can also find that as the model becomes more and more complex, the training and testing time of the model also increase, but most of the computational time is still spent in the process of model training.

4.3.4 Ablation Analyses on Tensor Canonical Correlation Analysis Scheme. TCCA is used to excavate the global correlation from the features of the whole C9 layers, but we still do not know whether the improvement of TCCA is brought by using features from subnetworks instead of TCCA's ability of using correlation of different tasks, and whether TCCA performs better than a projection matrix that is learnt by back propagation.

To address the above questions, we use a projection matrix instead of TCCA when the other settings are unchanged. At this time, since it is not necessary to consider the correlations among high-level features, it is only necessary to fully excavate the useful information among high-level features. For the convenience of calculation, we merge the fully-connected layer feature vectors of each subnetwork into a matrix that is mapped into a lower dimension matrix via projection matrix. The final predictions are made via *softmax*. The size of the projection matrix is 100×100, which is learnt by back propagation. This model is represented as MTCN with projection matrix.

We can learn from Table 9 that even if there are correlation excavating operations among the subnetworks, the accuracy of MTCN without TCCA on CelebA and LFWA is the worst among the three comparison algorithms. Correlation excavating operations among the subnetworks are still local operations that cannot fully exploit the effects of two or several attributes on other single or several attributes. Based on this model, Subnetworks with projection matrix model converts the fully connected layer feature vectors of all attributes into a 100×100 matrix via a projection matrix. Although this method does not fully consider the global correlation between multiple attributes, all attribute feature information is fused into a more comprehensive feature matrix through the projection matrix conversion operation. However, with TCCA, MTCN achieves the best performance among the comparison algorithms on CelebA and LFWA. The main reason is that MTCN can fully exploit the global correlations among all attribute features, especially the correlation of several attributes on several other attributes.

We can conclude that Subnetworks with projection matrix model uses features from subnetwork to enhance the original feature information of all attributes. Although there are two correlation excavating operations between subnetworks, the model only excavates correlations between multiple attributes and individual attribute. Therefore, the correlations among multiple attributes and other multiple attributes are not considered. Furthermore, there are not only positive correlations but also negative correlations among multiple attributes. With TCCA, MTCN can fully consider and utilize various correlations among attributes. Therefore, MTCN with TCCA achieves the best performance in all comparison algorithms.

4.3.5 Ablation Analyses on End to End Learning Scheme. MTCN does not use the end to end learning method, mainly because the TCCA conversion process is an approximate conversion process. It is also because of this process that a generalization matrix is used to keep the prediction

Table 10. Results on Different Models and Datasets

Methods	Results on CelebA		Results on LFWA	
	Average accuracy	Computational time	Average accuracy	Computational time
MTCN without TCCA	91.95%	7.33 hours	86.67%	3.08 hours
MTCN using end to end learning	92.37%	9.08 hours	87.23%	3.97 hours
MTCN with TCCA	92.97%	11.5 hours	87.86%	5.5 hours

results of all transformed features stable. When the MTCN adopts the end to end learning method, the loss of the whole system is jointly generated by the feature extraction process of the neural network, feature conversion process via TCCA, and the final predictions with generalization matrix. The total loss of an update process can be expressed as

$$\begin{aligned} Loss_{total} = Loss(f(Wx+b)) + Loss(TCCA) \\ + Loss(Generation\,Matrix), \end{aligned} \tag{32}$$

where $Loss_{total}$, $Loss(f(Wx+b))$, $Loss(TCCA)$, and $Loss(Generation\,Matrix)$ denote the whole loss, the loss of neural network, the loss of TCCA, and the loss of generation matrix, respectively.

Therefore, back propagation is used to update all the weights and biases. In such learning process, the entire system does not update the parameters of the TCCA, which means that the losses generated by the TCCA are all fed back to the previous neural network. Even if the system is trained, these losses generated by TCCA always exist, and they are eliminated via the front neural network. We present the losses of MTCN without TCCA as

$$Loss_{total} = Loss(f(Wx+b)). \tag{33}$$

We can draw a conclusion that the MTCN with end to end learning approach can excavate the correlation among face attributes, but back propagation only updates the neural network and generalization matrix while it does not update TCCA. Therefore, we suppose that the feature extraction ability of a single neural network may be stronger than that of MTCN with end to end learning, which we compare the performance of the two networks in terms of computational time and average accuracy.

As can be seen from the Table 10, the end to end learning method can accelerate the entire training and testing process, but the accuracies do not increase compared with those of MTCN, mainly because $Loss(TCCA)$ is not used to update TCCA during the update process. $Loss(f(Wx+b))$ and $Loss(TCCA)$ are all used to update the whole neural network, which results in the neural network part not only extracting useful information from the attributes but also extracting some information to compensate for the loss causing by TCCA conversion. When we do not adopt the end-to-end learning method, the losses generated by the TCCA are compensated by the generalization matrix. We present predictions of the front neural network model in the case of end-to-end learning: the prediction accuracies on the CelebA and LFWA datasets are 89.44% and 84.37%, respectively. The results are worse than those of MTCN without TCCA, which also prove the validity of MTCN. The front neural network model of MTCN firstly fully learns the face attribute information and the local correlations, and then the TCCA excavates and utilizes the attribute global correlations. Finally, the generalization matrix ensures that the whole prediction results have good robustness. These are the main reason why MTCN has achieved the best performance in all comparison algorithms.

5 CONCLUSIONS

This article proposes a novel MTCN algorithm to estimate human face attributes. MTCN consists of three parts: low-level feature sharing, high-level feature differentiation and correlation

excavation, and correlations of global features learning via TCCA and generalization matrix. MTCN first utilizes the multi-task learning model to excavate the correlations among different attributes, and then the TCCA algorithm is used to ensure that different attributes learn the different degrees of the correlations from other attributes. Finally, the generalization matrix to ensure that MTCN has better generalization performance. CelebA and LFWA datasets are utilized to validate the performance of MTCN, and our MTCN obtains the best performance compared with the latest multi-attribute recognition algorithms under the same settings.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Retrieved from https://arXiv:1603.04467.

[2] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. 2015. Multi-task CNN model for attribute prediction. *IEEE Trans. Multimedia* 17, 11 (Nov. 2015), 1949–1959. DOI: https://doi.org/10.1109/TMM.2015.2477680

[3] Jiajiong Cao, Yingming Li, and Zhongfei Zhang. 2018. Partially shared multi-task convolutional neural network with local constraint for face attribute learning. In *Proceedings of the CVPR*. 4290–4299.

[4] J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[5] Pierre Comon, Xavier Luciani, and André L. F. De Almeida. 2009. Tensor decompositions, alternating least squares and other tales. *J. Chemo.: J. Chemo. Soc.* 23, 7–8 (2009), 393–405.

[6] Cottrell, W Garrison, Metcalfe, and Janet. 1990. EMPATH: Face, emotion, and gender recognition using holons. In *Proceedings of the NIPs*. 564–571.

[7] A. Dantcheva and F. Bremond. 2017. Gender estimation based on smile-dynamics. *IEEE Trans. Info. Forensics Secur.* 12, 3 (Mar. 2017), 719–729. DOI: https://doi.org/10.1109/TIFS.2016.2632070

[8] Hamdi Dibeklioğlu, Fares Alnajar, Albert Ali Salah, and Theo Gevers. 2015. Combining facial dynamics with appearance for age estimation. *IEEE TIP* 24, 6 (2015), 1928–1943.

[9] Hui Ding, Hao Zhou, Shaohua Kevin Zhou, and Rama Chellappa. 2018. A deep cascade network for unaligned face attribute classification. In *Proceedings of the AAAI*.

[10] M. Duan, K. Li, and K. Li. 2018. An ensemble CNN2ELM for age estimation. *IEEE Trans. Info. Forensics Secur.* 13, 3 (Mar. 2018), 758–772. DOI: https://doi.org/10.1109/TIFS.2017.2766583

[11] Mingxing Duan, Kenli Li, Xiangke Liao, Keqin Li, and Qi Tian. 2019. Features-enhanced multi-attribute estimation with convolutional tensor correlation fusion network. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 3s (2019), 1–23.

[12] Max Ehrlich, Timothy J. Shields, Timur Almaev, and Mohamed R. Amer. 2016. Facial attributes classification using multi-task representation learning. In *Proceedings of the CVPR Workshops*. 47–55.

[13] Nour El Din Elmadany, Yifeng He, and Ling Guan. 2016. Multiview learning via deep discriminative canonical correlation analysis. In *Proceedings of the IEEE ICASSP*. 2409–2413.

[14] S. Fu, H. He, and Z. G. Hou. 2014. Learning race from face: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 12 (Dec. 2014), 2483–2509. DOI: https://doi.org/10.1109/TPAMI.2014.2321570

[15] Yun Fu, Guodong Guo, and Thomas S. Huang. 2010. Age synthesis and estimation via faces: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 11 (2010), 1955–1976.

[16] Lei Gao, Rui Zhang, Lin Qi, Enqing Chen, and Ling Guan. 2018. The labeled multiple canonical correlation analysis for information fusion. *IEEE Trans. Multimedia* 21, 2 (2018), 375–387.

[17] G. Guo and G. Mu. 2010. Human age estimation: What is the influence across race and gender? In *Proceedings of the CVPR Workshops*. 71–78. DOI: https://doi.org/10.1109/CVPRW.2010.5543609

[18] Guodong Guo and Guowang Mu. 2014. A framework for joint estimation of age, gender and ethnicity on a large database. *Image Vision Comput.* 32, 10 (2014), 761–770.

[19] Hu Han, Anil K. Jain, Fang Wang, Shiguang Shan, and Xilin Chen. 2018. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 11 (2018), 2597–2609.

[20] Emily M. Hand and Rama Chellappa. 2017. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *Proceedings of the AAAI*. 4068–4074.

[21] David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* 16, 12 (2004), 2639.

[22] Sandor Szedmak Hardoon, David R. and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* 16, 12 (2004), 2639–2664.

[23] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the CVPR*. 5375–5384.

[24] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2020. Deep imbalanced learning for face recognition and attribute prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 11 (2020), 2781–2794. DOI: 10.1109/TPAMI.2019.2914680

[25] Gary B. Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. Erik Learned-Miller, Andras Ferencz, and Frédéric Jurie, Marseille, France. (inria-00321923). https://hal.inria.fr/inria-00321923/file/Huang_long_eccv2008-lfw.pdf.

[26] Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, and Alexander Hauptmann. 2018. Gnas: A greedy neural architecture search method for multi-attribute learning. In *ACM Multimedia*. 2049–2057.

[27] S. J. Hwang, F. Sha, and K. Grauman. 2011. Sharing features between objects and their attributes. In *Proceedings of the CVPR*. 1761–1768. DOI: https://doi.org/10.1109/CVPR.2011.5995543

[28] Mahdi M. Kalayeh, Boqing Gong, and Mubarak Shah. 2017. Improving facial attribute prediction using semantic segmentation. In *Proceedings of the CVPR*. 6942–6950.

[29] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. 2007. Tensor canonical correlation analysis for action classification. In *Proceedings of the CVPR*. IEEE, 1–8.

[30] Pieter M. Kroonenberg and Jan De Leeuw. 1980. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika* 45, 1 (1980), 69–97.

[31] Neeraj Kumar, Peter Belhumeur, and Shree Nayar. 2008. FaceTracer: A search engine for large collections of images with faces. In *Proceedings of the ECCV*. 340–353.

[32] Young Ho Kwon and N. Da Vitoria Lobo. 1994. Age classification from facial images. In *Proceedings of the CVPR*. 762–767.

[33] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2006. On the best Rank-1 and Rank-(R1, R2, . . . , RN) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* 21, 4 (2006), 1324–1342.

[34] Gil Levi and Tal Hassncer. 2015. Age and gender classification using convolutional neural networks. In *Proceedings of the CVPR Workshops*. 34–42.

[35] Qiaozhe Li, Xin Zhao, Ran He, and Kaiqi Huang. 2019. Visual-semantic graph reasoning for pedestrian attribute recognition. In *Proceedings of the AAAI*, Vol. 33. 8634–8641.

[36] Zhifeng Li, Dihong Gong, Qiang Li, Dacheng Tao, and Xuelong Li. 2016. Mutual component analysis for heterogeneous face recognition. *ACM Trans. Intell. Syst. Technol.* 7, 3 (2016), 28.

[37] Giuseppe Lisanti, Svebor Karaman, and Iacopo Masi. 2017. Multi channel-Kernel Canonical Correlation Analysis for Cross-View Person Reidentification. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 2 (2017), 13.

[38] Fan Liu, Jinhui Tang, Yan Song, Liyan Zhang, and Zhenmin Tang. 2015. Local structure-based sparse representation for face recognition. *ACM Trans. Intell. Syst. Technol.* 7, 1 (2015), 2.

[39] Kuan Hsien Liu, Shuicheng Yan, and C. C. Jay Kuo. 2015. Age estimation via grouping and decision fusion. *IEEE Trans. Info. Forensics Secur.* 10, 11 (2015), 2408–2423.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the ICCV*. 3730–3738. DOI: https://doi.org/10.1109/ICCV.2015.425

[41] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. 2015. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE Trans. Knowl. Data Eng.* 27, 11 (2015), 3111–3124.

[42] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. 2015. Hierarchical convolutional features for visual tracking. In *Proceedings of the ICCV*. 3074–3082.

[43] S. Mehrkanoon and J. A. K. Suykens. 2018. Regularized semipaired kernel CCA for domain adaptation. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 7 (July 2018), 3199–3213. DOI: https://doi.org/10.1109/TNNLS.2017.2728719

[44] Venkatesh N. Murthy, Subhransu Maji, and R. Manmatha. 2015. Automatic image annotation using deep learning representations. *Proceedings of the 5th ICMR*. 603–606.

[45] X. Ning, W. Li, B. Tang, and H. He. 2018. BULDP: Biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition. *IEEE Trans. Image Process.* 27, 5 (May 2018), 2575–2586. DOI: https://doi.org/10.1109/TIP.2018.2806229

[46] G. J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang. 2012. Exploring context and content links in social media: A latent space method. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 5 (May 2012), 850–862. DOI: https://doi.org/10.1109/TPAMI.2011.191

[47] Guo Jun Qi, Xian Sheng Hua, and Hong Jiang Zhang. 2009. Learning semantic distance from community-tagged media collection. In *Proceedings of the ICME*. 243–252.

[48] R. Ranjan, V. M. Patel, and R. Chellappa. 2017. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* PP, 99 (2017), 1–1. DOI: https://doi.org/10.1109/TPAMI.2017.2781233

[49] Rasmus Rothe, Radu Timofte, and Luc Van Gool. 2018. Deep expectation of real and apparent age from a single image without facial landmarks. *Int. J. Comput. Vision* 126, 2–4 (2018), 144–157.
[50] Ethan M. Rudd, Manuel Günther, and Terrance E. Boult. 2016. Moon: A mixed objective optimization network for the recognition of facial attributes. In *Proceedings of the ECCV*. Springer, 19–35.
[51] C. O. Sakar and O. Kursun. 2017. Discriminative feature extraction by a neural implementation of canonical correlation analysis. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 1 (Jan 2017), 164–176. DOI: https://doi.org/10.1109/TNNLS.2015.2504724
[52] Richard Socher and Fei Fei Li. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proceedings of the CVPR*. 966–973.
[53] Zichang Tan, Jun Wan, Zhen Lei, Ruicong Zhi, Guodong Guo, and Stan Z. Li. 2017. Efficient group-n encoding and decoding for facial age estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 11 (2017), 2610–2623.
[54] Zichang Tan, Yang Yang, Wan Jun, Guodong Guo, and Stan Z. Li. 2020. Relation-aware pedestrian attribute recognition with graph convolutional networks. In *Proceedings of the AAAI*.
[55] Zichang Tan, Yang Yang, Jun Wan, Hanyuan Hang, Guodong Guo, and Stan Z. Li. 2019. Attention-based pedestrian attribute analysis. *IEEE Trans. Image Process.* 28, 12 (2019), 6126–6140.
[56] J. Tang, Y. Tian, P. Zhang, and X. Liu. 2018. Multiview privileged support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 8 (Aug. 2018), 3463–3477. DOI: https://doi.org/10.1109/TNNLS.2017.2728139
[57] Michal Uricar, Radu Timofte, Rasmus Rothe, Jiri Matas, and Luc Van Gool. 2016. Structured output SVM prediction of apparent age, gender and smile from deep features. In *Proceedings of the CVPR Workshops*. 730–738.
[58] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. 2002. Inferring a semantic representation of text via cross-language correlation analysis. In *Proceedings of the NIPs*. 1497–1504.
[59] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. [n.d.]. On deep multi-view representation learning. In *Proceedings of the ICML*. 1083–1092.
[60] Z. Wu, Q. Ke, J. Sun, and H. Y. Shum. 2011. Scalable face image retrieval with identity-based quantization and multireference reranking. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 10 (Oct. 2011), 1991–2001. DOI: https://doi.org/10.1109/TPAMI.2011.111
[61] Liping Xie, Dacheng Tao, and Haikun Wei. 2018. Early expression detection via online multi-instance learning with nonlinear extension. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 5 (2018), 1486–1496.
[62] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *Proceedings of the CVPR*. 3441–3450.
[63] Xinghao Yang, Weifeng Liu, Dapeng Tao, and Jun Cheng. 2017. Canonical correlation analysis networks for two-view image recognition. *Info. Sci. Int. J.* 385, C (2017), 338–352.
[64] Ting Yao, Tao Mei, and Chong Wah Ngo. 2015. Learning query and image similarities with ranking canonical correlation analysis. In *Proceedings of the ICCV*. 28–36.
[65] Dong Yi, Zhen Lei, and Stan Z. Li. 2014. Age estimation by multi-scale convolutional network. In *Proceedings of the ACCV*. 144–158.
[66] J. Yu, X. Yang, F. Gao, and D. Tao. 2017. Deep multimodal distance metric learning using click constraints for image ranking. *IEEE Trans. Cybernet.* 47, 12 (Dec 2017), 4014–4024. DOI: https://doi.org/10.1109/TCYB.2016.2591583
[67] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. 2019. Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 2 (2019), 661–674.
[68] Yang Zhong, Josephine Sullivan, and Haibo Li. 2016. Face attribute prediction using off-the-shelf CNN features. In *Proceedings of the IJCB*. 1–7.

Received March 2020; revised August 2020; accepted August 2020