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

Mingxing Duan[1,2], Kenli Li[1], Qi Tian[3],
[1] College of Information and Engineering, Hunan University, Changsha, China
[2] School of Computer Science, National University of Defense Technology, Changsha, China
[3] Department of Computer Science, University of Texas at San Antonio, USA
duanmingxing16@nudt.edu.cn, lkl@hnu.edu.cn, qi.tian@utsa.edu

ABSTRACT

Face multi-attribute prediction benefits substantially from multi-task learning (MTL), which learns multiple face attributes simultaneously to achieve shared or mutually related representations of different attributes. The most widely used MTL convolutional neural network is heuristically or empirically designed by sharing all of the convolutional layers and splitting at the fully connected layers for task-specific losses. However, it is improper to view all low and mid-level features for different attributes as being the same, especially when these attributes are only loosely related. In this paper, we propose a novel multi-attribute tensor correlation neural network (MTCN) for face attribute prediction. The structure shares the information in low-level features (*e.g.,* the first two convolutional layers) but splits that in high-level features (*e.g.,* from the third convolutional layer to the fully connected layer). At the same time, during high-level feature extraction, each subnetwork (*e.g.,* Age-Net, Gender-Net, ..., and Smile-Net) excavates closely related features from other networks to enhance its features. Then, we project the features of the C9 layers of the fine-tuned subnetworks into a highly correlated space by using a novel tensor correlation analysis algorithm (NTCCA). The final face attribute prediction is made based on the correlation matrix. Experimental results on benchmarks with multiple face attributes (CelebA and LFWA) show that the proposed approach has superior performance compared to state-of-the-art methods.

Keywords

Multi-task learning; Correlation; Tensor correlation analysis algorithm; Attribute prediction

1. INTRODUCTION

Human face attribute estimation has received a large amount of attention in visual recognition research because a face attribute provides a wide variety of salient information, such as a person's identity, age, race, gender, hair style, and clothing. Recently, many researchers have used face attributes in real-life applications, such as (i) identification, surveillance, and internet access control [8], [18], *e.g.,* automatic detection of juveniles on the Internet, or surveillance at unusual hours or in unusual places; (ii) face retrieval [25], [29], *e.g.,* automatic finding of person(s) of interest with provided attributes in a database; and (iii) social media [23], [24], *e.g.,* automatic recommendation of makeup or hair styles.

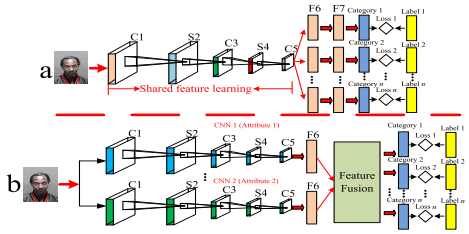

Figure 1: The methods used for face attribute prediction.

In spite of the recent progress in face attribute estimation [16], [6], [27], much of the prior work has been limited to predicting a single face attribute or learning a separate model to estimate each face attribute. As is known, face attributes are strongly related, such as goatee and male, heavy makeup and wearing lipstick and other relationships, and fully exploiting the correlation can help each task be learned better. A joint estimation of face attributes can address this embarrassing situation by exploring attribute correlations [19], [2], [7], [26], [11], and can achieve state-of-the-art performance for some face attribute predictions. These methods can be divided into two categories: multi-task learning (MTL) and multi-CNN fusion. Learning tasks in parallel while utilizing shared information to seek correlations is the main point of MTL, and in most work, as in Fig. 1 (a), there is shared representation from the first convolution layer to the last fully connected layer. However, [22] proved that it is not sensible to completely share representations between tasks, and these approaches ignore the differences and interactions among these attributes. In contrast, as illustrated in Fig. 1 (b), although the multi-CNN fusion method addresses the differences and explores the correlation through the output of the fully connected layer, it is difficult to realize end-to-end learning and neglect the correlation between the intermediate different attribute features.

Further, [12] proposed a multi-task deep convolutional

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

neural network for attribute prediction via sharing the lower layers in the CNN instead of all of its layers, and this process achieved better performance on many attributes. It follows that attributes and objects share a low-dimensional representation, which allows the object classifier to be regularized[15]. This approach does not fully explore the correlation among the high-level features of the face attributes, and each face attribute prediction should not only consider their difference but also utilize attribute correlation. Motivated by the analysis above, we propose a novel multi-task learning structure for face attributes that shares information in the lower-level feature layers and learns the differences and correlations among the high-level features.

At the same time, a large amount of work has proven that each face attribute estimation can be enhanced based on others, such as gender estimation based on smile dynamics [3], age estimation combined with smile dynamics [4], and age estimation affected by gender and race [10]. Although some face attribute predictions benefit from others, the degrees of influence on an attribute among other different attributes are not the same, and a unified correlation mechanism might not be appropriate. Consequently, a perfect face attribute should not only adequately seek the differences and correlations among the attributes but should also attempt to exploit the specific degrees of correlation among them. A novel tensor correlation analysis algorithm (NTCCA) is proposed to exploit the detailed correlations of the high-level features from the C9 layer of the fine-tuned subnetworks. A generalization matrix is utilized to ensure that each projected feature space is more highly correlated, which makes each face attribute fully exploit a maximal benefit from the others. Parts of the training dataset are used to train this matrix, and the experimental results indicate that this operation makes the whole system more stable and robust.

In this paper, a multi-task correlation learning neural network (MTCN) is proposed to predict face attributes. The system tries its best to capture the correlations among these attributes, which includes sharing information in low-level feature layers and splitting that in the high-level feature layers while extracting related information from other subnetworks to enhance its useful features and, finally, excavating the correlation among the C9 layers with a novel tensor correlation analysis algorithm (NTCCA). The detailed process of multi-task correlation learning is shown in Fig. 2. We first train the subnetwork with the corresponding attributes on CelebA or LFWA, and the fine-tuned MTCN is used to predict the attributes of CelebA or LFWA, which is our MTCN without NTCCA. Then, the features of the fine-tuned subnetworks for an image in the C9 layer are built into a tensor, and NTCCA is utilized to project the original features into the highly correlated feature space. Finally, CelebA and LFWA are used to verify the performance of the fine-tuned MTCN. The experimental results demonstrate that our approach significantly outperforms the state-of-the-art methods by achieving average accuracies of 92.97% and 87.96% on CelebA and LFWA, respectively.

2. RELATED WORK

2.1 Tensor Canonical Correlation Analysis

The n-mode product of $\mathcal{X}$ with the matrix $U \in R^{J_n \times I_n}$ is then denoted as $\mathcal{M} = \mathcal{X} \times_n U$, which is an $I_1 \times I_2 \cdots I_{n-1} \times J_n \times I_{n+1} \cdots \times I_N$ tensor with the element

$$M(i_1, ..., i_{n-1}, j_p, i_{n+1}, ..., i_N) = \sum_{i_n=1}^{I_n} \mathcal{X}(i_1, i_2, ..., i_N) U(j_n, i_n). \tag{1}$$

The product of $\mathcal{X}$ and a sequence of matrices $\{U_n \in R^{J_n \times I_n}\}_{n=1}^{N}$ is a $J_1 \times J_2 \times \cdots \times J_N$ tensor denoted by

$$\mathcal{M} = \mathcal{X} \times_1 U_1 \times_2 U_2 \cdots \times_N U_N. \tag{2}$$

The CANDECOMP / PARAFAC (CP) decomposition 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, $a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)}$, which are *rank one* tensors, and can be denoted as

$$\begin{aligned} \mathcal{X} &\cong \sum_{r=1}^{R} \lambda_r a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)} \\ &= \Lambda \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)} \end{aligned} \tag{3}$$

Given m views $\{X_p\}_{p=1}^{m}$ of samples, in which $X_p = \{\mathrm{x_{p1}}, \mathrm{x_{p2}}, ..., \mathrm{x_{pN}}\} \in R^{d_p \times N}$, the variance matrices are $C_{pp} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{x}_{pn}\mathrm{x}_{pn}^T$, $p = 1, 2, ..., m$. Then, the covariance tensor among all of views is calculated as

$$\mathcal{C}_{1,2,...,m} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \mathrm{x}_{1n} \circ \mathrm{x}_{2n} \circ ... \circ \mathrm{x}_{mn} \tag{4}$$

where $\mathcal{C}$ is a tensor with $d_1 \times d_2 \times ... \times d_m$. According to the traditional two-view CCA [13], exploration is performed to maximize the correlation among the canonical variables $\mathrm{z}_p = X_p^T \mathrm{h}_p$, $p = 1, 2, ..., m$, in which $\{\mathrm{h}_p \in R^{d_p \times 1}\}_{p=1}^{m}$ denotes the canonical vectors. Therefore, the optimization problem is

$$\begin{aligned} \arg\max_{\{\mathrm{h}_p\}} &= \mathrm{corr}(\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_m) \\ s.t.\ & z_p^T z_p = 1,\ p = 1,\ ...,\ m \end{aligned}, \tag{5}$$

Here, $\mathrm{corr}(\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_m) = (\mathrm{z}_1 \odot \mathrm{z}_2 \odot ..., \odot \mathrm{z}_m)^T \mathrm{e}$ expresses the canonical correlation, where $\bigodot$ denotes the element-wise product, and $\mathrm{e} \in R^N$. According to TCCA, the optimization problem (5) is equivalent to

$$\begin{aligned} \arg\max_{\{\mathrm{h}_p\}} &\ \rho = \mathcal{C}_{1,2,\ ...,\ m} \bar{\times}_1 \mathrm{h}_1^T \bar{\times}_2 \mathrm{h}_2^T ... \bar{\times}_m \mathrm{h}_m^T \\ s.t.\ & \mathrm{h}_p^T C_{pp} \mathrm{h}_p = 1,\ p = 1, 2,\ ...,\ m \end{aligned}, \tag{6}$$

where $\bar{\times}_p$ denotes the p-mode contracted tensor-vector product. Let $\mathrm{u}_p = \widetilde{C}_{pp}^{1/2} \mathrm{h}$ and $\mathcal{M} = \mathcal{C}_{1,2,\ ...,\ m} \bar{\times}_1 \widetilde{C}_{11}^{1/2} \mathrm{h} \bar{\times}_2 \widetilde{C}_{22}^{1/2} \mathrm{h}$ $... \bar{\times}_m \widetilde{C}_{mm}^{1/2} \mathrm{h}$. Then, the optimization problem in (6) is described as

$$\begin{aligned} \arg\max_{\{\mathrm{h}_p\}} &\ \rho = \mathcal{M} \bar{\times}_1 \mathrm{u}_1^T \bar{\times}_2 \mathrm{u}_2^T ... \bar{\times}_m \mathrm{u}_m^T \\ s.t.\ & \mathrm{u}_p^T \mathrm{u}_p = 1,\ p = 1, 2,\ ...,\ m \end{aligned}, \tag{7}$$

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

According to [17], Equation (7) is equivalent to seeking the best rank-1 approximation of the tensor $\mathcal{M}$, *i.e.*, the best *rank one* CP decomposition of the tensor $\mathcal{M}$. This construct denoted as

$$\mathcal{M} \approx \sum_{k=1}^{r} \rho_k \mathrm{u}_1^{(k)} \circ \mathrm{u}_2^{(k)} \circ ... \circ \mathrm{u}_p^{(k)}, \tag{8}$$

The alternating least squares (ALS) algorithm is used to approximately seek the solutions. Letting $U_p = [u_p^{(1)}, ..., u_p^{(r)}]$, the projected data for the p'th view can be calculated as

$$Z_p = X_p^T \widetilde{C}_{pp}^{-1/2} U_p. \quad (9)$$

The different $Z_{p_{p=1}}^{m}$ are concatenated as the final representation $Z \in R^{(mr)\times N}$ for the subsequent learning.

3. PROPOSED METHOD

3.1 Low-level Feature Sharing for Face Attributes

The convolutional layers of a typical CNN model provide multiple levels of abstraction in the feature hierarchies [21]. The features in the earlier layers retain higher spatial resolution for precise localization with low-level visual information. Because max pooling is used in the CNNs, the spatial resolution is gradually reduced with an increase in network depth. Therefore, the features in high-level layers capture more semantic information and fewer fine-grained spatial details. The face attributes (*e.g.,* lips, nose, hair) keep more semantic information than spatial resolution; in other words, the high-level features extracted from a face image are beneficial for face attribute prediction. Hence, for face multi-attribute prediction, the low-level features can be shared. According to [21] and [12], because the first and second convolutional layers retain higher spatial resolution with low-level visual information, our MTCN shares low-level features from the input to the second convolutional layer. Fig. 2 shows a full schematic diagram of our network architecture.

3.2 Differentiation and Correlation in High-level Layers

From the third convolutional layer, we split the network into multi-subnetworks. This arrangement is chosen because different CNNs trained by different targets can be considered different feature descriptors, and the features learned from them can be seen as different views/representations of the data. These subnetworks have the same network structure and aim to predict different face attributes.

At the same time, based on [3], [4], [10], and [6], each of the face attribute estimations can be enhanced based on the other attributes, and each of our subnetworks seeks useful information from the other networks in the same layers to enhance itself. This operation appears twice in the C7 and C9 layers because these layers have more semantic information.

In the first stage, as shown in Fig. 3, the convolutional neural network is trained on datasets. In this situation, the whole structure is an end-to-end learning network. During the process of feed-forward processing, the low-level features are shared until the third convolutional layer and split at the high-level layers for task-specific losses.

Due to the specificity of the MTCN, backpropagation is a crucial step, and the gradients transferred from the output to C9, C9 to C7, and C5 to C3 are difficult to compute. We present the detailed derivations and the implementation in the following subsections. First, we use the cross-entropy loss function for the subnetworks, and the loss is

$$C = -\frac{1}{N}\sum_{i=1}^{N}(y_i \ln p_i + (1-y_i)(1-\ln p_i)). \quad (10)$$

where p_i denotes the probability of an attribute produced by our proposed network. We use y_i to denote the ground-truth of the attribute and N to denote the number of training examples.

3.2.1 *Gradients Transferred from the C9 layer to the N8 layer*

Our MTCN has two specific feature extraction stages, in which the convolutional layer extracts features from both its own network and from the same level layer of other subnetworks. For this reason, the operations in the whole subnetworks are the same in this stage, and we present only the detailed gradient transferred in Gender-Net. K is the number of subnetworks. We assume that the weights and biases between C9 and the fully connected layer are $\mathbf{w}_{cf}$ and $\mathbf{b}_{cf}$ and that those between the N8 layer and C9 layer are $\mathbf{w}_{nc}$ and $\mathbf{b}_{nc}$. Here, X_c^i and X_n^i express the output of the convolutional and normalization layers of the ith sample. Although the C9 layer of Gender-Net extracts features from multiple subnetworks, we do not design other convolutional kernels for these feature maps. For example, (X_1^i, X_2^i, ..., X_K^i) denotes the corresponding feature maps of the K subnetworks, and the outputs of the C9 layer of Gender-Net are

$$X_c^i = f(X_1^i \mathbf{w}_{nc} + X_2^i \mathbf{w}_{nc}, ..., + X_K^i \mathbf{w}_{nc} + \mathbf{b}_{nc}). \quad (11)$$

Let us calculate the partial derivative of the cross-entropy cost with respect to the weights and biases. By applying the chain rule, we obtain

$$\frac{\partial C}{\partial w_{nc}} = \frac{\partial C}{\partial X_c^i}\frac{\partial X_c^i}{\partial w_{nc}}, \quad (12)$$

$$\frac{\partial C}{\partial b_{nc}} = \frac{\partial C}{\partial X_c^i}\frac{\partial X_c^i}{\partial b_{nc}}, \quad (13)$$

while

$$\frac{\partial C}{\partial X_c^i} = -\frac{1}{N}\sum_{i=1}^{N}\frac{(y - f(w_{cf}, X_c^i, b_{cf}))f'(w_{cf}, X_c^i, b_{cf})w_{cf}}{f(w_{cf}, {X_c}^i, b_{cf})(1 - f(w_{cf}, {X_c}^i, b_{cf}))}, \quad (14)$$

We use the definition of the ReLU function, $max(0, x)$, and then $f'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}$, where $x = \sum_i w_{cf} X_c^i + b_{cf}$. Thus,

$$\frac{\partial C}{\partial X_c^i} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} w_{cf}\frac{(y - f(w_{cf}, X_c^i, b_{cf}))}{f(w_{cf}, X_c^i, b_{cf})(f(w_{cf}, X_c^i, b_{cf}) - 1)} & x > 0 \\ 0 & x \le 0 \end{cases}, \quad (15)$$

and according to equation (11),

$$\frac{\partial X_c^i}{\partial w_{nc}} = f'(w_{nc}, X_n^i, b_{nc})\sum_{j=1}^{K} X_j^i = \begin{cases} \sum_{j=1}^{K} X_j^i & x > 0 \\ 0 & x \le 0 \end{cases}, \quad (16)$$

$$\frac{\partial X_c^i}{\partial b_{nc}} = f'(w_{nc}, X_n^i, b_{nc}) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}, \quad (17)$$

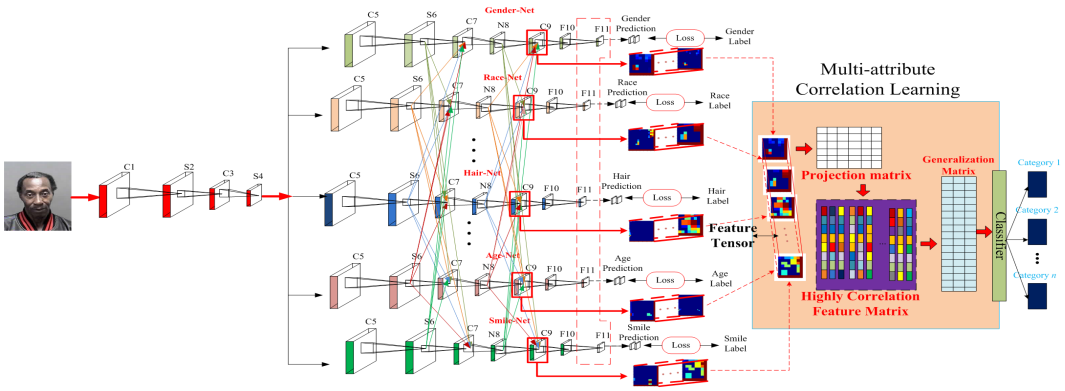

Figure 2: Full schematic diagram of our network architecture. (C1, C3, ..., C9) denote the corresponding convolutional layers, (S2, S4, S6) represent pooling and normalization operations, N8 signifies only the normalization operation, and (F10 and F11) express the fully connected layers. The structure shares the information from C1 to S4 but splits that in high-level features (e.g., from the third convolutional layer to the fully connected layer). The feature maps of the C9 layers of the fine-tuned subnetworks are built into a feature tensor, and the tensor is projected into a highly correlated space via NTCCA, based on which the finial predictions are made.

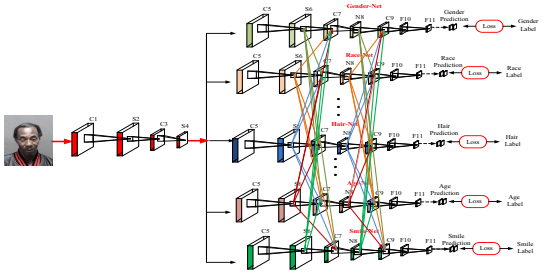

Figure 3: The learning process of the subnetworks.

Therefore,

$$\frac{\partial C}{\partial w_{nc}} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} \frac{\left(y-f(w_{cf},X_c^i,b_c)\right)w_{cf}X_j^i}{f(w_{nc},X_n^i,b_{nc})\left(f(w_{nc},X_n^i,b_{nc})-1\right)} & x>0 \\ 0 & x\le 0 \end{cases}, \tag{18}$$

$$\frac{\partial C}{\partial b_{nc}} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{\left(y-f(w_{cf},X_c^i,b_c)\right)w_{cf}}{f(w_{nc},X_n^i,b_{nc})\left(f(w_{nc},X_n^i,b_{nc})-1\right)} & x>0 \\ 0 & x\le 0 \end{cases}, \tag{19}$$

and we can update the weights and biases in this layer as follows:

$$w_{nc} = w_{nc} + \eta\frac{\partial C}{\partial w_{nc}}, \tag{20}$$

$$b_{nc} = b_{nc} + \eta\frac{\partial C}{\partial b_{nc}}. \tag{21}$$

where η is the learning rate.

3.2.2 *Gradients Transferred from the N8 layer to the S6 layer*

The partial derivative of the cross-entropy cost with respect to the weights and biases from the C7 layer to the S6 layer is the same as that from the C9 layer to the N8 layer.

It is important to consider how to transfer the gradients from the N8 layer to the C7 layer in Gender-Net because the C9 layer extracts features from multiple subnetworks; these features affect the gradients simultaneously because Gender-Net is a full subnetwork. In this time, $\mathbf{w}_{sc}^{'}$ and $\mathbf{b}_{sc}^{'}$ signify the weights and biases between the S6 layer and the C7 layer, respectively, and $X_c^{i'}$ denotes the output of the C7 layer. We apply the chain rule twice to compute the partial derivative as follows:

$$\frac{\partial C}{\partial w_{sc}^{'}} = \frac{\partial C}{\partial X_c^i}\frac{\partial X_c^i}{\partial X_c^{i'}}\frac{\partial X_c^{i'}}{\partial w_{sc}^{'}}, \tag{22}$$

$$\frac{\partial C}{\partial b_{sc}^{'}} = \frac{\partial C}{\partial X_c^i}\frac{\partial X_c^i}{\partial X_c^{i'}}\frac{\partial X_c^{i'}}{\partial b_{sc}^{'}}, \tag{23}$$

due to $X_c^i = f(X_c^{i'}w_{sc} + \ldots + b_{sc})$ and $f'(x) = \begin{cases} 1 & x>0 \\ 0 & x\le 0 \end{cases}$, where $x = \sum_i w_{sc}X_s^i + b_{sc}$. We can find

$$\frac{\partial X_c^i}{\partial X_c^{i'}} = f'\left(w_{sc},X_s^i,b_{sc}\right)w_{sc} = \begin{cases} w_{sc} & x>0 \\ 0 & x\le 0 \end{cases}, \tag{24}$$

and

$$\frac{\partial X_c^{i'}}{\partial w_{sc}^{'}} = f'(w_{sc}^{'},X_s^{i'},b_{sc}^{'})\sum_{j=1}^{K'}X_j^{i'} = \begin{cases} \sum_{j=1}^{K'}X_j^{i'} & x>0 \\ 0 & x\le 0 \end{cases}, \tag{25}$$

$$\frac{\partial X_c^{i'}}{\partial b_{sc}^{'}} = f'(w_{sc}^{'},X_s^{i'},b_{sc}^{'}) = \begin{cases} 1 & x>0 \\ 0 & x\le 0 \end{cases}, \tag{26}$$

Therefore,

$$\frac{\partial C}{\partial w_{sc}} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K'} \frac{\left(y-f(w_{cf},X_c^i,b_c)\right)w_{cf}w_{sc}X_j^{i'}}{f(w_{sc},X_s^i,b_{sc})\left(f(w_{sc},X_s^i,b_{sc})-1\right)} & x>0 \\ 0 & x\le 0 \end{cases}, \tag{27}$$

$$\frac{\partial C}{\partial b_{sc}^{'}} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{\left(y-f(w_{cf},X_c^i,b_c)\right)w_{cf}w_{sc}}{f(w_{sc},X_s^i,b_{sc})\left(f(w_{sc},X_s^i,b_{sc})-1\right)} & x>0 \\ 0 & x\leq 0 \end{cases}, \tag{28}$$

Then, the weights and biases between the S6 layer and the C7 layer can be updated as

$$w_{sc}^{'} = w_{sc}^{'} + \eta\frac{\partial C}{\partial w_{sc}^{'}}, \tag{29}$$

$$b_{sc}^{'} = b_{sc}^{'} + \eta\frac{\partial C}{\partial b_{sc}^{'}}. \tag{30}$$

3.2.3 *Gradients Transferred from Subnetworks to a Shared Single Network*

The weights and biases between the S4 and C5 layers can be updated based on the corresponding subnetworks. How to transfer the gradients from the subnetworks to a single network is another crucial problem. Due to the distinctiveness of our MTCN, we adopt a joint gradient transfer strategy to compute the gradients. C_1, C_2, ..., C_K denote the cross-entropy losses of the whole subnetworks. Additionally, $\mathbf{w}_{sc}^{''}$ and $\mathbf{b}_{sc}^{''}$ express the weights and biases between the S4 layer and the C5 layer, and $\mathbf{w}_{sc}^{'''}$ and $\mathbf{b}_{sc}^{'''}$ signify the weights and biases between the S2 layer and the C3 layer. The joint gradient transferred strategy is

$$\Delta w = \frac{\partial C_1}{\partial w_{sc}^{'''}} + \frac{\partial C_2}{\partial w_{sc}^{'''}} + \ ... \ + \frac{\partial C_K}{\partial w_{sc}^{'''}}, \tag{31}$$

$$\Delta b = \frac{\partial C_1}{\partial b_{sc}^{'''}} + \frac{\partial C_2}{\partial b_{sc}^{'''}} + \ ... \ + \frac{\partial C_K}{\partial b_{sc}^{'''}}, \tag{32}$$

where $\frac{\partial C_1}{\partial w_{sc}^{'''}}$, ..., $\frac{\partial C_K}{\partial w_{sc}^{'''}}$ and $\frac{\partial C_1}{\partial b_{sc}^{'''}}$, ..., $\frac{\partial C_K}{\partial b_{sc}^{'''}}$ can be calculated by the chain rule based on the calculation of the partial derivatives above.

Therefore, we can update the weights and biases between the S2 layer and the C3 layer as

$$w_{sc}^{'''} = w_{sc}^{'''} + \eta\Delta w, \tag{33}$$

$$b_{sc}^{'''} = b_{sc}^{'''} + \eta\Delta b. \tag{34}$$

3.3 Multi-attribute Tensor Correlation Learning Framework

In the first stage, the subnetworks not only consider the differences among them but also extract the related information to enhance themselves. Although this novel design can achieve better performance than can most of the compared methods, we do not fully consider the specific degrees of correlation among the face attributes. Hence, based on the fine-tuned network, we want to further excavate the detailed correlation information, so a novel TCCA approach (called NTCCA) is proposed to explore the detailed correlations among the high-level features of these subnetworks. Unlike TCCA, which aims to directly maximize the correlation between the canonical variables of all views [20], our proposed NTCCA maximizes the correlation of all of the feature maps in C9 for an image.

To explore the correlation among the different types of features in C9 for an image, we consider $X_l^i = \{\{X_1^1, X_2^1, ..., X_L^1\}, \{X_1^2, X_2^2, ..., X_L^2\}, ..., \{X_1^K, X_2^K, ..., X_L^K\}\}$, $l = 1, 2, ..., L$ and $i = 1, 2, ..., K$. The size of the feature map in C9 is $\kappa\times\kappa$, and X_l^i composes a 3-D tensor, $\mathcal{X} \in R^{\kappa\times\kappa\times KL}$ where KL denotes the whole feature maps. Based on $\mathcal{X}$, we redefine the feature map as $\{X_p\}_{p=1}^{KL}$ and $X_p = \{x_{p1}, x_{p2}, ..., x_{p\kappa}\} \in R^{\kappa\times\kappa}$. The variance matrices can be denoted as $C_{dd} = \frac{1}{\kappa}\sum_{j=1}^{k} x_{pj}x_{pj}^T$, and the covariance tensor among X_1, X_2, ..., X_{KL} is calculated as

$$\mathcal{C}_{1,2,...,(KL)} = \frac{1}{\kappa}\sum_{j=1}^{\kappa} x_{1j}\circ x_{2j}\circ\cdots\circ x_{(KL)j}, \tag{35}$$

where $\mathcal{C}$ is a tensor of dimension $\kappa\times\kappa\times\cdots\times\kappa$ and $\circ$ expresses the outer product.

Without loss of generality, we first obtain the canonical correlation as Equation (36), where the canonical variables $z_p = X_p^T h_p$.

$$\begin{aligned} &\arg\max \rho = corr(z_1, z_2, \ \cdots, \ z_{KL}) \\ &s.t.\ z_p^T z_p = 1, \ p = 1, \ 2, \ \cdots, \ (KL), \end{aligned} \tag{36}$$

According to TCCA, Equation (36) is equivalent to $C_{1,2,\ ...,\ (KL)} \bar{\times}_1 h_1^T \bar{\times}_2 h_2^T \bar{\times} \ \cdots \ \bar{\times}_{(KL)} h_{(KL)}^T$, and the correlation can be further calculated as

$$\begin{aligned} &\arg\max \rho = C_{1,2,\ ...,\ (KL)} \bar{\times}_1 h_1^T \bar{\times} \ \cdots \ \bar{\times}_{(KL)} h_{(KL)}^T \\ &s.t.\ h_p^T C_{pp} h_p = 1, \end{aligned} \tag{37}$$

where $X_p X_p^T = C_{pp}$ and $\bar{\times}_p$ denote the p-mode contracted tensor-vector product.

According to Equations (7) and (8), the alternating least squares (ALS) algorithm is used to seek approximate solutions. Letting $U_p = [u_p^{(1)}, ..., u_p^{(r)}]$, the projected data for the p'th view can be calculated as

$$Z_p = X_p^T \widetilde{C}_{pp}^{-1/2} U_p. \tag{38}$$

Then, we concatenate the different $\{Z_p\}_{p=1}^{(KL)}$ as the final representation $Z \in R^{(KLr)\times\kappa}$. Because the method presented above is only used to calculate the correlation of multiple attributes of an image, a generalization matrix is utilized to ensure that the projected results exhibit more stabilization and higher correlation. Parts of the training dataset are used to train the matrix through algorithm 1. Our goal is to estimate multiple attributes for an image; thus, a joint attribute estimation model is utilized to calculate the loss of the whole system. For a face image with M attributes, a joint attribute estimation model can be formulated as follows:

$$\epsilon = \arg \ min \sum_{i=1}^{M} C_i + \gamma\Phi(W_j). \tag{39}$$

where C_i expresses the cross-entropy loss of the ith attribute, $\Phi(\cdot)$ denotes a regularization term to penalize the complexity of the weights, and $\gamma > 0$ is a regularization parameter.

During this process, the neural network is not updated, and we only update W and b. Algorithm 1 is as follows:

CelebA and LFWA datasets are used in our experiments [19] and they are divided into training dataset, validation dataset, and testing dataset. Till now, our MTCN has been fine-tuned and the training process are roughly as follows:
Step 1: Train MTCN without NTCCA on the training datasets with the corresponding attributes and a fine-tuned MTCN can be used to make predictions;
Step 2: Train the generalization matrix with NTCCA on

Algorithm 1 Novel Tensor Canonical Correlation Analysis

Require:
The training set: N face images;
Iterations t and max iterations t_{max};
Error ϵ and minimum error e_{min}; Learn rate η.

Ensure:
Output layer weight and bias: W and b.

Initialize the output layer weight and bias: W and b;
for i in range (N)
  Map KL attribute feature space into another space according to Equation (38);
  $\{X_1, X_2, ..., X_{(KL)}\} \rightarrow \{Z_1, Z_2, ..., Z_{(KL)}\}$;
  Calculate muli-attribute output: $y = Z \cdot W + b$;
  Calculate final total loss ϵ according to Equation (39);
  if $(\epsilon \leq e_{min})$ or $(t \geq t_{max})$
    Use the fine-tuned model to predict the multi-attribute tasks;
  else Compute the modified weight coefficient:
    $\Delta w = \eta \frac{\partial \epsilon}{\partial w}$;
    Compute the modified biases coefficient:
    $\Delta b = \eta \frac{\partial \kappa}{\partial b}$;
    Update output weights $W = W + \Delta w$;
    Update output biases $b = b + \Delta b$;
  end if
end for
Return updated output layer weight W and bias b.

one third of the training datasets;
Step 3: Verify the performance of the fine-tuned MTCN on the testing datasets.

4. EXPERIMENTS

4.1 Datasets

4.1.1 CelebA

CelebA is a large-scale face attribute database [19]; it contains 10K identities, and each identity has 20 images. Each image has 40 attributes (see Table 1), such as gender, race, and smiling, which makes it challenging for face attribute prediction. The dataset contains 200,000 images: 160,000 are used for training, 20,000 for validation, and 20,000 for testing. Because the CelebA dataset is so large, we do not need to augment it in any way.

4.1.2 LFWA

LFWA is another unconstrained face attribute database [19], and its face images are from the LFW database [14]. It has 40 attributes, which have the same annotations as in the CelebA database. The LFWA dataset consists of 13,143 images, of which, 6,263 were used for training, 2,800 for validation, and 4,080 for testing. If we did not augment the training dataset, then the network would have severely overfit the dataset because of the large number of parameters. We follow the data augmentation scheme presented in [12] and we have over 75,000 images for training.

4.2 Implementation Details

Our proposed structure is implemented using the publicly available Tensorflow [1] code. The entire network in this paper is trained using an NVIDIA Tesla P100. First, we resize the input image to 256 × 256 pixels, and then, a 224 × 224 crop is selected from the center of the image or the four corners from the entire processed image. We also adopt different dropout measures to limit the risk of overfitting. The network is initialized with random weights following a Gaussian distribution; the mean is 0, and the standard deviation is 0.01. A base learning rate of 10^{-4} is used, and it is reduced by 10% every 100,000 iterations. To train the MTCN, we use batches of size 100, and we train both datasets for 30 epochs. Overall, training with NTCCA takes approximately 10 hours for the CelebA dataset and approximately 4 hours for the LFWA dataset, and nearly 1.5 hours is required to calculate the generalization matrix W. Each experiment is conducted four times, and we obtain the average of the relevant results. Because codes of the baseline methods used in subsequent sections are not available in the public domain, we directly report the results in the corresponding publications.

Table 1: Summary of the 40 face attributes provided in the CelebA dataset.

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

Table 2: Subnetwork 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
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
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
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

4.2.1 Network Structure

The neural network of the MTCN consists of two parts: the shared network and 40 subnetworks. The 40 subnetworks have the same network layers, such as convolutional layers, contrast normalization layer, pooling layer, ReLU nonlinear function, and identical network parameters. The detailed subnetwork configurations are shown in Table 2. The convolutional layer is followed by ReLU, which is a max pooling and a local response normalization layer. Every F10 layer has 4098 units and is followed by a ReLU and 50% dropout to avoid overfitting. Each F11 layer is fully connected to a corresponding F10 layer, which also has 4098

units, and it is also followed by ReLU and a 50% dropout. The final fully connected layer connects F11 with 1000 units.

4.3 Results

The results obtained for CelebA and LFWA by the proposed approach and several state-of-the-art approaches [19], [28], [12], [11], [9], and [5] are presented in Table 3. The MTCN with NTCCA outperforms [19], [28], [12], [11], [9], and [5] for most of the 40 face attributes in both the CelebA and LFWA. For the CelebA results, in terms of the average accuracies, our MTCN with NTCCA improves on [19] by 5.67%, on [28] by 2.03%, on [12] by 1.68%, on [11] by 0.37%, on [9] by 1.96%, on [9] (unaligned) by 2.65%, and on [5] by 1.74%. For the LFWA results, our MTCN with NTCCA improves on [19] by 4.11%, on [12] by 1.65%, and on [11] by 1.81%. Although our MTCN achieves better performance among these compared methods, we do not know how much of an effect our MTCN with NTCCA has on the performance of the whole or some attribute predictions and whether our MTCN has worked on the related face attributes. Therefore, we conduct a further analysis based on Table 3 in the following sections.

4.3.1 Ablation Analyses on the CelebA Dataset

We do not expect to see an increase in performance with MTCN for every attribute because some attributes do not have strong relationships with others, but most attributes achieved better estimations compared to the state-of-the-art methods. From the prediction presented in Table 3, these attributes can be divided into three major categories based on the results of our method: **I**) attributes (♯ 1, 5, 6, 7, 10, 11, 14, 15, 16, 17, 18, 21, 23, 25, 27, 30, 31, 36, 39) that are relatively easy to predict using our MTCN; most of the results exceed 95%, but those achieved using the compared methods are lower than 95%. Each of these attributes is correlated with one or more other attributes, and our MTCN excavates these correlations in different levels, which is one of the most important reasons that it can obtain the best performance of all; for example, ♯ 25 ($NoBeard$) relates to {♯ 2 ($ArchedEyebrows$), 3($BushyEyebrows$), 6($Bald$), 7($Bangs$), 10($BrownHair$), 11($GrayHair$), 12($BigLips$), 18($Goatee$), 19($HeavyMakeup$), 20($HighCheekbones$), 22($Male$))}; **II**) the estimation of attributes (♯ 26 and 28) is less than 80%; they are easily influenced by the shooting angle and pose, and few of the attributes are highly related; and **III**) these attributes are related to the attributes in **I**. Most of the time, the attributes in **III** can enhance the features of the attributes in **I**, while those of **III** benefit less from those of **I**. For example, {♯ 2 ($ArchedEyebrows$) and ♯ 25 ($NoBeard$)}, {♯ 3 ($BushyEyebrows$) and ♯ 25 ($NoBeard$)}, {♯ 20 ($HighCheekbones$) and (♯ 25 ($NoBeard$), 32 ($Smiling$))}.

Table 4 presents the average accuracies of the methods for the three categories. For category **I**, the average accuracies of our MTCN without NTCCA are 96.46%, which improves on [19] by 13.04%, on [28] by 1%, on [12] by 0.83%, on [9] by 0.99%, on [9] (unaligned) by 1.32%, and on [5] by 0.83%. With NTCCA, MTCN improves the average accuracy by 1.12% compared to that without NTCCA, and it shows better performance than does [11]. Then, for category **II**, for the average accuracies of [19], [28], [12], [11], [9], [9] (unaligned), [5], our MTCN without NTCCA, and our MTCN with NTCCA are 77%, 76.1%, 76.66%, 78%, 75.31%, 75.56%, 76.19%, 77.08%, and 78.49%, respectively. We find that our MTCN with NTCCA achieves the best performance. Finally, for category **III**, the average accuracy of MTCN with NTCCA is 89.88%, which improves the average accuracy by 0.99% compared to MTCN without NTCCA, and it exceeds all of the compared methods listed above.

Based on the analysis above, we can learn that if the face attribute relates to the others and MTCN is trained with a large enough dataset, the proposed method can show good performance via excavating the correlations among these attributes, such as the performance on categories **I** and **III**. Without NTCCA, the performance of our MTCN is nearly the same as that of the state-of-the-art methods, mostly because of the novel design of the network, which not only fully considers the differences among the face attributes but also extracts related information to enhance itself. Further, it attempts to maximize the correlation among the high-level features through the NTCCA. Compared with the state-of-the-art methods on CelebA, our method not only improves the average accuracy of the attributes taken as a while but also greatly increases the poor accuracies of single attributes predicted by the compared methods; for example, the predictions for the attribute $Bangs$ are nearly 72%, while that of our MTCN is 95.44%.

4.3.2 Ablation Analyses on the LFWA Dataset

Compared with CelebA, LFWA is a relatively small dataset, so all of the average accuracies are lower than those on CelebA. Although our MTCN achieves the best performance of all of the compared algorithms, the trends in the accuracies of some of the attribute predictions are not the same as those in CelebA. For example, the accuracy of $Bangs$ (♯ 7) on LFWA is 84.51%, and it belongs to category **II**, but $Bangs$ is in category **I** on CelebA, and $BlondHair$ (♯ 9) is in category **I** on LFWA but belongs to category **II** on CelebA. Although LFWA is a small dataset, the accuracies of most of the attributes decrease slightly compared with those on CelebA. The augmentation scheme on LFWA is an important reason, but a more important reason is attributed to the novel structure of considering the correlations of the attributes in different levels.

Without loss of generality, we still divide these attributes into three categories according to the results of our MTCN on LFWA. Comparing the three categories with those on CelebA, we can learn that our MTCN is effective in the case of a small number of images. The details are as follows: **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 results exceed 90%, but those of the compared methods are lower than 90%; **II**) the estimation of attribute (♯ 26) is less 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 results exceed 80%. Table 5 shows the detailed average accuracies of the three categories.

In terms of the attributes in category **I**, those on CelebA include attributes (♯ 1, 5, 6, 7, 10, 11, 14, 15, 16, 17, 18, 21, 23, 25, 27, 30, 31, 36, 39), while LFWA has attributes (♯ 5, 6, 9, 10, 11, 16, 18, 19, 20, 21, 23, 27, 30, 32, 35, 36, 37, 38, 40). This result indicates that attributes (♯ 1, 7, 14, 15, 17, 25, 31, 39) do not belong to category **I** from CelebA in LFWA but that attributes (♯ 9, 19, 20, 32, 35, 37, 38, 40) are in category **I**, which belongs to **III** in CelebA. The size of the dataset affects the predictions of attributes (♯ 1, 7, 14, 15, 17, 25, 27, 31, 39), but our MTCN minimizes that

Table 3: Attribute estimation accuracies (in %) for the 40 binary attributes (see Table 2) on the CelebA and LFWA databases by the proposed approach and the state-of-the-art methods [19], [28], [12], [11], [9], and [5]. The average accuracies of [19], [28], [12], [11], [9], [9](unaligned), [5], and the proposed approach are 87.30%, 90.94%, 91.29%, 92.60%, 91.01%, 90.32%, 91.23%, 91.95%(Ours), and 92.97%(Ours), respectively, on CelebA, and the average accuracies of [19], [12], [11], and the proposed approach are 83.85%, 86.31%, 86.15%, 86.81%(Ours), and 87.96%(Ours), respectively, on LFWA. The highest accuracy for each attribute is in bold.

	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 [19]	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 [28]	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 [12]	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 [11]	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
	AFFACT [9]	94.21	82.12	82.83	83.75	99.06	96.05	70.88	83.82	90.32	96.07	95.50	89.16	92.41	94.41	96.18	99.61	97.31	98.28	91.10	86.88
	AFFACT Unaligned [9]	94.09	81.27	80.36	84.89	97.82	95.49	71.42	81.83	85.88	95.17	94.52	87.72	90.59	95.10	95.94	99.38	97.21	97.89	90.82	86.11
	PaW [5]	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
	MTCN without NTCCA	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 NTCCA	**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 [19]	**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 [12]	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 [11]	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
	MTCN without NTCCA	80.59	85.14	82.35	83.78	92.01	92.78	80.64	84.51	92.17	97.28	87.97	80.91	83.00	79.01	80.24	91.67	85.58	88.74	95.72	88.63
	MTCN with NTCCA	81.68	**86.23**	**83.01**	**84.33**	**92.16**	**93.44**	**84.51**	**85.17**	**93.20**	**98.09**	**89.47**	**81.83**	84.52	**83.27**	**82.00**	92.84	**87.12**	**89.81**	**96.41**	**89.75**

	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 [19]	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 [28]	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 [12]	98.17	93.74	96.88	87.23	96.05	75.84	97.05	77.47	93.81	95.16	97.85	92.73	83.58	83.91	90.43	99.05	94.11	86.63	96.51	88.48
	DMTL [11]	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
	AFFACT [9]	98.26	92.60	96.89	87.23	95.99	75.79	97.04	74.83	93.29	94.45	97.83	91.77	84.10	85.65	90.20	99.02	91.69	87.85	96.90	88.66
	AFFACT Unaligned [9]	97.29	92.82	96.89	87.15	95.33	74.87	96.97	76.24	91.74	94.54	97.46	90.45	82.17	83.37	90.33	98.66	92.99	87.55	96.43	86.21
	PaW [5]	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
	MTCN without NTCCA	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 NTCCA	**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 [19]	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 [12]	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 [11]	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
	MTCN without NTCCA	93.68	85.64	94.31	82.49	82.00	77.58	92.47	84.09	85.84	87.13	82.71	91.80	79.03	81.00	95.00	91.49	94.68	90.73	81.06	86.84
	MTCN with NTCCA	**94.21**	**86.73**	**95.67**	**83.51**	**82.43**	**78.85**	**93.68**	**84.93**	**87.00**	**88.39**	**84.11**	**92.77**	**80.00**	81.45	**95.73**	**92.38**	**95.69**	**91.75**	**82.00**	**88.04**

Table 4: The average accuracies of the three categories on CelebA.

Methods	Category I	Category II	Category III
LENet+ANet[19]	83.42%	77%	85%
MOON[28]	95.46%	76.1%	87.99%
MCNN+AUX[12]	95.47%	75.31%	88.18%
DMTL[11]	95.14%	75.56%	87.06%
AFFACT[9]	95.63%	76.19%	88.41%
AFFACT Unaligned[9] (unaligned)	95.63%	76.66%	88.5%
PaW[5]	97.31%	78%	89.42%
MTCN without NTCCA	96.46%	77.08%	89.01%
MTCN with NTCCA	97.58%	78.49%	89.88%

Table 5: The average accuracies of the three categories on LFWA.

Methods	Category I	Category II	Category III
LENet+ANet[19]	89.17%	74%	79.76%
MCNN+AUX[12]	91.38%	77.39%	82.39%
DMTL[11]	91.67%	75%	81.95%
MTCN without NTCCA	91.81%	77.58%	82.96%
MTCN with NTCCA	92.77%	78.85%	84.26%

influence by making full use of the correlations among the attributes. Additionally, to the advantage of our system, the predictions of attributes (♯ 9, 19, 20, 32, 35, 37, 38), which are relatively difficult to predict, are not strongly affected by the size of the dataset.

In conclusion, face attributes are related, and the degrees of correlation among the different attributes are different. Excavating the related information at different levels can improve the performance of the attribute predictions. MTCN attempts to capture the correlation from different levels of features among the different attributes, such as sharing information in low-level feature layers and splitting it in the high-level feature layers while extracting related information from other subnetworks to enhance its own useful features and excavating the correlations of high-level features. These are the main reasons why our MTCN can achieve better performance on a relatively small dataset, even if it is used without NTCCA, and why the overall performance of our system on LFWA is close to that for the same attribute on CelebA. Because of its novel design compared with other methods, MTCN not only achieves the best performance, but also greatly improves the accuracies of the predictions on some single attributes, such as *Bangs* and *Blurry*.

5. CONCLUSIONS

This paper presents a novel multi-task tensor correlation neural network (MTCN) for facial attribute prediction. Compared to the existing approaches, the proposed method fully explores the correlations at different levels, including sharing information in the low-level feature layers, splitting that

in the high-level feature layers while extracting related information from other subnetworks to enhance its features and excavating the correlation of high-level features with NTCCA. Then, our MTCN makes final decisions for each attribute prediction. Extensive experiments demonstrate the effectiveness of our proposed system. The experimental results show that fully exploiting the correlations among the face attributes can achieve better performance, even if the training dataset is not large enough. In the future, we will improve the hybrid systems to achieve better prediction performance for the attributes in categories **II** and **III**.

6. REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia*, 17(11):1949–1959, Nov 2015.

[3] A. Dantcheva and F. BrÍęmond. Gender estimation based on smile-dynamics. *IEEE Transactions on Information Forensics and Security*, 12(3):719–729, March 2017.

[4] H. Dibeklioglu, F. Alnajar, A. A. Salah, and T. Gevers. Combining facial dynamics with appearance for age estimation. *IEEE Transactions on Image Processing*, 24(6):1928–1943, June 2015.

[5] H. Ding, H. Zhou, S. K. Zhou, and R. Chellappa. A deep cascade network for unaligned face attribute classification. *arXiv preprint arXiv:1709.03851*, 2017.

[6] M. Duan, K. Li, and K. Li. An ensemble CNN2ELM for age estimation. *IEEE Transactions on Information Forensics and Security*, 13(3):758–772, March 2018.

[7] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer. Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55, 2016.

[8] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010.

[9] M. Günther, A. Rozsa, and T. E. Boult. Affact-alignment free facial attribute classification technique. *arXiv preprint arXiv:1611.06158*, 2016.

[10] G. Guo and G. Mu. Human age estimation: What is the influence across race and gender? In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 71–78, June 2010.

[11] H. Han, K. J. A, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[12] E. M. Hand and R. Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI*, pages 4068–4074, 2017.

[13] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639, 2004.

[14] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. *Month*, 2007.

[15] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *CVPR 2011*, pages 1761–1768, June 2011.

[16] Y. H. Kwon and N. D. Vitoria Lobo. Age classification from facial images. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 762–767, 1994.

[17] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank-(r 1 , r 2 ,. . ., r n) approximation of higher-order tensors. *Siam Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2006.

[18] K. H. Liu, S. Yan, and C. C. J. Kuo. Age estimation via grouping and decision fusion. *IEEE Transactions on Information Forensics and Security*, 10(11):2408–2423, 2015.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, Dec 2015.

[20] 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*, 27(11):3111–3124, 2015.

[21] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[22] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[23] G. J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang. Exploring context and content links in social media: A latent space method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):850–862, May 2012.

[24] G. J. Qi, X. S. Hua, and H. J. Zhang. Learning semantic distance from community-tagged media collection. In *International Conference on Multimedia 2009, Vancouver, British Columbia, Canada, October*, pages 243–252, 2009.

[25] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *CoRR*, abs/1603.01249, 2016.

[26] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[27] R. Rothe, R. Timofte, and L. V. Gool. Deep

expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, pages 1–14, 2016.

[28] E. M. Rudd, M. Günther, and T. E. Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conference on Computer Vision*, pages 19–35. Springer, 2016.

[29] Z. Wu, Q. Ke, J. Sun, and H. Y. Shum. Scalable face image retrieval with identity-based quantization and multireference reranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1991–2001, Oct 2011.