Deep Relative Attributes

Xiaoshan Yang, Tianzhu Zhang, Member, IEEE, Changsheng Xu, Fellow, IEEE, Shuicheng Yan, Senior Member, IEEE, M. Shamim Hossain, Senior Member, IEEE, Ahmed Ghoneim, Member, IEEE.

Abstract—Relative attribute learning aims to learn the ranking function describing the relative strength of the attribute. Most of current learning approaches learn linear ranking function for each attribute by use of the hand-crafted visual features. Different from the existing work, in this paper, we propose a novel deep relative attributes (DRA) algorithm to learn visual features and the effective nonlinear ranking function to describe the relative attribute of image pairs in a unified framework. Here, visual features and the ranking function are learned jointly, and they can benefit each other. The proposed DRA model is comprised of 5 convolutional neural layers, 5 fully connected layers, and a relative loss function which contains the contrastive constraint and the similar constraint corresponding to the ordered image pairs and the un-ordered image pairs, respectively. To train the DRA model effectively, we make use of the transferred knowledge from the large scale visual recognition on ImageNet [1] to the relative attribute learning task. We evaluate the proposed DRA model on three widely used datasets. Extensive experimental results demonstrate that the proposed DRA model consistently and significantly outperforms the state-of-the-art relative attribute learning methods. On the public OSR, PubFig and Shoes datasets, compared with the previous relative attribute learning results [2], the average ranking accuracies have been significantly improved by about 8%, 9%, and 14%, respectively.

Index Terms—Relative attributes, Deep learning.

I. INTRODUCTION

Visual attributes are intrinsic properties in images with human-designed names (e.g., ‘natural’, ‘smiling’), and they are valuable as higher semantic cues than low level visual features in many interesting scenarios. For example, researchers have shown that visual attributes are valuable for facial verification [3], object recognition [4], [5], image retrieval/search [6], [7], [8], [9], [10], [11], [12], [13], [14], video retrieval and recommendation [15], [16], generating descriptions of unfamiliar objects [17] and transfer learning [18], [19], [20], [21]. Many attributes mining and learning methods have also been proposed [22], [23], [24]. In these methods, the attributes are binary, which indicates the presence (or absence) of a certain property in an image. Compared with the binary attributes, using relative attributes is a much richer way for humans to describe objects semantically with relative visual properties. The consecutive relative values of the attributes can reflect not only whether the attribute appears in an image, but also the strength of the attribute. As a richer language of visual description than the commonly used binary attributes, relative attribute learning has gained much attention and can be used in many applications especially social event analysis [25], [26], [27], [28], and zero-shot learning [2], [29], [30], [31].

Most of the existing relative attribute learning algorithms are based on the ranking SVM framework [2], [29], [30] to learn a ranking function for each attribute. Here, the value of the ranking score denotes the strength of the attribute in an image with respect to other images. Despite remarkable progress in this field, there exists significant room for improvement, especially in the following three aspects: (1) Existing relative attribute methods rely on traditional hand-crafted features, such as gist descriptor [2], [31] and color histogram [2], [31], which may not optimally capture the most appropriate visual features to describe relative attributes. (2) Most of the relative attribute learning methods [2], [29], [30], [31] only learn a linear or shallow ranking function to obtain the relative score of image pair for a specific attribute. The linear or shallow models are simple, and may not best represent the mapping from visual features of images pair to the relative score of attributes. (3) Existing relative attribute learning methods [2], [29], [30], [31] perform feature extraction and ranking function learning separately, which cannot capture the most useful features for describing visual attributes of images.

To deal with the above issues, we propose a novel deep relative attributes (DRA) algorithm to learn visual features and the more effective nonlinear ranking function to describe the relative attribute of image pair in a unified framework. In this paper, with the same pipeline as in [2], we learn the ranking function for each relative attribute independently. As shown in Figure [1], the proposed DRA model is comprised of 5 convolutional neural layers, 5 fully connected layers, and a relative loss function. The convolutional neural layers are adopted to learn middle level visual features for attribute representation, and the fully connected layers are adopted to learn a nonlinear ranking function to map the learned visual features by the convolutional neural layers to the relative score of a specific attribute. The relative loss function contains the contrastive constraint of the ordered image pairs and the similar constraint of the un-ordered image pair. As a result,
the relative loss function can make the output of the last fully connected layer reflect the relative score of the attribute. In our DRA model, the visual features and the ranking function are learned jointly in a unified convolutional neural network framework, and they can benefit each other. More effective visual features can improve the ranking accuracy of the relative attribute, while a better ranking function can be used to guide the more appropriate visual features learning. In the proposed DRA model, there are million-scale parameters, such as the convolutional kernels in the convolutional layer, and the weights and the bias in the fully connected layer, which require large scale labeled data for training. However, the existing largest public dataset for relative attribute learning only contains about 10 thousand-scale labeled images. To overcome this issue, we make use of the transferred knowledge from the large scale visual recognition on ImageNet [1]. Thus, we adopt the trained image classification model [32], [33] to initialize the low level layers of the proposed DRA model. Then, the proposed DRA model is trained on the relative attributes dataset with the labeled image pairs.

The contributions of the proposed DRA are four-fold:

- To the best of our knowledge, the proposed DRA model is the first work to learn relative attributes directly using CNNs, though there are deep CNN based methods for binary attributes.
- Compared with conventional hand-crafted features and linear ranking SVM based relative attribute learning, we adopt convolutional neural networks to learn more effective nonlinear functions and map the original images to obtain their relative strength values of the attribute.
- In the proposed DRA model, the visual features and effective nonlinear ranking functions are learned jointly in a unified framework to benefit each other.
- Extensive experimental results demonstrate that the proposed DRA model consistently and significantly outperforms state-of-the-art relative attribute learning methods on three challenging benchmarks. On the public OSR, PubFig and Shoes datasets, compared with the previous relative attribute learning methods [2], the average ranking accuracies have been significantly improved by about 8%, 9%, and 14%, respectively.

The rest of this paper is organized as follows. In Section II, we summarize the related work. Our method and the optimization are introduced in Section III. Experimental results are reported and analyzed in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

In this section, we review the related work about binary attributes, relative attributes, and deep learning which are the three most related topics to the proposed method.

**Binary attributes:** Visual attribute learning allows prediction of color or texture types [34], and can also help obtain a mid-level cue for object or face recognition [35], [3], [5]. Attributes can also facilitate zero-shot learning [35], [4], [36] and part localization [17], [37], [38]. To avoid defining attribute vocabularies manually, some methods aim to explore attribute-related concepts on the Web [39], [40], extract them from existing knowledge sources [41], [6] or discover them interactively [41]. There are also some methods proposed for attribute mining. Zhang et al. [42] propose to automatically discover attribute from an arbitrary set of image and text pairs. To detect generic facial attribute by leveraging visual and contextual cues, Chen et al. propose to automatically acquire training images from publicly available community-contributed photos in an unsupervised manner [43]. To detect various facial attributes such as gender, age and more which consume more computation and storage resources, Lin et al.
propose a compression framework to find fewer significant latent human topics to approximate more facial attributes [44]. In contrast to the relative attribute learning approaches, all such methods restrict the attribute to be binary without considering the relative information through attributes.

**Relative attributes:** Using relative attributes is a semantically rich way to describe and compare objects in the world, and more powerful than existing binary attributes in uniquely identifying an image. In [2], relative attributes are first proposed based on the learning to rank framework, whose ranking function is learned for each attribute to denote the relative values or ranking scores. The ranking functions of all attributes are learned independently in [2], and ignore the correlations among multiple attributes. To improve this method, the multi-task learning is introduced in [31] to learn ranking functions of multiple attributes jointly. Relative attributes are also used for many other applications. In [30], the active learning framework is adopted to include feedback on not only the label but also the attributes. In [29], as human-interpretable mid-level visual concepts, the relative attributes are used for a supervisor to provide feedback to the classifier. In [45], a relative attribute feedback strategy is adopted for image search. Here, the ranking functions of attributes are learned iteratively according to the user feedback to make the images with top ranking scores close to the user’s preference. These existing relative attributes methods are based on hand-crafted features to learn linear functions to map these features to the relative scores of the corresponding attributes. Different from these methods, the proposed deep relative attributes can learn image features and more effective nonlinear functions for attributes in a unified framework.

**Deep learning:** In recent years, deep models including deep belief networks (DBNs) [46], deep Boltzmann machines (DBMs) [47], stacked auto-encoders (SAEs) [48], [49] and convolutional neural networks (CNNs) [50], [52] have drawn much attention due to their encouraging performances. As effective feature learning methods, the deep models have been widely used in many applications, such as large scale object recognition [32], [1], [51], [52], [53], [54], human action recognition [55], face point detection [56], and social event analysis [57]. The most relevant methods to the proposed model are deep metric learning for face verification [58], [59] and deep ranking for fine-grained image similarity learning [60]. In the deep metric learning methods [58], [59], the trained Siamese networks aim to predict whether the two input images are the same person or not. The input of the networks is always an image pair. In contrast, the top fully connected layer of the proposed DRA maps the features of two images to two continuous strength values of the attribute. The great value denotes the strong strength of the attribute in the image while small value denotes weak strength. In the deep metric learning methods, the difference of two images is measured by Euclidean distance [58] and absolute difference [59], respectively. In contrast, we adopt the direct difference among two output values. Thus after training, the different output values for two images are able to not only represent whether two images are similar, but also give their strength order with regard to the attribute. In the test phase, the previous deep metric learning methods can only predict whether two images belong to the same person or not. Thus, the input must be an image pair. In contrast, the learned DRA model can predict the strength value of any individual image. The input is a single image. Moreover, the inputs to the networks [58] are hand-crafted low-level visual features including DSIFT, LBP and SSIFT. The deep ranking [60] learns a ranking function by a triplet-based network architecture, where each network is a combination of the convolutional neural networks and two low-resolution paths to extract low resolution visual features. Different from this method, the proposed deep relative attributes algorithm adopts a single CNN. In the forward propagation, different inputs and outputs of the two images are computed by the same parameters in each layer of the CNN. Though the single CNN model is adopted, the two images are propagated forward through the convolutional and fully connected layers, separately.

### III. The proposed Deep Relative Attributes

In this section, we firstly show the problem description of deep relative attribute learning. Then, we introduce the deep network structure of the proposed model and the detail of each layer. At last, we illustrate the forward and backward propagation schemes to optimize the proposed model.

#### A. Problem Description

The goal of relative attribute learning is to learn a ranking function for each attribute with a number of human labeled ordered or un-ordered image pairs. Given a test image, the score of the ranking function can be used to denote the strength of each attribute in the image [2]. In this paper, we focus on learning the ranking functions for the relative attributes independently. For simplicity, we adopt \( f(x) \) to denote the ranking function corresponding to a specific attribute \( a \). For the attribute \( a \), we use \( P \) to denote a set of ordered image pairs and \( Q \) a set of un-ordered image pairs. If image pair \((x_i, y_i) \in P\), it means that the image \( x_i \) has a higher relative value of attribute \( a \) than image \( y_i \). If image pair \((x_i, y_i) \in Q\), the image \( x_i \) and image \( y_i \) have similar relative values of attribute \( a \). With these notations, the relative attribute learning for the attribute \( a \) can be formulated as learning \( f(x) \) by satisfying the following constraints:

\[
\forall (x_i, y_i) \in P, \quad f(x_i) > f(y_i) \quad (1)
\]

\[
\forall (x_i, y_i) \in Q, \quad f(x_i) = f(y_i) \quad (2)
\]

In the traditional attribute learning methods [2], [31], the hand-crafted features are adopted, and the learned ranking function \( f(x) \) is linear. Different from these methods, our aim is to learn visual features and nonlinear ranking function jointly in a unified framework to benefit each other under convolutional neural networks. The details are introduced in the next subsection.
B. Deep Network Structure

To achieve the above goal, we propose a novel deep relative attribute learning model as shown in Figure 1. Here, we show the training and testing process of the DRA model with image pairs for the attribute natural. The DRA model contains 5 convolutional layers (Conv1, Conv2, Conv3, Conv4, Conv5) and 5 fully connected layers (FC6, FC7, FC8, FC9, FC10). Different from the traditional CNNs [32], in the training phase, the input to the DRA model is an image pair \((x, y)\) with relative attribute assignment \(l\) which denotes the label of the image pair. The \(l = 1\) means that image \(x\) has larger attribute value than image \(y\) (ordered image pair) while the \(l = 0\) means the two images have similar attribute values (un-ordered image pair). In the forward propagation, different inputs and outputs of the two images are computed by the same parameters in each layer. Though the same CNN model is adopted, the two images are propagated forward through the convolutional and fully connected layers, separatively. The outputs \(F_{10}^{l}\) and \(F_{10}^{q}\) in the last fully connected layer denote the relative values of the corresponding input images \(x\) and \(y\) with regard to the attribute natural. Following the fully connected layer, a relative loss function is adopted to constrain the relative output values of the image pair. By the contrastive constraint \(\max(0, \tau - (F_{10}^{l} - F_{10}^{q}))\) and the similar constraint \(\frac{1}{\tau} (F_{10}^{l} - F_{10}^{q})^2\) for the natural attribute in the loss function, the ordered image pair \((l = 1)\) will be constrained to have the discrepant outputs while the un-ordered image pair \((l = 0)\) will be constrained to have the same or very close outputs. In the test phase, the single CNN with the learned parameters is used to predict the strength value of any individual image with regard to the natural attribute. It is worth noting that, as shown in Figure 1 for each convolutional layer, we show the size and number of the convolutional filters. For each fully connected layer, we show the dimension of the output feature vector. We do not show the pooling, normalization and dropout layers after the convolutional layers or the fully connected layer. For all the convolutional layers, we adopt the same manner of pooling or normalization as AlexNet [32], [33]. The dropout is only carried out after the \(F_6\) and \(F_7\) layers.

**Convolutional Layers.** For the \(m^{th}\) convolution layer, we denote its output as \(h_m(x) = s(W_m \ast h_{m-1}(x) + b_m)\), \(m \in \{1, \ldots, 5\}\). Here, \(\ast\) denotes the convolutional operation, \(W_m\) and \(b_m\) are the convolutional kernel and bias. \(s(x) = \max(0, x)\) denotes the non-saturating nonlinearity activation function which is also used as the rectified linear units (ReLU) in [32].

**Fully Connected Layers.** For the \(m^{th}\) fully connected layer, we denote its output as \(h_m(x) = s(W_m h_{m-1}(x) + b_m)\), \(m \in \{6, \ldots, 10\}\). Here, \(W_m\) and \(b_m\) are the weight matrix and bias, respectively. For the activation function, the same rectified linear units \(s(x) = \max(0, x)\) as in the convolutional layer is adopted.

**Relative Loss Function.** For a specific attribute \(a\), the loss function for training the ranking function \(f(x)\) is defined as the sum of the contrastive constraint, the similar constraint and the regularization item:

\[
L = \frac{1}{2|\mathcal{G}|} \sum_{(x, y) \in \mathcal{G}} \left[ l_i L_p(x_i, y_i) + (1 - l_i) L_q(x_i, y_i) \right] + \lambda ||\Theta||_2^2. \tag{3}
\]

Here, the \(\mathcal{G} = \mathcal{P} \cup \mathcal{Q}\) contains all ordered and un-ordered image pairs annotated for a specific attribute. \(l_i\) denotes the label of the \(i^{th}\) image pair. \(l_i = 1\) means that image \(x_i\) has larger attribute values than image \(y_i\) (ordered image pair) while \(l_i = 0\) means the two images have similar attribute values (un-ordered image pair). \(L_p(x_i, y_i) = \max(0, \tau - (f(x_i) - f(y_i)))\) and \(L_q(x_i, y_i) = (f(x_i) - f(y_i))^2\) denote the contrastive constraint for the ordered image pairs and the similar constraint for the un-ordered image pairs respectively. For the image \(x_i\) and the image \(y_i\), \(f(x_i)\) and \(f(y_i)\) denote the one dimensional output attribute strength values \(F_{10}^{l}\) and \(F_{10}^{q}\) in Figure 1 at the top fully connected layer. \(\Theta\) contains all parameters of the proposed DRA model including the convolution kernels in the convolutional layers, the transformation matrices in the fully connected layers and the biases. In the training phase, the relative loss is used to learn the parameters of the CNN model. For the image \(x_i\) and the image \(y_i\), the learned CNN model can output values \(f(x_i)\) and \(f(y_i)\) which have the same ranking order with the labeled order of the two images with regard to the attribute. The \(\tau\) controls the relative margin among the attribute values of ordered image pair. During training, \((f(x_i) - f(y_i))\) can be larger or smaller than \(\tau\) but will be constrained to be no less than \(\tau\). The \(\lambda\) is used to control the regularization item.

We give more detailed explanations of the loss function in two cases as follows. (1) If \(l_i = 0\), which means image \(x_i\) has the same attribute value with image \(y_i\), the contrastive constraint will be zero while the minimization of the similar constraint \((f(x_i) - f(y_i))^2\) will make \(f(x_i)\) and \(f(y_i)\) have the same value. (2) If \(l_i = 1\), which means image \(x_i\) has greater attribute value than image \(y_i\), the similar constraint will be zero while the minimization of the contrastive constraint \(\max(0, \tau - (f(x_i) - f(y_i)))\) will make \(f(x_i)\) have greater value than \(f(y_i)\). For this case, we illustrate it with two subcases. (a) If \(f(x_i) \geq f(y_i) + \tau\), the loss will be zero which is just what we want. Thus the minimization will do nothing and have no any penalty. (b) If \(f(x_i) < f(y_i) + \tau\), the loss will be a positive value \(\tau - (f(x_i) - f(y_i))\). Thus the minimization will make it close to zero until \(f(x_i) > f(y_i) + \tau\).

C. Optimization

The optimization of the proposed DRA model is similar to the conventional neural networks, where the stochastic gradient descent is adopted. The kernels in the convolutional layers and the weights in the fully connected layers are updated through the forward and backward propagations. In the forward propagation of the training phase, given two input images, different outputs of the two images in each layer are computed by the same operation. Though the same CNN model is adopted, the two images are propagated forward through the convolutional and fully connected layers separately. In the backward propagation, the gradients with
parameters in each layer are calculated based on the gradients with regard to the outputs in each layer are calculated for the two images separately. Then, the gradients with regard to the parameters in each layer are calculated based on the gradients of the outputs for two images.

(1) Forward propagation. In the forward propagation, for an image pair \((x_i, y_i)\), the image \(x_i\) will be propagated through the convolutional layers and the fully connected layers: \(x \to C_1 \to C_2 \to C_3 \to C_4 \to C_5 \to F_6 \to F_7 \to F_8 \to F_9 \to F_{10}\). Meanwhile, the image \(y_i\) will also be propagated through the same convolutional layers and the same fully connected layers: \(y \to C'_1 \to C'_2 \to C'_3 \to C'_4 \to C'_5 \to F'_6 \to F'_7 \to F'_8 \to F'_9 \to F'_{10}\). The outputs \(F'_10\) and \(F'_{10}\) of the last fully connected layer denote the relative strength values of the two images with regard to the attribute. Then the relative loss of the networks is computed based on the outputs of the fully connected layers \(F_{10}\) and \(F'_{10}\).

(2) Backward propagation. In the backward propagation, the partial gradients of the loss function in Eq. (3) are firstly computed with regard to the outputs \(f(x_i) = F_{10}\) and \(f(y_i) = F'_{10}\) of the last fully connected layer. Then the errors computed in the loss function will be propagated backward to the remaining fully connected layers and the convolutional layers, the parameters \(\Theta\) including the weights \(\{W_m\}_{m=1}^{M}\) and biases \(\{b_m\}_{m=1}^{M}\) of the fully connected layers and the kernels in the convolutional layers will be updated. In a specific layer, for an input image pair \((x_i, y_i)\), once the gradients of the outputs for image \(x_i\) and \(y_i\) are separately computed, the gradients of the same parameters will be computed based on them. Then the parameters will be updated according to their partial gradients. More details of the partial gradients in the relative loss function and the fully connected layers are listed as follows. The partial gradients in the convolutional layers are computed in the similar way as the conventional convolutional neural networks.

Partial gradients in the relative loss function. We use \(f(x)\) to denote the output of the last fully connected layer \(F_{10}\) for the attribute. If we denote \(d_i = f(x_i) - f(y_i)\) and \(\text{sign}(\cdot)\) as a binary sign function, the partial gradients of the loss function in Eq. (3) with regard to \(f(x)\) can be computed as:

\[
\frac{\partial L}{\partial f(x)} = \begin{cases} 
  \frac{1}{|Q|} d_i, & \text{if} \ (x_i, y_i) \in Q \\
  -\frac{1}{|P|} \text{sign}(\tau - d_i), & \text{if} \ (x_i, y_i) \in P
\end{cases}
\]

Partial gradients in the fully connected layer. For simplicity, we denote \(f(x) = h_M(x) = s(W_M x + b_M)\) as the output of the last fully connected layer. The partial gradients of the loss function in Eq. (3) with regard to the weights \(W_M\) of the \(M^{th}\) fully connected layer \((F_{10})\) can be computed as:

\[
\frac{\partial L}{\partial W_M} = \sum_{(i,j) \in P} \left( \frac{\partial L(x_i)}{\partial W_M} + \frac{\partial L(y_i)}{\partial W_M} \right) + \lambda W_M 
\]

The partial gradients with regard to the bias is:

\[
\frac{\partial L}{\partial b_M} = \sum_{(i,j) \in P} \left( \frac{\partial L(x_i)}{\partial b_M} + \frac{\partial L(y_i)}{\partial b_M} \right).
\]

Here, \(\frac{\partial L(x)}{\partial W_M}\) denotes the contribution of image \(x\) to the partial gradients of the whole loss with regard to \(W_M\), and \(\frac{\partial L(x)}{\partial W_M}\) denotes the contribution of image \(y\).

\[
\frac{\partial L(x_i)}{\partial W_M} = \begin{cases} 
  \frac{1}{|Q|} d_i, & \text{if} \ (x_i, y_i) \in Q \\
  -\frac{1}{|P|} \text{sign}(\tau - d_i), & \text{if} \ (x_i, y_i) \in P
\end{cases}
\]

\[
\frac{\partial L(y_i)}{\partial W_M} = \begin{cases} 
  -\frac{1}{|Q|} d_i, & \text{if} \ (x_i, y_i) \in Q \\
  \frac{1}{|P|} \text{sign}(\tau - d_i), & \text{if} \ (x_i, y_i) \in P
\end{cases}
\]

Here, the partial gradients of \(h_M(x)\) are computed as:

\[
\frac{\partial h_M(x)}{\partial W_M} = s'(W_M h_{M-1}(x) + b_M)(h_{M-1}(x))^T.
\]

Similarly, the contributions to the gradients with regard to \(b_M\) of images \(x_i\) and \(y_i\) can be computed as follows:

\[
\frac{\partial L(x_i)}{\partial b_M} = \begin{cases} 
  \frac{1}{|Q|} d_i, & \text{if} \ (x_i, y_i) \in Q \\
  -\frac{1}{|P|} \text{sign}(\tau - d_i), & \text{if} \ (x_i, y_i) \in P
\end{cases}
\]

\[
\frac{\partial L(y_i)}{\partial b_M} = \begin{cases} 
  -\frac{1}{|Q|} d_i, & \text{if} \ (x_i, y_i) \in Q \\
  \frac{1}{|P|} \text{sign}(\tau - d_i), & \text{if} \ (x_i, y_i) \in P
\end{cases}
\]

Here, the partial gradients of \(h_M(x)\) are computed as:

\[
\frac{\partial h_M(x)}{\partial b_M} = s'(W_M h_{M-1}(x) + b_M).
\]

IV. EXPERIMENTS

In this section, we present experimental results on evaluation of the proposed algorithm against several state-of-the-art methods for relative attribute learning on three benchmark datasets.

A. Datasets

We evaluate the proposed algorithm on three popularly used relative attribute learning datasets:

(1) Outdoor Scene Recognition (OSR) [2]. This dataset contains 2688 images from 8 categories including tall-building, inside-city, street, highway, coast, open-country, mountain, forest. All these 8 categories are assigned with relative values of 6 attributes: natural, open, perspective, size-large, diagonal-plane, depth-close. The strength values of the 6 relative attributes on 8 classes are shown in Table I.

(2) Public Figure Face (PubFig) [2]. This dataset contains 800 images from 8 random identities including Alex-Rodriguez, Clive-Owen, Hugh-Laurie, Jared-Leto, Miley-Cyrus, Scarlet-Johansson, Viggo-Mortensen, Zac-Efron. All these 8 identities are assigned with relative values of 11
attributes including Male, White, Young, Smiling, Chubby, Visible-Forehead, Bushy-Eyebrows, Narrow-Eyes, Pointy-Nose, Big-Lips, Round-Face. The strength values of the 11 relative attributes on 8 classes are shown in Table II.

(3) Shoes Dataset [2], [45]. This dataset includes 14658 images with 10 categories of shoes collected from like.com, athletic-shoes, boots, clogs, flats, high-heels, pumps, rainboots, sneakers, stiletto, wedding-shoes. All these categories are assigned with relative values of 10 attributes including pointy-at-the-front, open, bright-in-color, covered-with-ornaments, shiny, high-at-the-heel, long-on-the-leg, formal, sporty, feminine. The strength values of the 10 relative attributes on 10 classes are shown in Table III.

B. Evaluated Algorithms

To evaluate the effectiveness of the proposed deep relative attributes algorithm, we compare the following attributes learning methods: (1) Relative Attributes (RA) [2] algorithm learns a linear ranking function for each attribute independently with a learning to rank formulation. (2) Multi-Task Learning (MTL) [31] method learns the linear ranking functions for all attributes simultaneously in a multi-task learning framework. (3) Relative Attributes with Deep features (RAD) adopts the relative attribute learning method [2] with deep learning features extracted from the seventh fully connected layer (F7) of the AlexNet [32], [33] which is pretrained on ImageNet images for the LSVRC2012 [1]. (4) The proposed Deep Relative Attributes (DRA) model learns deep visual feature and nonlinear ranking function jointly.

C. Implementation Details

We implement the proposed DRA model based on the public deep learning library Caffe [33], and train a convolutional neural network model for each attribute. The parameter $\tau$ in the relative loss function is set to 1.0. $\lambda$ is set to 5e-5. More details are illustrated as follows.

Layer Structure: In the first convolutional layer Conv1, the input images are resized to 227×227 uniformly without cropping in our experiment. To facilitate the weight transfer from the pre-trained model, the number of maps and the output dimensions of the first 5 convolutional layers (C1, C2, C3, C4 and C5), and the first two fully connected layers (F8 and F9) are set according to the AlexNet provided in Caffe [33]. The remaining layers are newly created. The dimensions of the outputs in the F8 layer, the F9 layer, and the F10 layer are set to 1000, 500 and 1, respectively. The outputs in the last fully connected layer F10 denote the relative values of the images with regard to the attribute.

Weights Initialization: The weights in the first 5 convolutional layers and the first 2 fully connected layers are initialized according to the BVLC AlexNet model [33] which wins the large scale visual recognition challenge (LSVRC2012) [32]. The reference model is pre-trained on about 1 million images with 1000 categories on ImageNet. The remaining 3 fully connected layers F8, F9, and F10 are initialized with Gaussian filter with standard deviation 0.005 and constant bias 0.

### Table I

<table>
<thead>
<tr>
<th>Classes</th>
<th>T</th>
<th>I</th>
<th>S</th>
<th>H</th>
<th>C</th>
<th>O</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Open</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Perspective</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Size-large</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Diagonal-plane</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Depth-close</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

**TABLE I**

**RELATIVE ORDERING OF ATTRIBUTES ON OSR DATASET.** T(TALL BUILDING), I(INSIDE CITY), S(STREET), H(HIGHWAY), C(COAST), O(OPEN COUNTRY), M(MOUNTAIN), F(FOREST).

### Table II

<table>
<thead>
<tr>
<th>Classes</th>
<th>A</th>
<th>C</th>
<th>H</th>
<th>J</th>
<th>M</th>
<th>S</th>
<th>V</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>White</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Young</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Smiling</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Chubby</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Visible-forehead</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Bushy-eyebrows</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Narrow-eyes</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Pointy-nose</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Big-lips</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Round-face</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

**TABLE II**

**RELATIVE ORDERING OF ATTRIBUTES ON PUBFIG DATASET.** A(ALEX RODRIGUEZ), C(CLIVE OWEN), H(HUGH LAURIE), J(JARED LETO), M(MILEY CYRUS), S(SCARLETT JOHANSSON), V(VIGGO MORTENSEN), Z(ZAC EFRON).

### Table III

<table>
<thead>
<tr>
<th>Classes</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
<th>H</th>
<th>P</th>
<th>R</th>
<th>SN</th>
<th>ST</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pointy-at-the-front</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Open</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Bright-in-color</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Covered-with-ornaments</td>
<td>4</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Shiny</td>
<td>2</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>High-at-the-heel</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>Long-on-the-leg</td>
<td>7</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Formal</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Sporty</td>
<td>10</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Feminine</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>

**TABLE III**

**RELATIVE ORDERING OF ATTRIBUTES ON SHOES DATASET.** A(ATHLETIC SHOES), B(BOOTS), C(CLOGS), F(FLATS), H(HIGH HEELS), P(PUMPS), R(RAIN BOOTS), SN(SNEAKERS), ST(STILETTO), W(WEDDING SHOES)

### Table IV

**RANKING ACCURACIES OF THE 4 COMPARED RELATIVE ATTRIBUTE LEARNING METHODS FOR ALL 6 ATTRIBUTES ON THE OSR DATASET.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>94.82</td>
<td>96.47</td>
<td>98.20</td>
<td>99.47</td>
</tr>
<tr>
<td>Open</td>
<td>91.01</td>
<td>92.88</td>
<td>94.79</td>
<td>97.81</td>
</tr>
<tr>
<td>Perspective</td>
<td>86.56</td>
<td>88.39</td>
<td>93.66</td>
<td>97.19</td>
</tr>
<tr>
<td>Size-large</td>
<td>86.37</td>
<td>88.50</td>
<td>93.84</td>
<td>96.88</td>
</tr>
<tr>
<td>Diagonal-plane</td>
<td>88.00</td>
<td>90.87</td>
<td>94.88</td>
<td>98.46</td>
</tr>
<tr>
<td>Depth-close</td>
<td>88.35</td>
<td>89.05</td>
<td>95.18</td>
<td>97.24</td>
</tr>
</tbody>
</table>

| Avg     | 89.19  | 91.03    | 95.09 | 97.84 |

**TABLE IV**

**RANKING ACCURACIES OF THE 4 COMPARED RELATIVE ATTRIBUTE LEARNING METHODS FOR ALL 6 ATTRIBUTES ON THE OSR DATASET.**
Learning Rate: Since there are only thousand-scale training images available for each attribute, we fix the learning rates in the first 4 convolutional layers (C1, C2, C3, C4) as zeros to avoid overfitting. The learning rates for the layers C5, F6 and F7 are all set to 0.001, while the learning rates for the layers F8, F9 and F10 are all set to 0.01. The scheme for the learning rate setting is also consistent with the hierarchical nature of the features of different layers in the networks. As illustrated in [51], the output features of the bottom convolutional layers respond to corners and other edge/color conjunctions. The middle layers have more complex invariances, capturing similar textures (e.g. mesh patterns) while the top layers are more class-specific. Since the proposed convolutional networks for relative attribute learning are initialized with the weights of the model trained for the 1000-class classification task. The top layers need to be trained carefully due to the task difference. However, the bottom layers are more likely to share common features since they are the low level features, such as corners and other edge/color conjunctions. Thus, the learning rates of the bottom layers of the proposed DRA model can be assigned with small values or even fixed.

D. Learned Ranking Results

We train all the relative attribute learning methods, including RA [2], MTL [31], RAD, and the proposed DRA, using image pairs comprised of all the annotated training images in the dataset. On the OSR dataset, there are 240 training images which can generate about 10k image pairs for each attribute. On the PubFig dataset, it includes 241 training images, and can generate about 7k image pairs for each attribute. On the Shoes dataset, there are 240 training images which can generate about 6k image pairs for each attribute. We adopt the same evaluation scheme illustrated in [2]. Given a specific attribute, we predict the order of an image pair (i, j) in the test set by their relative values which are generated by the learnt relative attribute model for this attribute. The predictions are then compared to the ground-truth relative ordering. For all the evaluated algorithms, the ranking results on the OSR dataset, the PubFig dataset, and the Shoes dataset are shown in Table V, Table VI and Table VII respectively. Note that, for the MTL [31] method, we use the public code provided by the authors. The time and memory costs are extremely large for the joint ranking learning. Therefore, we use about 3k image pairs for training due to the limitation of our computer hardware. But absolutely fair comparisons between the MTL model and the proposed DRA model can be found in Figure 2.

Based on the results in Tables V, VI and VII, it is clear that the proposed DRA model consistently and significantly outperforms the state-of-the-art methods on all datasets. Compared with the relative attribute learning (RA) method [2], which is based on linear ranking SVM, the average accuracy of our DRA approach is increased about 8% on the OSR dataset, 9% on the PubFig dataset, and 14% on the Shoes dataset. The RAD method is based on the deep features extracted by the reference model trained for the large scale image classification task, and obtains better performance than the RA [2], which demonstrates the effectiveness of the deep visual features. However, the RAD method still cannot outperform the proposed DRA method. This is due to that, compared with the RAD method, 2 extra fully connected layers are added and trained in the proposed DRA for the relative attribute learning task. All these experimental results show that the proposed DRA method can learn not only much more effective task-specific visual features for image representation, but also much more effective nonlinear ranking functions to describe the relative scores of the attributes.

To show the effect of the number of training image pairs, we give the average accuracies of all attributes with different numbers of image pairs as shown in Figure 2. We can see that with a few number of image pairs, the proposed DRA model cannot show significantly better performances than other methods. This is because the proposed DRA algorithm needs more image pairs to learn the large number of model parameters. With more training image pairs, the gap between the proposed DRA model and other baseline methods is enlarged. It is worth noting that the proposed DRA can perform well with hundreds of training image pairs.
E. Zero-Shot Learning Results

The zero-shot learning is an application of relative attribute learning, and aims to classify $N$ image categories where only $S$ of them are provided with training images and no training images are provided for the other $U$ categories \cite{2}. Here, $N = S + U$ and the $S$ categories are called “seen” categories while the $U$ categories are called “unseen” during training.

With the same experimental setup as in \cite{2}, we adopt Gaussian distribution as the generative model and estimate the means $\{\mu_i\}_{i=1}^N$ and the covariance matrices $\{\Sigma_i\}_{i=1}^N$ for all the $N$ categories. (1) For the $S$ seen categories $\{c^i\}_{i=1}^S$, we learn $K$ predicting models $\{f^k(x)\}_{k=1}^K$ using the proposed DRA method for all $K$ relative attributes based on the category relationships with regard to each attribute. Then these $K$ relative attributes models are used to predict the relative values of all $K$ attributes for a given image. Thus each image $x$ from the $S$ seen categories can be represented as a $K$ dimensional vector $\tilde{x} \in \mathbb{R}^K$ indicating the relative values of all $K$ attributes. The means $\{\mu_i\}_{i=1}^S$ and the covariance matrices $\{\Sigma_i\}_{i=1}^S$ of the $S$ seen categories are estimated according to the relative values (or ranking-scores) of the training images. (2) For the $U$ unseen categories $\{c^i\}_{i=1}^U$, since there are no training images provided, the means $\{\mu_i\}_{i=1}^U$ and the covariance matrices $\{\Sigma_i\}_{i=1}^U$ of the generative Gaussian models are set based on the parameters of the seen categories and guided by the relative orders of the seen categories and the unseen categories with regard to all the $K$ attributes. For example, for the $k^{th}$ attribute $a_k$, if the unseen category $c^r_k$ is described as $c^r_k \succ c^u_k \succ c^s_k$, then the $k^{th}$ component of the mean $\mu^r_k$ is set to $(\mu^s_k + \mu^u_k)/2$. Here, $c^s_k$ and $c^u_k$ are the seen categories, $\mu^s_k$ and $\mu^u_k$ are their means of the Gaussian distributions. More details for generating the means $\{\mu_i\}_{i=1}^U$ and the covariance matrices $\{\Sigma_i\}_{i=1}^U$ for unseen categories could be found in \cite{2}.

During the testing, a new image $x$ is assigned with a $K$ dimensional vector $\tilde{x}$ by the $K$ relative attributes predicting models $\{f^k(x)\}_{k=1}^K$. It is then assigned with a seen or unseen category based on the learned generative Gaussian models of the seen and unseen categories:

$$c^* = \arg \max_{i \in \{1, \ldots, N\}} P(\tilde{x}|\mu_i, \Sigma_i)$$ (14)

The experimental results of the zero shot learning on the OSR, PubFig and Shoes datasets are shown in Figure 3. It is clear that, for the RA \cite{2}, MTL \cite{31}, RAD, and the proposed DRA methods, the zero shot image classification performances are improved significantly with the increase of the number of training image pairs.
of the seen categories. On all the three datasets, the MTL method consistently performs much better than the RA method due to the joint learning scheme. On the OSR dataset, the proposed DRA model performs the best when the number of the seen categories is greater than 4. The RAD method shows better result than the proposed DRA when there are 3 seen categories. On the PubFig dataset, the DRA method performs better than all baselines especially when the seen categories is larger than 6. On the Shoes dataset, the MTL method performs the best when the number of seen categories is smaller than 6. However, the proposed DRA outperforms the MTL method when the number of seen categories is greater than 6.

F. Discussions

Convergence Analysis: To explore the convergence of the proposed relative convolutional neural networks, in Figure 4, we show the training losses of the DRA model trained with different numbers of iterations. Here, we show the convergency curves in the first 100 iterations of all the 6 attributes on the OSR dataset. For each attribute, 500 image pairs are used for training. We can see that the relative convolutional neural networks can converge quickly especially in the first 60 iterations. In Figure 5, we show the ranking accuracies of the proposed DRA model trained with different numbers of iterations. Here, we only show the accuracies of the 6 attributes in the first 100 iterations, which makes the same conclusion that the DRA can converge extremely fast at the beginning. As illustrated in Section V-D, the Shoes dataset is the largest labeled image dataset for relative attribute learning, and only contains 240 training images which can generate about 6k image pairs for each attribute. With the scarce training data, it is important for the proposed DRA to have a fast convergence to obtain the prospective performance.

Layer Analysis: To show the effect of the layer structure in the proposed DRA, in Figure 6, we give the ranking accuracies on the OSR dataset with different layer settings. Here, the DRA is the proposed model, and the DRA1 model has the DRA structure without the F8 layer. Compared with the DRA1 model, the proposed DRA has about 1% accuracy improvements for all 6 attributes, which demonstrates the effectiveness of the fully connected layer. The DRA2 denotes the DRA1 structure without dropout after the F7 layer. Compared with the DRA1, the performance degradation of the DRA2 shows that the dropout layer is indispensable for training deep neural networks. The DRA3 denotes the DRA structure without the F8 and F9 layers. The large margin of the performance degradation shows that the F8 and F9 layers play an important role in improving the performance of the DRA for relative attribute learning.

Effect of τ: In the relative loss function, the τ controls the relative margin among the attribute values of the ordered image pair. Theoretically, the larger τ results in more distinguishable attribute values, thus obtains better ranking performance. However, the too large τ may increase the difficulty for the DRA model training. To support this point, we show the effect of τ on the OSR dataset in Figure 7. We can see that the performances are consistent while the τ is set between 0 and
Research at King Saud University, Riyadh, Saudi Arabia for authors extend their appreciation to the Deanship of Scientific Programs, Project No. 2012CB316304, and the Importation National Program on Key Basic Research Project (973 Science Foundation of China (No. 61225009, 61303173, and on the OSR dataset.

Fig. 6. Performance on the OSR dataset with different settings. DRA denotes the proposed deep relative attribute learning method. DRA1 model has the DRA structure without the $F_5$ layer. DRA2 denotes the DRA1 structure without dropout after the $F_7$ layer. DRA3 denotes the DRA structure without the $F_5$ and $F_9$ layers.

Fig. 7. Ranking accuracies with different $t$ for all the 6 relative attributes on the OSR dataset.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (No. 61225009, 61303173, 61432019, 61572498, 61532009, 61472379, 61572296), the National Program on Key Basic Research Project (973 Program, Project No. 2012CB316304), and the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under Grant IDHT20140224. The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the research group project no. RGP-229.

REFERENCES


Xiaoshan Yang received the master’s degree in computer science from Beijing Institute of Technology, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree at the Multimedia Computing Group, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He was an intern in China-Singapore Institute of Digital Media, Singapore, from Sept. 2013 to April 2014. His research interests include multimedia analysis and computer vision.
IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.

He is session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He served as Program Chair of ACM Multimedia 2009. He has served as an Associate Editor of IEEE Trans. on Multimedia, Comm. of ACM/IEEE Transactions on Multimedia Computing, Communications and Applications in 2012 and the Best Editorial Member Award of ACM/Springer Multimedia Systems Journal in 2008. He served as Program Chair of ACM Multimedia 2009. He has served as an Associate Editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He is IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.

M. Shamim Hossain (Graduate Student M’02–SM’09) is an Associate Professor with the Department of Software Engineering, King Saud University, Riyadh, KSA. Dr. Shamim Hossain received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Canada, in 2009. His research interests include Internet of things (IoT), cloud and multimedia for healthcare, Multimedia big data, social media, serious games, and biologically inspired approach for multimedia and software system. He has authored and co-authored around 100 publications including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. He has served as a member of the organizing and technical committees of several international conferences and workshops. He was the recipient of the Outstanding Paper award from an IEEE Conference, and “Research in Excellence” award from King Saud University. He has served as co-chair, general chair, workshop chair, publication chair, publicity chair, and TPC for over 12 IEEE and ACM conferences and workshops. He is on the editorial board of IEEE Access, and International Journal of Multimedia Tools and Applications (Springer). He serves/serve as a Lead Guest Editor of IEEE Transactions on Cloud Computing, IEEE Transactions on Information Technology in Biomedicine (Currently, IEEE J-BHI), IEEE Communication Magazine, Future Generation Computer Systems (Elsevier), Computers & Electrical Engineering (Elsevier), International Journal of Multimedia Tools and Applications (Springer), Cluster Computing (Springer), SENSORS (MDPI) and International Journal of Distributed Sensor Networks (Hindawi). Dr. Shamim is a Senior Member of IEEE, a member of ACM SIGMM.