Deep Relative Attributes

Article · December 2015

Source: arXiv

CITATIONS
10

READS
88

3 authors, including:

Erfan Noury
University of Maryland, Baltimore County
3 PUBLICATIONS **27** CITATIONS
SEE PROFILE

Ehsan Adeli
Stanford University
73 PUBLICATIONS **478** CITATIONS
SEE PROFILE

Deep Relative Attributes

Yaser Souri[1], Erfan Noury[2], Ehsan Adeli-Mosabbeb[3]
[1]Sobhe [2]Sharif University of Technology [3]University of North Carolina at Chapel Hill
souri@sobhe.ir, erfan.noury@live.com, eadeli@unc.edu

Abstract

Visual attributes are great means of describing images or scenes, in a way both humans and computers understand. In order to establish a correspondence between images and to be able to compare the strength of each property between images, relative attributes were introduced. However, since their introduction, hand-crafted and engineered features were used to learn complex models for the problem of relative attributes. This limits the applicability of those methods for more realistic cases. We introduce a two part deep learning architecture for the task of relative attribute prediction. A convolutional neural network (ConvNet) architecture is adopted to learn the features with addition of an additional layer (ranking layer) that learns to rank the images based on these features. Also an appropriate ranking loss is adapted to train the whole network in an end-to-end fashion. Our proposed method outperforms the baseline and state-of-the-art methods in relative attribute prediction on various datasets. Our qualitative results also show that the network is able to learn effective features for the task. Furthermore, we use our trained models to visualize saliency maps for each attribute.

1. Introduction

Visual attributes are linguistic terms that bear semantic properties of (visual) entities, often shared among categories. They are both human understandable and machine detectable, which makes them appropriate for better human machine communications. Visual attributes have been successfully used for many applications, such as image search [20], interactive fine-grained recognition [1, 2] and zero-shot learning [23, 30].

Traditionally, visual attributes were treated as binary concepts [10, 8], as if they are present or not, in an image. Parikh and Grauman [30] introduced a more natural view on visual attributes, in which pairs of visual entities can be compared, with respect to their relative strength of any specific property. With a set of human assessed relative orderings of image pairs, they learn a global ranking function for each attribute (Figure 1). While binary visual attributes relate properties to entities (*e.g.*, a dog being furry), relative attributes make it possible to relate entities to each other in terms of their properties (*e.g.*, a bunny being furrier than a dog).

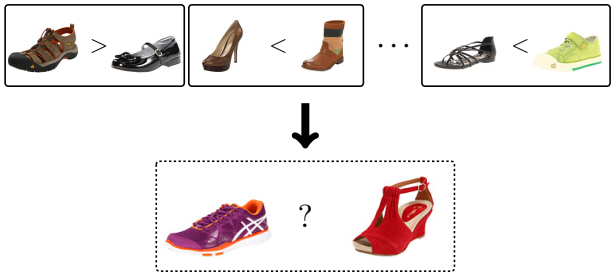

Figure 1: Visual Relative Attributes. This figure shows samples of training pairs of images from the UT-Zap50K dataset, comparing shoes in terms of the *comfort* attribute (top). The goal is to compare a pair of two novel images of shoes, respective to the same attribute (bottom).

Many have tried to build on the seminal work of Parikh and Grauman [30] with more complex and task-specific models for ranking, while still using hand crafted visual features, such as GIST [29] and HOG [4]. Recently, Convolutional Neural Networks (ConvNets) have proved to be successful in various visual recognition tasks, such as image classification [21], object detection [11] and image segmentation [28]. Many attribute the success of ConvNets to their ability to learn multiple layers of visual features from data.

In this work, we propose to use a ConvNet-based architecture to learn the ranking of images, using relatively annotated images for each attribute. Pairs of images with similar and/or different strengths of some particular attribute are presented to the network. The network learns a series of visual features, which are known to work better than engineered visual features [31] for various tasks. After feature learning and extraction layers, we further propose to add another layer to the network, to rank the images. The layers could be learned through gradient descent (described in more details later) simultaneously. As a result, it would be

possible to learn (or fine-tune) the features through back-propagation, while learning the ranking layer. Interweaving the two processes leads to a set of learned features that appropriately characterize each single attribute. Our qualitative investigation of the learned feature space further confirms this assumption. This escalates the overall performance and is the main advantage of our proposed method over previous methods. Furthermore, in almost all previous works on relative attributes, the training phase often can only handle inequality relations (*i.e*., one image being less or more strong, in terms of a specific attribute, than the other). The equality relation can happen quite frequently when humans are qualitatively deciding about the relations of attributes in images. In previous works this is not explored, even in some, the equality relation could not be incorporated in the training phase, due to method limitations. Our proposed method introduces an easy and elegant way to deal with equality relations (*i.e*., an attribute is similarly strong in two images).

It is noteworthy to pinpoint that by exploiting the saliency maps of the unsupervised learned features for each attribute, similar to [35], we can discover the pixels which contribute the most towards an attribute in the image. This can be used to coarsely localize the attribute.

Our approach achieves very competitive results and improves the state-of-the-art (with a large margin in some datasets) on major publicly available datasets for relative attribute prediction, both coarse and fine-grained.

The rest of the paper is organized as follows: Section 2 discusses the related works. Section 3 illustrates our proposed method. Then, Section 4 exhibits the experimental setup and results, and finally, Section 5 concludes the paper.

2. Related Works

We usually describe visual concepts with their attributes.Attributes are, therefore, mid-level representations for describing objects and scenes. In an early work on attributes, Farhadi *et al*. [8] proposed to describe objects using mid-level attributes. In another work [9], the authors described images based on a semantic triple "object, action, scene". In the recent years, attributes have shown great performance in object recognition [8, 39], action recognition [18, 26] and event detection [27]. Lampert *et al*. [23] predicted unseen objects using a zero-shot learning framework, in which binary attribute representation of the objects were incorporated.

On the other hand, comparing attributes enables us to easily and reliably search through high-level data derived from *e.g*., documents or images. For instance, Kovashka *et al*. [19] proposed a relevance feedback strategy for image search using attributes and their comparisons. In order to establish the capacity for comparing attributes, we need to move from binary attributes towards describing attributes relatively. In the recent years, relative attributes have attracted the attention of many researchers. For instance, a linear relative comparison function is learned in [30], based on RankSVM [16] and a non-linear strategy in [25]. In another work, Datta *et al*. [5] used trained rankers for each facial image feature and formed a global ranking function for attributes.

Through the process of learning the attributes, different types of low-level image features are incorporated. For instance, Parikh and Grauman [30] used 512-dimensional GIST [29] descriptors as image features, while Jayaraman *et al*. [15] used histograms of image features, and reduced their dimensionality using PCA. Other works tried learning attributes through *e.g*., local learning [46] or fine-grained comparisons [44]. Yu and Grauman [44] proposed a local learning-to-rank framework for fine-grained visual comparisons, in which the ranking model is learned using only analogous training comparisons. In another work [45], they proposed a local Bayesian model to rank images, which are hardly distinguishable for a given attribute. However, none of these methods leverage the effectiveness of feature learning methods and only use engineered and hand-crafted features for predicting relative attributes.

As could be inferred from the literature, it is very hard to decide what low-level image features to use for identifying and comparing visual attributes. Recent studies show that features learned through the convolutional neural networks (CNNs) [24] (also known as deep features) could achieve great performance for image classification [22] and object detection [12]. Zhang *et al*. [47] utilized CNNs for classifying binary attributes. In other works, Escorcia *et al*. [7] proposed CCNs with attribute centric nodes within the network for establishing the relationships between visual attributes. Shankar *et al*. [34] proposed a weakly supervised setting on convolutional neural networks, applied for attribute detection. Khan *et al*. [17] used deep features for describing human attributes and thereafter for action recognition, and Huang *et al*. [14] used deep features for cross-domain image retrieval based on binary attributes.

Neural networks have also been extended for learning-to-rank applications. One of the earliest networks for ranking was proposed by Burges*et al*. [3], known as RankNet. The underlying model in RankNet maps an input feature vector to a Real number. The model is trained by presenting the network pairs of input training feature vectors with differing labels. Then, based on how they should be ranked, the underlying model parameters are updated. This model is used in different fields for ranking and retrieval applications, *e.g*., for personalized search [37] or content-based image retrieval [43].

3. Proposed Method

We propose to use a ConvNet-based deep neural network that is trained to optimize an appropriate ranking loss for the task of predicting relative attribute strength. The network architecture consists of two parts, the *feature learning and extraction* part and the *ranking* part.

The feature learning and extraction part takes a fixed size image, I_i, as input and outputs the learned feature representation for that image $\psi_i \in \mathbb{R}^d$. Over the past few years, different network architectures for computer vision problems have been developed. These deep architectures can be used for extracting and learning features for different applications. For the current work, outputs of an intermediate layer, like the last layer before the probability layer, from a ConvNet architecture (*e.g.*, AlexNet [21], VGGNet [36] or GoogLeNet [38]) can be incorporated.

One of the most widely used models for relative attributes in the literature is RankSVM [16]. However, in our case, we seek a neural network based ranking procedure, to which relatively ordered pairs of feature vectors are inputted during training. This procedure should learn to map each feature vector to an absolute ranking, for testing purpose. Burges *et al*. [3] introduced such a neural network based ranking procedure that exquisitely fits our needs. We adopt a similar strategy and thus, the ranking part of our proposed network architecture is analogous to [3] (referred to as RankNet).

During training for a minibatch of image pairs and their target orderings, the output of the feature learning and extraction part of the network is fed into the ranking part and a ranking loss is computed. The loss is then back-propagated through the network, which enables us to simultaneously learn the weights of both feature learning and extraction (ConvNet) and ranking (RankNet) parts of the network. Further with back-propagation we can calculate the derivative of the estimated ordering with respect to the pixel values. In this way, we can simply generate saliency maps for each attribute (see section 4.6). These saliency maps exhibit interesting properties, as they can be used to localize the pixels in the image that are informative about the strength of presence of the attribute.

3.1. RankNet: Learning to Rank Using Gradient Descent

This section briefly overviews the RankNet [3] procedure in our context. Given a set of pairs of sample feature vectors $\left\{(\psi_1^{(k)}, \psi_2^{(k)})|k \in \{1, \ldots, n\}\right\} \in \mathbb{R}^{d\times d}$, and target probabilities $\left\{t_{12}^{(k)}|k \in \{1, \ldots, n\}\right\}$, sample $\psi_1^{(k)}$ is to be ranked higher than sample $\psi_2^{(k)}$. We would like to learn a ranking function $f : \mathbb{R}^d \mapsto \mathbb{R}$, such that f specifies the ranking order of a set of features. Here, $f(\psi_i) > f(\psi_j)$ indicates that the feature vector ψ_i is ranked higher than ψ_j, denoted by $\psi_i \triangleright \psi_j$. The RankNet model [3] provides an elegant procedure based on neural networks to learn the function f.

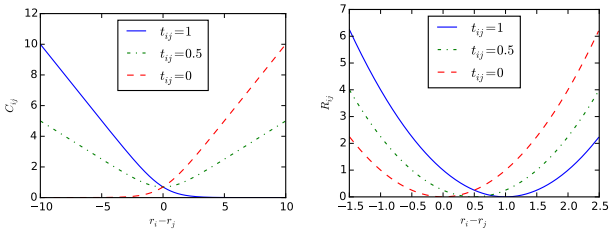

Figure 2: The ranking loss value for three values of the target probability (left). The squared loss value for three values of the target probability, typically used for regression (right).

Denoting $r_i \equiv f(\psi_i)$, RankNet models the mapping from rank estimates to posterior probabilities $p_{ij} = P(\psi_i \triangleright \psi_j)$ using a logistic function

$$p_{ij} := \frac{1}{1 + e^{-(r_i - r_j)}}. \tag{1}$$

The loss for the sample pair of feature vectors (ψ_i, ψ_j) along with target probability t_{ij} is defined as

$$C_{ij} := -t_{ij} \log(p_{ij}) - (1 - t_{ij}) \log(1 - p_{ij}), \tag{2}$$

which is the binary cross entropy loss. Figure 2 (left) plots the loss value C_{ij} as a function of $r_i - r_j$ for three values of target probability $t_{ij} \in \{0, 0.5, 1\}$. This function is quite suitable for ranking purposes, as it acts differently compared to regression functions.

Specifically, we are not interested in regression instead of the ranking for two reasons: First, we cannot regress the absolute rank of images, since the annotations are only available in pairwise ordering for each attribute, in relative attribute datasets (see section 4.1). Second, regressing the difference $r_i - r_j$ to t_{ij} is also inappropriate. Let's consider the squared loss

$$R_{ij} = \left[(r_i - r_j) - t_{ij}\right]^2, \tag{3}$$

which is typically used for regression, illustrated in Figure 2 (right). We observe that the regression loss forces the difference of rank estimates to be a specific value and disallows over-estimation. Furthermore, its quadratic natures makes it sensitive to noise. This sheds light into why regression objective is the wrong objective to optimize when the goal is ranking.

Note that when $t_{ij} = 0.5$, and no information is available about the relative rank of the two samples, the ranking cost becomes symmetric. This can be used as a way to

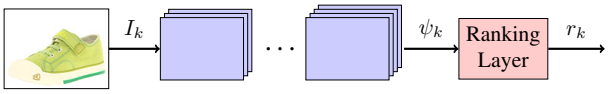

Figure 4: During testing, we only need to evaluate r_k for each test image. Using this value, we can easily infer the relative or absolute ordering of test images, regarding the attribute of interest.

train on patterns that are desired to have similar ranks. This is somewhat scarce in the previous works on relative attributes. Furthermore, this model asymptotically converges to a linear function which makes it more appropriate for problems with noisy labels.

Training this model is possible using stochastic gradient descent or its variants like RMSProp.While testing, we only need to estimate the value of $f(\psi_i)$, which resembles the absolute rank of the test sample. Using $f(\psi_i)$s, we can easily infer the relative ordering or absolute ordering of test pairs.

3.2. Deep Relative Attributes

Our proposed model is depicted in figure 3. The model is trained separately, for each attribute. During training, pairs of images (I_i, I_j) are presented to the network, together with the target probability t_{ij}. If for the attribute $I_i \triangleright I_j$ (image i exhibits more of the attribute than image j), then t_{ij} is expected to be larger than 0.5 depending on our confidence on the relative ordering of I_i and I_j. Similarly, if $I_i \triangleleft I_j$, then t_{ij} is expected to be smaller than 0.5, and if it is desired that the two images have the same rank, t_{ij} is expected to be 0.5. Because of the nature of the datasets, we chose t_{ij} from the set $\{1, 0.5, 0\}$, according to the available annotations in the dataset.

The pair of images then go though the feature learning and extraction part of the network (ConvNet). This procedure maps the images onto feature vectors ψ_i and ψ_j, respectively. Afterwards, these feature vectors go through the ranking layer, as described in section 3.1. We choose the ranking layer to be a fully connected neural network layer with linear activation function, a single output neuron and weights w and b. It maps the feature vector ψ_i to the estimated absolute rank of that feature vector r_i, where

$$r_i := w^T \psi_i + b \tag{4}$$

and $r_i \in \mathbb{R}$. The two estimated ranks r_i and r_j are then combined to output the estimated posterior probability $p_{ij} = P(I_i \triangleright I_j)$, which is used along with the target probability t_{ij} to calculate the loss. This loss is then back-propagated through the network and is used to update the weights of the whole network, including both the weights of the feature learning and extraction sub-network and the ranking layer.

During testing, as shown in Figure 4, we only need to calculate the estimated absolute rank r_k for each test image I_k. Using these estimated absolute ranks we can then easily infer the relative or absolute attribute ordering, for all test pairs.

4. Experiments

To evaluate our proposed method, we quantitatively compare it with the state-of-the art methods, as well as an informative baseline on all publicly available benchmarks for relative attributes to our knowledge. Furthermore, we perform multiple qualitative experiments to show the capability and superiority of our method.

4.1. Datasets

To assess the performance of the proposed method, we have evaluate it on all publicly available datasets to our knowledge: **Zappos50K** [44] (both coarse and fine-grained versions), **LFW-10** [33] and for the sake of completeness and comparison with previous works, on **PubFig** and **OSR** datasets of [30].

UT-Zap50K [44] dataset is a collection of images with annotations for relative comparison of 4 attributes. This dataset contains two collections: Zappos50K-1, in which relative attributes are annotated for coarse pairs, where the comparisons are relatively easy to interpret, and Zappos50K-2, where relative attributes are annotated for fine-grained pairs, for which making the distinction between them is hard according to human annotators. Training set for Zappos50K-1 contains approximately 1500 to 1800 annotated pairs of images for each attribute. These are divided into 10 train/test splits which are provided alongside the dataset and used in this work. Meanwhile, Zappos50K-2 only contains a test set of approximately 4300 pairs, which are used for training the set of images in Zappos50K-1.

We have also conducted experiments on the **LFW-10** [33] dataset. This dataset has 2000 images of faces of people and annotations for 10 attributes. For each attribute, a random subset of 500 pairs of images have been annotated for each train and test set.

PubFig [30] dataset (a set of public figure faces), consists of 800 facial images of 8 random subjects, with 11 attributes. **OSR** [30] dataset contains 2688 images of outdoor scenes in 8 categories, for which 6 relative attributes are defined. The ordering of samples in both PubFig and OSR datasets are annotated in a category level, *i.e.*, all images in a specific category may be ranked higher, equal, or lower than all images in another category, with respect to an attribute. This sometimes causes annotation inconsistencies [33]. In our experiments, we have used the provided train/test split of PubFig and OSR datasets.

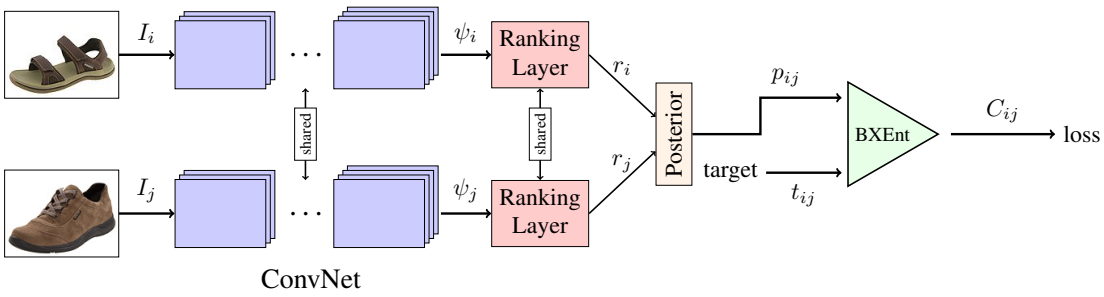

Figure 3: The overall schematic view of the proposed method during training. The network consists of two parts, the *feature learning and extraction* part (labeled ConvNet in the figure), and the *ranking* part (the Ranking Layer). Pairs of images are presented to the network with their corresponding target probabilities. This is used to calculate the loss, which is then back-propagated through the network to update the weights.

4.2. Experimental setup

We train our proposed model (described in Section 3) for each attribute, separately. In our propoised model, it is possible to train multiple attributes at the same time, however, this is not done due to the structure of datasets, where for each training pair of images only a certain attribute is annotated.

We have used the Lasagne [6] deep learning framework to implement our model. In all our experiments, for the feature learning and extraction part of the network, we use the VGG-16 model of [36] and trim out the probability layer (all layers up to fc7 are used, only the probability layer is not included). We initialize the weights of the model using a pretrained model on ILSVRC 2014 dataset [32] for the task of image classification. These weights are fine-tuned as the network learns to predict the relative attributes (see section 4.5). The weights w of the ranking layer are initialized using the Xavier method [13], and the bias is initialized to 0.

For training, we use stochastic gradient descent with RMSProp [40] updates and minibatches of size 32 (16 pair of images). We set the learning rate for all experiments to 10^{-4} (for all weights and biases both the feature learning and extraction layers and the ranking layer), initially, then RMSProp changes the learning rates according to its algorithm. We have also used weight decay (ℓ_2 norm regularization), with a fixed 0.005 multiplier. Furthermore, when calculating the binary cross entropy loss, we clip the estimated posterior p_{ij} to be in the range $[10^{-7}, 1 - 10^{-7}]$. This is used to prevent the loss from diverging.

In each epoch, we randomly shuffle the training pairs. The number of epochs of training were chosen to reflect the training size. For Zappos50K and LFW-10 datasets, we train for 5 and 50 epochs, respectively. For PubFig and OSR datasets, we train for 120 epochs due to the small number of training samples. Also, we have added random horizontal flipping of the training images as a way to augment the training set for the PubFig and OSR datasets.

4.3. Baseline

As a baseline, we have also included results for the RankSVM method (as in [30]), when the features given to the method were computed from the output of the VGG-16 pretrained network on ILSVRC 2014.

Using this baseline we can evaluate the extent of effectiveness of off-the-shelf ConvNet features [31] for the task of ranking. In a sense, comparing this baseline with our proposed method reveals the effect of features fine-tuning, for the task.

4.4. Quantitative Results

Following [30, 44, 33], we report the accuracy in terms of the percentage of correctly ordered pairs. For our proposed method, we report the mean accuracy and standard deviation over 3 separate runs.

Table 1 shows our results on the OSR dataset. Our method outperforms the baseline and the state-of-the-art on this dataset, on all attributes except for 'Natural' attribute, where the baseline outperforms our method with a small margin. One possible cause of this could be that the pretrained network of the baseline is still appropriate for this dataset, since the dataset contains natural images. This is a relatively easy dataset, and we would assume that it cannot show the ability of our method to learn better features.

Table 2 shows our results on the PubFig dataset. On this dataset, our results are very competitive with the state-of-the-art. We think this is due to label inconsistency in this dataset, low number of training samples, and the fact that the images in the dataset are very tightly cropped to the face. This makes the decision about the attributes very local, while our method performs the ranking and feature extraction in a global manner.

Table 3 shows our results on the LFW-10 dataset. On this dataset, our method outperforms the state-of-the-art by

Table 1: Results for the OSR dataset

Method	Natural	Open	Perspective	Large Size	Diag	ClsDepth	Mean
Relative Attributes [30]	95.03	90.77	86.73	86.23	86.50	87.53	88.80
Relative Forest [25]	95.24	92.39	87.58	88.34	89.34	89.54	90.41
Fine-grained Comparison [44]	95.70	94.10	90.43	91.10	92.43	90.47	92.37
VGG16-fc7 (baseline)	97.98	87.82	89.01	88.25	89.91	90.70	90.61
RankNet (ours)	97.76 (± 0.25)	94.48 (± 0.90)	92.37 (± 0.34)	92.70 (± 1.01)	95.14 (± 0.26)	91.44 (± 2.69)	**93.98** (± 0.35)

Table 2: Results for the PubFig dataset

Method	Male	White	Young	Smiling	Chubby	Forehead	Eyebrow	Eye	Nose	Lip	Face	Mean
Relative Attributes [30]	81.80	76.97	83.20	79.90	76.27	87.60	79.87	81.67	77.40	79.17	82.33	80.53
Relative Forest [25]	85.33	82.59	84.41	83.36	78.97	88.83	81.84	83.15	80.43	81.87	86.31	83.37
Fine-grained Comparison [44]	91.77	87.43	91.87	87.00	87.37	94.00	89.83	91.40	89.07	90.43	86.70	**89.72**
VGG16-fc7 (baseline)	80.84	73.39	79.41	76.23	74.69	80.52	75.38	77.78	76.15	78.14	80.01	77.50
RankNet (ours)	90.10 (± 1.05)	89.49 (± 0.59)	89.83 (± 0.37)	88.62 (± 1.59)	88.72 (± 0.75)	92.33 (± 0.80)	88.13 (± 1.83)	86.94 (± 3.36)	86.30 (± 1.60)	89.79 (± 0.45)	92.71 (± 1.87)	**89.36** (± 0.57)

a large margin. Meanwhile, the baseline cannot achieve a comparative result. This shows that for achieving good results, the feature learning and extraction part have had a large impact.

Tables 4 and 5 show the results on Zappos50K-1 and Zappos50K-2 datasets, respectively. Our method, again, achieves the state-of-the-art accuracy on both fine-grained and coarse-grained datasets. However, our baseline method does not achieve a good result on this dataset. We would assume this is due to the fact that the images in the Zappos50K dataset are not natural images. So the extracted features in the baseline method are not appropriate for ranking. But our proposed method learns appropriate features for the task, given the large amount of training data available in this dataset.

Table 4: Results for the UT-Zap50K-1 (coarse) dataset

Method	Open	Pointy	Sporty	Comfort	Mean
Relative Attributes [30]	87.77	89.37	91.20	89.93	89.57
Fine-grained Comparison [44]	90.67	90.83	92.67	92.37	91.64
VGG16-fc7 (baseline)	62.33	59.57	61.33	61.00	61.08
RankNet (ours)	93.00 (± 1.15)	92.11 (± 1.58)	95.56 (± 1.26)	93.22 (± 2.41)	**93.47** (± 0.59)

Table 5: Results for the UT-Zap50K-2 (fine-grained) dataset

Method	Open	Pointy	Sporty	Comfort	Mean
Relative Attributes [30]	60.18	59.56	62.70	64.04	61.62
Fine-grained Comparison [44]	74.91	63.74	64.54	62.51	66.43
LocalPair + ML + HOG [42]	76.2	65.3	64.8	63.6	67.5
VGG16-fc7 (baseline)	52.55	52.65	51.52	53.01	52.43
RankNet (ours)	71.21 (± 1.50)	66.64 (± 1.81)	67.81 (± 0.84)	65.88 (± 1.92)	**67.88** (± 0.83)

4.5. Qualitative Results

Our proposed method uses a deep network with two parts, the feature learning and extraction part and the ranking part. During training, not only the weights for the ranking part are learned, but also the weights for the feature learning and extraction part of the network, which were initialized using a pretrained network, are fine-tuned. By fine-tuning the features, our network learns a set of features that are more appropriate for the images of that particular dataset, along with the attribute of interest. To show the effectiveness of fine-tuning the features of the feature learning and extraction part of the network, we have projected them into 2-D space using the t-SNE [41] method, as can be seen in Figure 5. The visualizations on the top of each figure show the images projected into 2-D space from the fine-tuned feature space. Each image is displayed as a point. It is clear from these visualizations that fine-tuned feature space is better in capturing the ordering of images with respect to the respective attribute. Since t-SNE embedding is a non-linear embedding, relative distances between points in the high-dimensional space and the low-dimensional embedding space are preserved, thus close points in the low-dimensional embedding space are also close to each other in the high-dimensional space. It can, therefore, be seen that fine-tuning indeed changes the feature space such that images with similar values of the respective attribute get projected into a close vicinity of the feature space. However, in the original feature space, images are projected according to their visual content, regardless of their value of the attribute.

Another property of our network is that it can achieve a total ordering of images, given a set of pairwise orderings. In spite of the fact that training samples are pairs of images annotated according to their relative value of the attribute,

Table 3: Results for the LFW-10 dataset

Method	Bald	DkHair	Eyes	GdLook	Mascu.	Mouth	Smile	Teeth	FrHead	Young	Mean
Fine-grained Comparison [25]	67.9	73.6	49.6	64.7	70.1	53.4	59.7	53.5	65.6	66.2	62.4
Relative Attributes [30]	70.4	75.7	52.6	68.4	71.3	55.0	54.6	56.0	64.5	65.8	63.4
Relative Parts [33]	71.8	80.5	90.5	77.6	67.0	77.6	81.3	76.2	80.2	82.4	78.5
Global + HOG [42]	78.8	72.4	70.7	67.6	84.5	67.8	67.4	71.7	79.3	68.4	72.9
VGG16-fc7 (baseline)	70.44	69.14	59.40	59.75	84.48	56.04	57.63	57.85	59.38	70.36	64.45
RankNet (ours)	81.27 (± 1.47)	88.92 (± 1.63)	91.98 (± 2.42)	72.03 (± 1.25)	95.40 (± 1.52)	89.04 (± 2.18)	84.75 (± 0.28)	89.33 (± 0.47)	84.11 (± 2.77)	73.35 (± 1.13)	**85.02** (± 0.59)

the network can generalize the relativity of attribute values to other pairs of images. Figure 6 shows that images can be ordered according to their value of the respective attribute.

4.6. Saliency Maps and Localizing the Attributes

We have also used the method of [35] to visualize the saliency of each attribute. Giving two image as inputs to the network, we take the derivative of the estimated posterior with respect to the input images and visualize them. Figure 7 shows some sample visualization for the LFW-10 dataset's test pairs. To generate this figure we have applied Gaussian smoothing to the saliency map. Using this visualization, we can localize the attribute using the same network that was trained to rank the attributes in an unsupervised manner, *i.e.*, although we haven't explicitly trained our network to localize the salient and informative regions of the image, it has implicitly learned to find these regions. We see that this technique is able to localize both easy attributes such as "Open Eyes" and abstract attributes such as "Good Looking". Our approach reveals some interesting properties about salient pixels for each attribute. For example, for the attribute "Open Eyes", not only pixels belonging to the eye are salient, but the pixels which belong to the mouth are also salient. This is because the person with open mouth usually has open eyes.

5. Conclusion

In this paper, we introduced an approach for relative attribute prediction on images, based on a convolutional neural network. Unlike previous methods that use engineered or hand-crafted features, our proposed method learns attribute-specific features, on-the-fly, during the learning procedure of the ranking function. Our results achieve state-of-the-art performance in relative attribute prediction on various datasets both coarse- and fine-grained. We qualitatively show that the feature learning and extraction part, effectively learns appropriate features for each attribute and dataset. Furthermore, we show that one can use a trained model for relative attribute prediction to obtain saliency maps for each attribute in the image.

References

[1] S. Branson, O. Beijbom, and S. Belongie. Efficient large-scale structured learning. In *CVPR*, 2013. 1

[2] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010. 1

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005. 2, 3

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 1

[5] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *FG*, pages 36–42, 2011. 2

[6] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, D. K. Rasul, CongLiu, Britefury, and J. Degrave. Lasagne: First release., Aug. 2015. 5

[7] V. Escorcia, J. Carlos Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *CVPR*, 2015. 2

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2

[9] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, pages 15–29. 2010. 2

[10] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, pages 433–440, 2007. 1

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010. 5

[14] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, 2015. 2

[15] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*, pages 1629–1636, 2014. 2

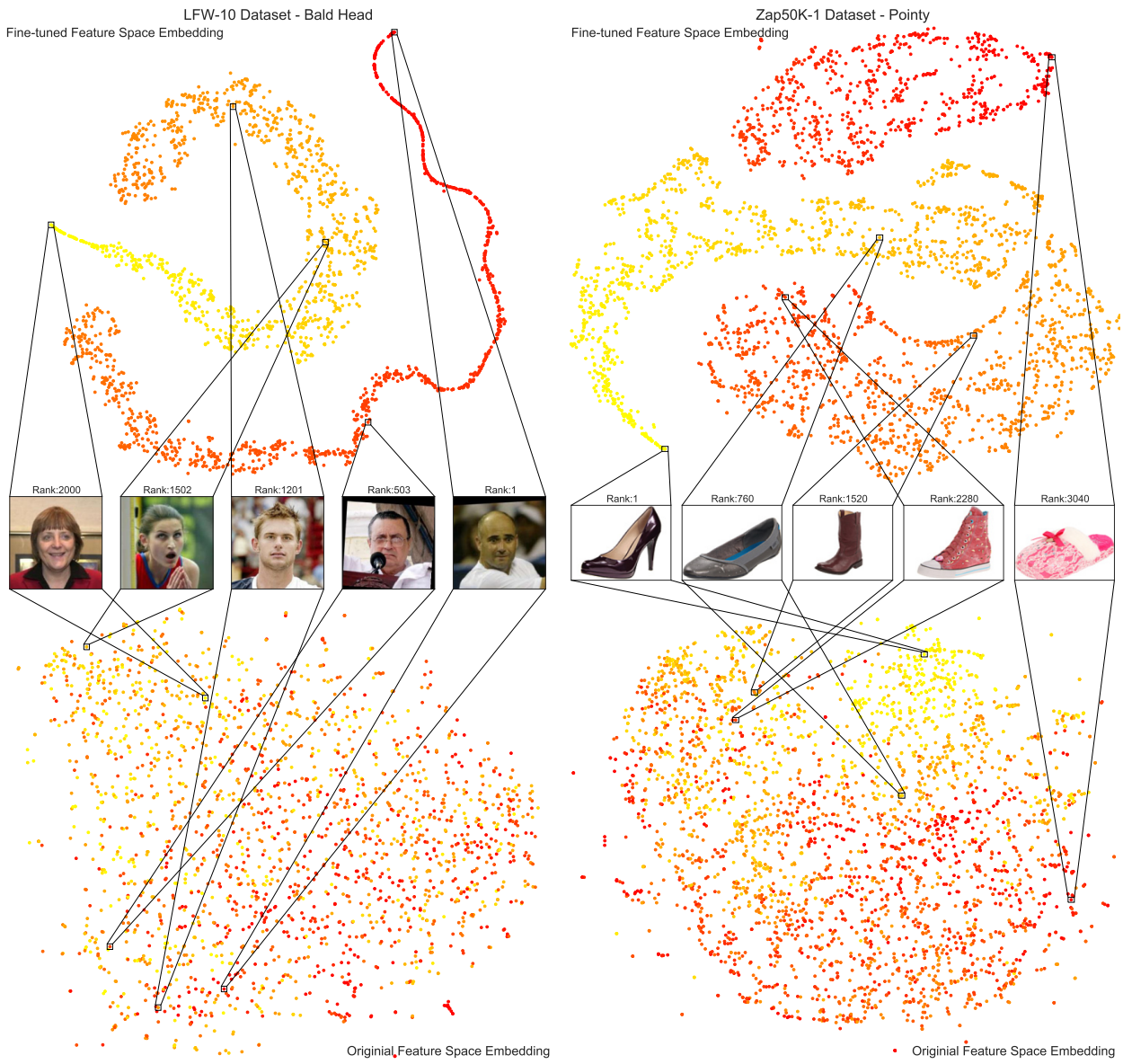

Figure 5: t-SNE embedding of images in fine-tuned feature space (top) and original feature space (bottom). The set of visualizations on the left are for the *Bald Head* attribute of the LFW-10 dataset. The set of visualizations on the right are for the *Pointy* attribute of the Zappos50K-1 dataset. Images in the middle row show a number of samples from the feature space. It is clear that images are ordered according to their value of the attribute. Each point is colored according to its value of the respective attribute, to discriminate images according to their value of the attribute.

[16] T. Joachims. Optimizing search engines using clickthrough data. In *ACM KDD*, pages 133–142, 2002. 2, 3

[17] F. Khan, R. Anwer, J. van de Weijer, M. Felsberg, and J. Laaksonen. Deep semantic pyramids for human attributes and action recognition. In *SCIA*, pages 341–353. 2015. 2

[18] F. Khan, J. van de Weijer, R. Anwer, M. Felsberg, and C. Gatta. Semantic pyramids for gender and action recognition. *IEEE TIP*, 23(8):3633–3645, 2014. 2

[19] A. Kovashka and K. Grauman. Attribute pivots for guiding relevance feedback in image search. In *ICCV*, pages 297–304, 2013. 2

[20] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image Search with Relative Attribute Feedback. In *CVPR*, 2012. 1

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1097–1105. 2012. 1, 3

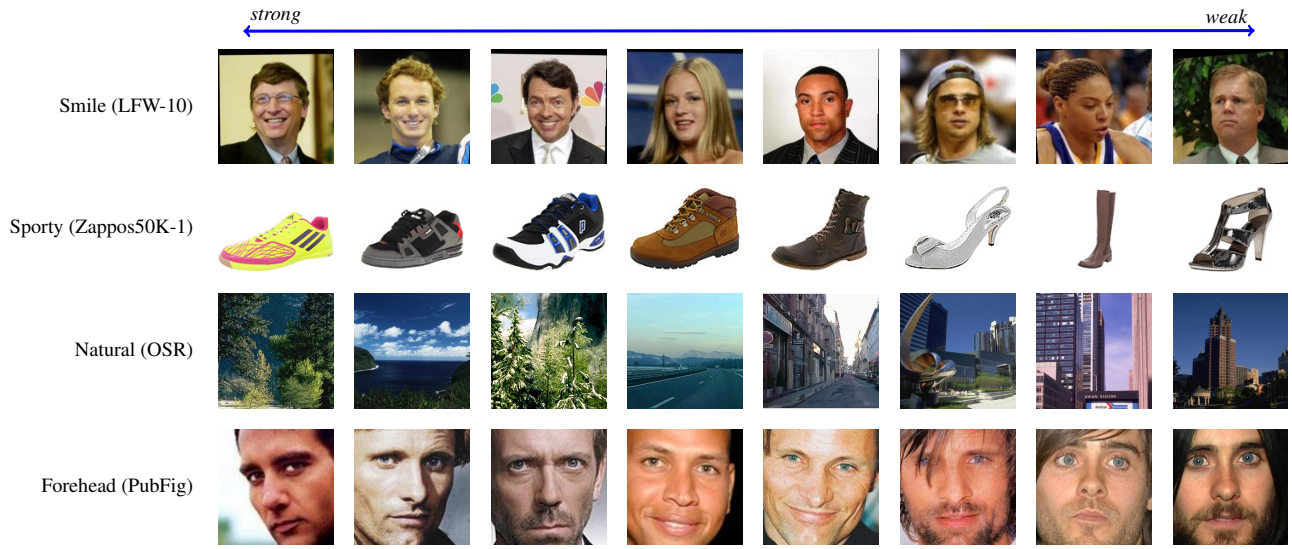

Figure 6: Sample images from different datasets, ordered according to the predicted value of their respective attribute.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012. 2

[23] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3):453–465, 2014. 1, 2

[24] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, pages 396–404. 1990. 2

[25] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*, volume 7724, pages 316–327. 2013. 2, 6, 7

[26] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, pages 3337–3344, 2011. 2

[27] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney. Video event recognition using concept attributes. In *WACV*, pages 339–346, 2013. 2

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[29] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 1, 2

[30] D. Parikh and K. Grauman. Relative attributes. *CVPR*, pages 503–510, 2011. 1, 2, 4, 5, 6, 7

[31] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPRW*, pages 512–519, 2014. 1, 5

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, pages 1–42. 5

[33] R. N. Sandeep, Y. Verma, and C. V. Jawahar. Relative parts: Distinctive parts for learning relative attributes. In *CVPR*, 2014. 4, 5, 7

[34] S. Shankar, V. K. Garg, and R. Cipolla. Deep-carving: Discovering visual attributes by carving deep neural nets. In *CVPR*, 2015. 2

[35] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2, 7, 10

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 5

[37] Y. Song, H. Wang, and X. He. Adapting deep ranknet for personalized search. In *WSDM*, 2014. 2

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 3

[39] R. Tao, A. W. Smeulders, and S.-F. Chang. Attributes and categories for generic instance search from one example. In *CVPR*, pages 177–186, 2015. 2

[40] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012. 5

[41] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(2579-2605):85, 2008. 6

[42] Y. Verma and C. Jawahar. Exploring locally rigid discriminative patches for learning relative attributes. 6, 7

[43] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *ACM MM*, pages 157–166, 2014. 2

[44] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 2, 4, 5, 6

Bald Head

Smile

Open Eyes

Good Looking

Figure 7: Saliency maps obtained from the network. First we feed two test images into the network and compute the derivative of the estimated posterior with respect to the pair of input images and use the method of [35] to visualize salient pixels with Gaussian smoothing. In each row, the two input images from the LFW-10 test set with their corresponding overlaid saliency maps are shown (the warmer the color of the overlay image, the more salient that pixel is).

[45] A. Yu and K. Grauman. Just noticeable differences in visual attributes. In *ICCV*, 2015. 2

[46] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, volume 2, pages 2126–2136, 2006. 2

[47] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose aligned networks for deep attribute modeling. In *CVPR*, pages 1637–1644, 2014. 2