A Deep Sum-Product Architecture for Robust Facial Attributes Analysis

Ping Luo[1,3] Xiaogang Wang[2] Xiaoou Tang[1,3*]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]Department of Electronic Engineering, The Chinese University of Hong Kong
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
pluo.lhi@gmail.com xgwang@ee.cuhk.edu.hk xtang@ie.cuhk.edu.hk

Abstract

Recent works have shown that facial attributes are useful in a number of applications such as face recognition and retrieval. However, estimating attributes in images with large variations remains a big challenge. This challenge is addressed in this paper. Unlike existing methods that assume the independence of attributes during their estimation, our approach captures the interdependencies of local regions for each attribute, as well as the high-order correlations between different attributes, which makes it more robust to occlusions and misdetection of face regions. First, we have modeled region interdependencies with a discriminative decision tree, where each node consists of a detector and a classifier trained on a local region. The detector allows us to locate the region, while the classifier determines the presence or absence of an attribute. Second, correlations of attributes and attribute predictors are modeled by organizing all of the decision trees into a large sum-product network (SPN), which is learned by the EM algorithm and yields the most probable explanation (MPE) of the facial attributes in terms of the region's localization and classification. Experimental results on a large data set with $22,400$ *images show the effectiveness of the proposed approach.*

1. Introduction

Visual attributes are properties observable from images, such as "*smiling*" and "*lighting*". They are powerful as high-level representations of images in a variety of tasks, such as object recognition, image-to-text matching, and attribute discovery.

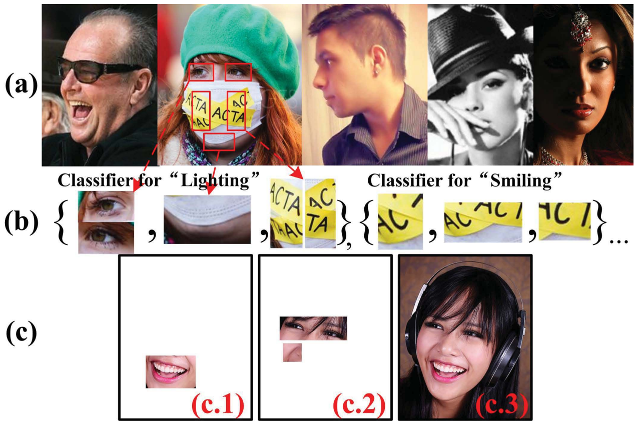


Figure 1. (a) Different kinds of face variations. (b) Conventional methods [12, 11] concatenate low-level features extracted from a few pre-defined functional regions, such as the eyes and chin shown in the second image of (a), and then learn a binary SVM to build a separate classifier for each attribute—this method is unreliable due to feature corruption at the functional regions. (c) Human vision can predict attributes, such as "*smiling*", even with large occlusions. This study intends to bridge this gap between computer vision and human vision.

The usefulness of face attributes has also been demonstrated in the applications of face search [11], ranking [19], and verification [12]. The success of these algorithms relies heavily on the accuracy of predicted attribute values, *i.e*. the scores of separate attribute classifiers. For example, Kumar *et al.* [11] created an image retrieval system in which users can search for face images based on text queries, such as "*young woman smiling*". Images are retrieved by simply ranking the values of face attributes. Recently, Siddiquie *et al.* [19] exploited the cooccurrences of attributes to improve image search performance. Since the query "*young woman smiling*" contains the attribute "*woman*", face images with attribute "*mustache*" will be disregarded, and those with attribute "*blond hair*" are likely to be selected.

Current methods [11, 12] concatenate low-level features extracted from a few pre-defined functional regions, *e.g.* eyes and chin, and then train a separate binary SVM classifier for each attribute. Although they are sufficient for well controlled environments, **the problem** is that if large face variations such as pose, lighting, and occlusion

[*] This work is supported by the General Research Fund sponsored by the Research Grants Council of the Kong Kong SAR (Project No. CUHK 416312 and CUHK 416510) and Guangdong Innovative Research Team Program (No.201001D0104648280).

are present (Fig.1 (a)), the scores of the separate classifiers become unreliable due to feature corruption at the functional regions as in Fig.1 (b). In this case, the attribute values can greatly bias the algorithms built on top of them. In uncontrolled scenarios, faces could be only partially visible for many reasons. Subjects may intentionally wear masks, sunglasses, or makeup to hide their identities. The same happens when cameras cannot capture faces with the optimal viewpoints, subjects occlude each other, or some regions have heavy shadows under lighting conditions. Even if faces are fully visible, some functional regions could be misdetected for various reasons. All the above pose a big challenge for attribute estimation.

Nevertheless, human vision can predict attributes even with the presence of large variations. There are three key **observations**. First, an attribute can be estimated even from small image regions. For instance, one can easily predict a person is "*smiling*" only by the mouth, as shown in Fig.1 (c.1). Second, if a region has been occluded, an attribute can still be inferred by its interdependence with respect to other regions. For example, according to the eyes and the nose-mouth line (c.2), it is reasonable to judge that the person is "*smiling*". Third, the presence of some attributes may indicate the absence or presence of others. For example, given the "*black hair*" and "*black eyes*" of the woman shown at the right, the attribute "*European*" is unlikely to be present.

In this work, we **propose** estimating attributes from face images that may be corrupted. Our method involves the following three steps: 1) automatically discover the discriminative regions for an attribute, 2) explore interdependencies of these regions, and 3) learn correlations of different attributes.

In the first step, we devise a discriminative binary decision tree for each attribute. Each node corresponds to a rectangle region in images, and contains a region detector as well as a regional classifier. The region detector provides reliable localization over a cluttered background and occluded faces, while the regional classifier predicts the attribute. Region discovery at each node can be considered as selecting the most discriminative region separating positive and negative samples.

In the second step, a learned decision tree is transformed to a sum-product tree (SPT) to explore interdependencies among discovered discriminative regions. SPT outputs the likelihood that an attribute will be present, using the scores of all the regional classifiers as input. If a region is misdetected, the output of the corresponding regional classifier can be efficiently marginalized during inference without affecting attribute estimation.

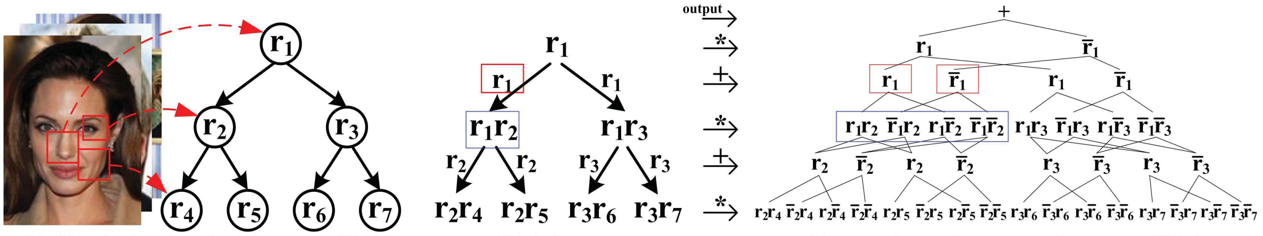
In the third step, we organize all the SPTs into a sum-product network (SPN) that models the joint probability of the SPTs (organzine the regional classifiers in a hierarchical manner) and the attribute labels by stacking many layers of sum and product nodes. As the SPN goes deeper, an exponentially large number of attribute correlations can be compactly encoded. Even for a large and deep architecture, SPN can be efficiently learned by the EM algorithm, and can perform exact inference, such as the most probable explanation (MPE), with which misdetection handling is like answering "probabilistic queries" [5]. We can efficiently derive the probabilities of the presences of attributes conditioned on only a few evidences, *i.e.* $p(\{\textit{presences of any subset of attributes}\}|\{\textit{scores of any subset of region classifiers}\})$.

The key **contributions** of our work can be summarized as follows. We propose a new deep SPN architecture for robust estimation of facial attributes. Our system models attribute correlations using a new network architecture that combines decision trees with SPN, and has some attractive advantages. 1) Discriminative regions for each attribute can be automatically discovered. 2) It does not require occluded training data, which may introduce bias on the distribution of occlusions. The joint distribution of classification scores and attribute labels can be learned from unoccluded training samples. 3) This new network architecture can efficiently compute the correlation between any subsets of attributes, which is not trivial (there are 2^n possible subsets for n attributes). 4) At the test stage, any types of misdetections can be handled by efficiently marginalizing the classification scores of the misdetected regions.

1.1. Related Work

We review several robust deep models to handle data corruption, such as noise and occlusion. These models can be divided into two categories: *implicitly* learning robust features [22, 4, 13], and *explicitly* modeling the structure of occlusions [21, 20, 18, 16, 15, 14]. To learn robust features, Vincent *et al.* [22] extended Hinton's deep autoencoder [8] by randomly corrupting the input of the restricted Boltzmann machine (RBM) at the pre-training stage. Recently, more efforts have been made to explicitly model the structure of noisy images. Tang *et al.* [21] introduced the robust Boltzmann machine, which couples noise estimation and feature learning. It distinguishes corrupted and uncorrupted pixels and finds useful latent representations. Nicolas *et al.* [18] proposed a more sophisticated model to separate appearance and occlusion boundaries of image patches with a field of RBMs.

These methods have certain drawbacks. The denoising autoencoder [22, 4] has a strong assumption that input is corrupted following a known distribution. The subsequent methods [21, 20, 18, 16] overcame this problem by learning noise models from training data. The learned models rely on noisy training samples, but the large noise space cannot be easily covered by the training set. We did not employ


r_1
r_2
r_3
r_4
r_5
r_6
r_7
r_1
r_1
r_1
r_1r_2
r_1r_3
r_2
r_2
r_3
r_3
r_2r_4
r_2r_5
r_3r_6
r_3r_7
output
*
+
*
+
*
+
r_1
$\overline{r}_1$
r_1
$\overline{r}_1$
r_1
$\overline{r}_1$
r_1r_2 $\overline{r}_1r_2$ $r_1\overline{r}_2$ $\overline{r}_1\overline{r}_2$ r_1r_3 $\overline{r}_1r_3$ $r_1\overline{r}_3$ $\overline{r}_1\overline{r}_3$
r_2 $\overline{r}_2$ r_2 $\overline{r}_2$ r_3 $\overline{r}_3$ r_3 $\overline{r}_3$
r_2r_4 $\overline{r}_2r_4$ $r_2\overline{r}_4$ $\overline{r}_2\overline{r}_4$ r_2r_5 $\overline{r}_2r_5$ $r_2\overline{r}_5$ $\overline{r}_2\overline{r}_5$ r_3r_6 $\overline{r}_3r_6$ $r_3\overline{r}_6$ $\overline{r}_3\overline{r}_6$ r_3r_7 $\overline{r}_3r_7$ $r_3\overline{r}_7$ $\overline{r}_3\overline{r}_7$

(a) discriminative decision tree (DDT) **(b) join tree** **(c) transformed sum-product tree (SPT)**

Figure 2. We first select discriminative regions with a discriminative decision tree (DDT) (a). To explore region interdependencies, each DDT is transformed to a sum-product tree (SPT) (c) through the representation of a join tree (b).

these methods since we do not know the prior distribution of noise in face images and our training set does not include noisy or occluded samples. Unlike previous deep models, our work does not need to employ or synthesis training data with noise and can naturally cope with any kinds of corruptions, because the scores of the misdetected region classifiers can be marginalized out.

We adopt the recently proposed SPN [17], which models probability distribution by stacking many layers of sums and products. We extend the traditional SPN by proposing a new architecture with 12 layers for modeling correlations of attributes. Our deep SPN learns the joint distribution of the regional classifiers' scores and the attributes' labels from the images without occlusions, and it can capture correlations between any subset of attributes. During inference, occlusions can be handled by marginalizing out the variables of the undetected regions given the property of SPN. Our SPN is more efficient and robust than the existing SPNs [17, 7], which use pixel values or low-level features as input, because we operate on classifiers' scores.

2. Discriminative Region Discovery

A binary discriminative decision tree (DDT) is learned for each attribute to discover discriminative regions. As shown in Fig.2 (a), each node corresponds to a rectangle region and contains a region detector and a regional classifier. Each node splits data into its children by first scanning images with the region detector and then classifying the attribute with the regional classifier once the region has been located. The split process stops when a pre-defined maximum depth is reached. In this work, the maximum depth is 3 and the number of nodes is $N = 7$. Fig.3 shows the selected discriminative regions of several attributes.

In order to learn DDT, three training sets are prepared: "presence", "absence", and "background". Both the "presence" and "absence" sets contain images with faces, indicating whether the attribute is "on" or "off". The "background" set contains images without faces. Training is conducted in four steps: 1) for each node, randomly sample a number of rectangles with various positions; 2) for each rectangle, use latent-SVM [6] to train the classifier with the "presence" set as positives, and the "absence" set as negatives; 3) at each node, select the rectangle with the maximum information gain [1] as a discriminative region; 4) learn the detector at each node based on the selected discriminative region with linear-SVM, which utilizes both the "presence" and "absence" sets as positive training samples, and the "background" set as negative training samples. Steps 1) and 3) are similar to [25], and step 4) is trivial. The details of step 2) are discussed below.

As shown in Fig.3, as face images are captured in an unconstrained environment, the regions extracted from different images at the same position are not well aligned. Therefore, given a sampled rectangle with its position denoted as $(\overline{x}, \overline{y})$, we train the region classifier r^c with latent-SVM that iterates between locally searching the region position (x, y) in each image and optimizing the parameters α by minimizing the objective function,

$$L(\alpha) = \frac{1}{2}\|\alpha\|^2 + C\sum_{i=1}^{n} \max(0, 1 - \ell_i r^c(I_i^{(x,y)})). \quad (1)$$

$I_i^{(x,y)}$ and $\ell_i \in \{1, -1\}$ indicate the region of the i-th image at position (x, y) and its label. $r^c(I_i^{(x,y)}) = max_{(x,y)\in\{(x,y)|dist((x,y),(\overline{x},\overline{y}))\le\tau\}}\alpha \cdot \Phi(I_i^{(x,y)})$ is a function that determines the region position (x, y) with the maximum classification score, according to the current parameters α. $dist(\cdot, \cdot)$ is the Euclidean distance. (x, y) is searched within radius τ around $(\overline{x}, \overline{y})$. Eq.1 can be efficiently optimized with the stochastic gradient descent [6].

3. Region Interdependencies

Region interdependence is modeled with a sum-product tree (SPT), which is a shallow sum-product network (SPN).

SPN Overview. SPN models joint probabilities and builds on the network polynomial [5], which is a multilinear function of variables' indicators. For example, consider the joint probability $P(A_1, A_2)$ of two binary variables, A_1, A_2, which can be written as a multi-linear function with only sums/products: $P(A_1, A_2) = P(A_1 = 1, A_2 = 1)\mathbf{1}(A_1 = 1)\mathbf{1}(A_2 = 1) + P(A_1 = 0, A_2 = 1)\mathbf{1}(A_1 = 0)\mathbf{1}(A_2 = 1) + P(A_1 = 1, A_2 = 0)\mathbf{1}(A_1 = 1)\mathbf{1}(A_2 = 0) +$

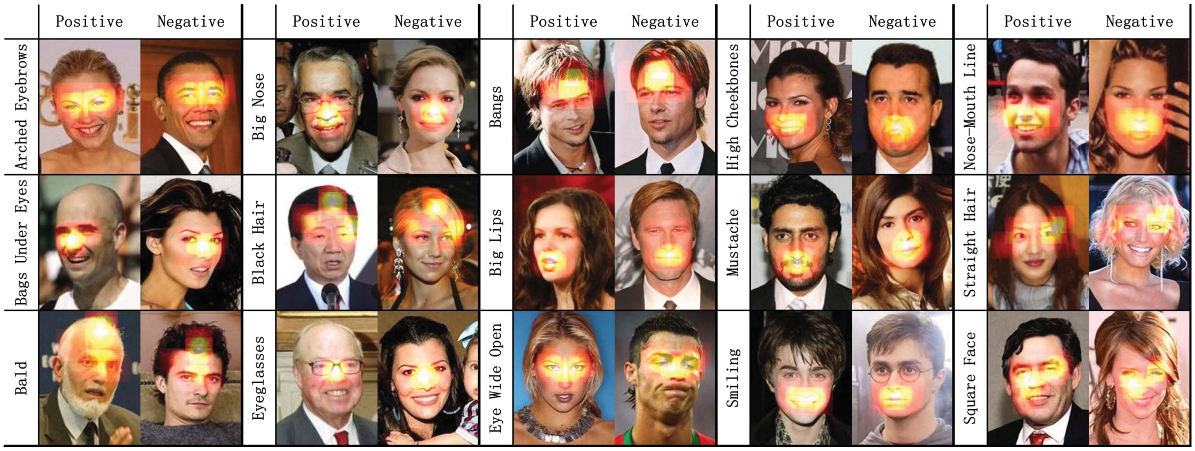


Figure 3. Seven discriminative regions on each image are selected for each attribute and visualized in red. Yellow indicates the area where the seven regions are highly overlapped. Also, a positive example and a negative example are shown for each attribute.

$P(A_1 = 0, A_2 = 0)\mathbf{1}(A_1 = 0)\mathbf{1}(A_2 = 0)$, where $\mathbf{1}(\cdot)$ is the indicator, *i.e.* $\mathbf{1}(x) = 1$ if x is true, otherwise $\mathbf{1}(x) = 0$. We can marginalize a variable, *e.g.* A_1, simply by setting all the indicators related to A_1 as 1 [17]. The above polynomial then becomes $P(A_2) = P(A_2 = 1)\mathbf{1}(A_2 = 1) + P(A_2 = 0)\mathbf{1}(A_2 = 0)$. Therefore, computing any conditional probability, *e.g.* $P(A_1|A_2) = \frac{P(A_1, A_2)}{P(A_2)}$, becomes easy. However, the network polynomial has a size exponential in the number of variables, *e.g.* 2^n for n binary variables.

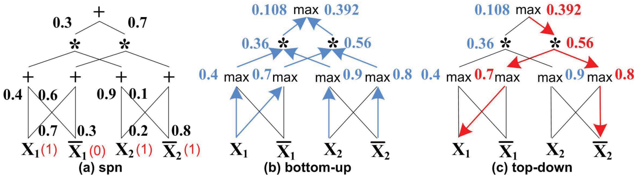


Figure 4. Example of SPN.

SPN compactly represents the network polynomial in a hierarchical manner. In the simple example shown in Fig.4 (a), SPN models the joint probability of two binary variables, X_1 and X_2, as a rooted acyclic graph with terminal, sum, and product nodes. We denote indicators $\mathbf{1}(X_i = 1), \mathbf{1}(X_i = 0)$ by $X_i, \overline{X_i}$. The terminals are the indicators $X_1, \overline{X_1}, X_2, \overline{X_2}$. The sum and product nodes are arranged in alternating layers: all the children of a product node are sums, and all the children of a sum node are products or terminals. An edge (v^+, v^*) that connects a sum node v^+ with its child v^* has weight $w_{v^+v^*} > 0$, and all the weights of its children $\sum_{\forall v^* \in Ch(v^+)} w_{v^+v^*} = 1$, where $Ch(v^+)$ is the set of children of v^+. An edge (v^*, v^+) that connects a product node v^* with its child $v^+ \in Ch(v^*)$ has uniform weight $w_{v^*v^+} = 1$. With this representation, the value of a sum node v^+, denoted as S_{v^+}, can be recursively derived as $S_{v^+} = \sum_{\forall v^* \in Ch(v^+)} w_{v^+v^*} S_{v^*}$, and $S_{v^*} = \prod_{v^+ \in Ch(v^*)} S_{v^+}$. Note that the value of the root equals the joint probability of the variables [17], *i.e.* $S(X_1, \overline{X_1}, X_2, \overline{X_2}) = P(X_1, X_2)$.

Fig.4 (a) illustrates an example when $X_1 = 1$ and X_2 is unobserved. Thus, by marginalizing X_2, $S(1, 0, 1, 1) = P(X_1 = 1) = 0.3 \times 0.4 + 0.7 \times 0.7 = 0.61$. With SPN, we can efficiently infer the value of an unobserved variable using the MPE inference [17], which has linear complexity in the number of nodes and is tractable in polynomial time. We first replace the sums with maximizations, as shown in Fig.4 (b), which means that the product node with maximum value is selected during inference. The MPE contains two passes. The first pass computes the value at each node from *bottom to top* (b), and then the second pass backtracks the path of the first pass from *top to bottom* (c) until it reaches the terminals. For instance, with $X_1, \overline{X_1}, X_2, \overline{X_2}$ being $1, 0, 1, 1$, the most probable explanation of X_2 when $X_1 = 1$ is 0. This is why we can estimate the occluded attributes conditioning on the observed ones.

DDT to SPT. Our SPT considers the scores of the regional classifiers $\{r_i\}$ in Eq.1 as terminals. To model region interdependencies and simultaneously maintain the discriminative power of DDT, we transform a DDT to SPT, which can better handle misdetection. If a region is not detectable, DDT must guess the output of the regional classifier at the misdetected node, while SPT can estimate the attribute by marginalizing the variables of the undetected regions.

A decision tree can be transformed to a sum-product structure with the join tree using the algorithm proposed in [10, 5]. A join tree is an undirected tree on which each node is a set of variables, called a *cluster*, and each edge is labeled with the intersection of the adjacent clusters, called a *separator*. The join tree of our decision tree is illustrated in Fig.2 (b). The sum-product structure is then determined by two types of mappings: 1) each instantiation of a cluster is mapped to a product node; and 2) each instantiation of a separator is mapped to a sum node.

For instance, as shown in Fig.2 (b) and (c), cluster $r_1 r_2$ (blue) has four instantiations, $r_1 r_2, \overline{r_1} r_2, r_1 \overline{r_2}, \overline{r_1}\,\overline{r_2}$, and is transformed to four products, while the separator r_1 (red) with two instantiations $r_1, \overline{r_1}$ is mapped to two sums. In Fig.2 (c), the SPT is obtained by replacing the instantiations with sum/product nodes.

4. Attribute Correlations with Deep SPN

4.1. Our Architecture

Our deep SPN jointly models the outputs of regional classifiers and the labels of attributes. It has two types of terminal nodes (Fig.5). 1) SPTs, denoted as $\{T_i\}_{i=1}^{K}$, contain terminals with *continuous* values, $\mathcal{R} = \{r_{ij}\}_{i=1,j=1}^{K,N}$, each of which indicates the classification score of the j-th regional classifier of the i-th attribute. 2) $\mathcal{A} = \{(A_i, \overline{A_i})\}_{i=1}^{K}$ are the *binary* indicators, each of which indicates the label of an attribute. $A_i = 1$ and $\overline{A_i} = 0$ denote that the i-th attribute is present; otherwise, $A_i = 0$ and $\overline{A_i} = 1$.

In the following, we denote the name of an attribute with bold $\mathbf{A_i}$ and its binary indicator with $A_i, \overline{A_i}$. Our network is shown in Fig.5, where different sets of nodes (in dashed lines) model correlations of different groups of attributes (in bold type). A set of product nodes represent the instantiations for an attribute group, *e.g*. a node in layer 4 can be viewed as $\overline{A_1} A_2 \overline{A_3} A_4$ for group $\mathbf{A_1 A_2 A_3 A_4}$ in layer 5. A set of sum nodes represents the possible correlations among a group of attributes, *e.g*. there are 3 possible correlations (3 sum nodes) of $\mathbf{A_1 A_2 A_3 A_4}$ in layer 5, each being the weighted sum of the product nodes (instantiations) in layer 4. Note that the number of possible correlations, *i.e*. the number of sum nodes of each attribute group, is an empirical parameter. Once it has been set, the number of product nodes for an attribute group is determined. In Fig.5, since both groups $\mathbf{A_1 A_2}$ and $\mathbf{A_3 A_4}$ have 3 sum nodes, there are $3 \times 3 = 9$ products in layer 4. In our experiment, we set the number of sum nodes in each group as a constant 20. Intuitively, a larger number of sums result in larger number of products, which means they have stronger representative power.

Our SPN is reconstructed as follows. 1) In layer 1, we treat the SPTs and the indicators of *two* attributes as one group, *e.g*. the group in blue in Fig.5. 2) In layer 2, a product node connects to *all* the sum nodes of a group in layer 1 if they represent the correlations of the same set of attributes; 3) In the upper layer, a sum node connects to *all* the products belonging to the same group of attributes, *e.g*. each sum of $\mathbf{A_1 A_2}$ in red connects to all the products in green, since they all relate to the correlations of attributes $\mathbf{A_1}$ and $\mathbf{A_2}$. A product node connects to *two* sum nodes to form the instantiations of an upper attribute group.

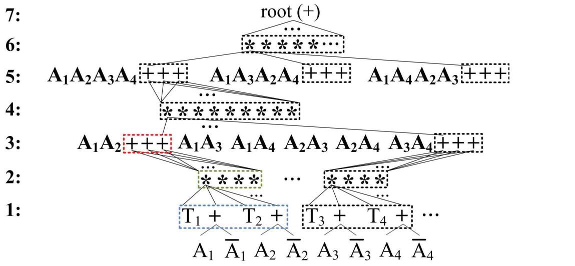


Figure 5. In our SPN, different attribute correlations are modeled by different groups of nodes, *e.g*. group $\mathbf{A_1 A_2 A_3 A_4}$ represents correlation among the first four attributes. When more layers are added, an exponentially large number of attribute correlations can be compactly encoded. T_i represents a SPT and it has $K \times N$ terminals $\mathcal{R} = \{r_{ij}\}_{i=1,j=1}^{K,N}$.

Inference. Given an image, we first run the region detectors with the sliding window scheme. If a region is located, apply the corresponding regional classifier. Let $\mathcal{R}^d$, $\mathcal{R}^{ud}$, and $\mathcal{A}$ denote the continuous terminals, *i.e*. classification scores of the detected and undetected regions, and the binary terminals respectively. Inferring the values of attributes is equivalent to maximize the posterior with the learned SPN,

$$\begin{aligned}\{\mathcal{A}^*, \mathcal{R}^{ud*}\} =& \arg\max\nolimits_{\mathcal{A},\mathcal{R}^{ud}} P(\mathcal{A}, \mathcal{R}^{ud} | \mathcal{R}^d) \\ =& \arg\max\nolimits_{\mathcal{A},\mathcal{R}^{ud}} \frac{S(\{\mathcal{A}, \mathcal{R}^{ud}, \mathcal{R}^d\})}{S(\mathcal{R}^d)_{\mathcal{A}=1,\mathcal{R}^{ud}=1}}.\end{aligned} \qquad (2)$$

$\mathcal{A}$ and $\mathcal{R}^{ud}$ are unobserved variables. $\mathcal{R}^d$ are observed variables. $S(\cdot)$ is the value of the root. Note that $S(\mathcal{R}^d)_{\mathcal{A}=1,\mathcal{R}^{ud}=1}$ is a constant computed by marginalizing $\mathcal{A}$ and $\mathcal{R}^{ud}$, *i.e*. setting the corresponding indicators to 1. Eq.2 can be efficiently optimized with the most probable explanation (MPE) introduced in Sec.3. The only difference is that we are dealing with a more sophisticated network.

4.2. Learning the Deep SPN

The learning algorithm is summarized in Alg.1. Given a set of training images $\{I_i\}_{i=1}^{n}$, we first obtain $\mathcal{R}_i^d$, $\mathcal{R}_i^{ud}$, and $\mathcal{A}_i$ from I_i. As occluded images are not required for training, $\mathcal{R}_i^{ud}$ could be empty. The training process iterates between learning weights $\mathbf{w}$ while keeping the network architecture fixed and pruning edges with zero weights.

Learning weights w. If $v_i^* \in Ch(v^+)$ is a child (product) of a sum node v^+, let $P(v_i^* | \mathcal{R}^d)$ be the probability of choosing the i-th child v_i^* at sum node v^+, conditioned on the observables $\mathcal{R}^d$. SPN can be learned with an EM-like algorithm following [5]. The *E-step* computes $P(v_i^* | \mathcal{R}^d)$, which can be viewed as the weights at each sum node, indicating which child to chose during inference. The *M-step* adds all these probabilities and renormalizes them to update the weights. Specifically, at the *E-step*, we use the MPE inference as in Sec.4.1 to select the child of a sum node. At the *M-step*, we maintain a count for each of its child, and increment the count if a child has been chosen in an iteration during training. Then the weights at each sum

Algorithm 1: Learning Deep SPN

```
Input: All the training images $\{I_i\}$ and the labels of 73 attributes
Output: $\mathbf{w}$ and network architecture
1. train DDT for each attribute as described in Sec.2
2. transform DDT to SPT by bi-breadth-first search in Sec.3
3. obtain $\{\mathcal{R}_i^d, \mathcal{R}_i^{ud}, \mathcal{A}_i\}$ from $\{I_i\}$, and initialize $\mathbf{w}$ and network
architecture as described in Sec.4.2
  repeat
    -Update $\mathbf{w}$
      E-step: infer $\mathcal{R}_i^{ud}, \mathcal{A}_i$ according to Eq.2, Sec.4.1
      M-step: renormalize $\mathbf{w}$
    -Prune edges with zero weights
  until converge
```

node are updated by normalizing the counts over all of its children. A similar strategy was used in [17, 7].

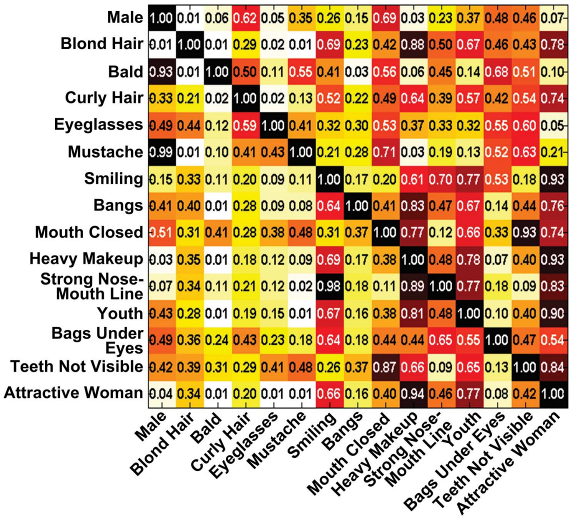
Initializing network architecture. We initialize the network following Sec.4.1. Each group at an upper layer represents the combination of two groups at the lower layer. Our goal is to obtain a compact network structure, *i.e*. to determine which two groups should be combined, because there are exponentially large numbers of combinations when SPN goes deeper. As we have 73 attributes in our experiment, layer 3 in Fig.5 may contain $\frac{73!}{(73-2)!2!} = 2628$ groups, each indicating correlation of two attributes. If we consider full combinations all the time, when more layers are added, learning SPN becomes impractical due to huge computational cost. For example, layer 5 will have $\frac{2628!}{(2628-2)!2!} = 3,451,878$ groups.

To ensure compactness as well as combination diversity, we introduce two constraints at layer 3 and the above layers: 1) a group in the previous layer can be combined only once; and 2) two groups with no intersection have higher priority to be combined. We adopt greedy search to find group combinations. Considering the groups at layer 3 of Fig.5 in sequence, we first search for $\mathbf{A_3A_4}$ to be combined with $\mathbf{A_1A_2}$, because they do not have overlap in attributes. We then consider $\mathbf{A_1A_2A_3A_4}$ as a new group at layer 5. And $\mathbf{A_1A_2}$ can no longer be combined at the next iteration. This strategy will reduce the number of groups at each layer by half at a time, while retaining the diversity of attribute combinations. Our deep SPN has 12 layers in total.

5. Experiments

Datasets. The previous works [11, 12] evaluated their methods on the subsets of FaceTracer and PubFig, but did not release the subsets selected. Therefore, it is impossible to directly compare with their published results on those datasets. To achieve a fair and extensive comparison, we construct a large dataset that is a composition of LFW [9], FaceTracer [11], and PubFig [12], and implement state-of-the-art methods for comparison. The face images are taken in uncontrolled environments with large variations in poses, lightings, expressions, and camera settings. $10,000$ unoccluded images are selected for training and another $10,000$ unoccluded images and $2,400$ occluded images (some occluded examples are shown in Fig.6) are selected for testing. These images were annotated by a professional image labeling company. An image was labeled by one subject because the data scale. Face images are roughly aligned based on the positions of eyes, cropped and normalized to 200×160.

Figure 6. Some images with occlusions in Experiment II.



	Male	Blond Hair	Bald	Curly Hair	Eyeglasses	Mustache	Smiling	Bangs	Mouth Closed	Heavy Makeup	Strong Nose-Mouth Line	Youth	Bags Under Eyes	Teeth Not Visible	Attractive Woman
Male	1.00	0.01	0.06	0.62	0.05	0.35	0.26	0.15	0.69	0.03	0.23	0.37	0.48	0.46	0.07
Blond Hair	0.01	1.00	0.01	0.29	0.02	0.01	0.69	0.23	0.42	0.88	0.50	0.67	0.46	0.43	0.78
Bald	0.93	0.01	1.00	0.50	0.11	0.55	0.41	0.03	0.56	0.06	0.45	0.14	0.68	0.51	0.10
Curly Hair	0.33	0.21	0.02	1.00	0.02	0.13	0.52	0.22	0.49	0.64	0.39	0.57	0.42	0.54	0.74
Eyeglasses	0.49	0.44	0.12	0.59	1.00	0.41	0.32	0.30	0.53	0.37	0.33	0.32	0.55	0.60	0.05
Mustache	0.99	0.01	0.10	0.41	0.43	1.00	0.21	0.28	0.71	0.03	0.19	0.13	0.52	0.63	0.21
Smiling	0.15	0.33	0.11	0.20	0.09	0.11	1.00	0.17	0.20	0.61	0.70	0.77	0.53	0.18	0.93
Bangs	0.41	0.40	0.01	0.28	0.09	0.08	0.64	1.00	0.41	0.83	0.47	0.67	0.14	0.44	0.76
Mouth Closed	0.51	0.31	0.41	0.28	0.38	0.48	0.31	0.37	1.00	0.77	0.12	0.66	0.33	0.93	0.74
Heavy Makeup	0.03	0.35	0.01	0.18	0.12	0.09	0.69	0.17	0.38	1.00	0.48	0.78	0.07	0.40	0.93
Strong Nose-Mouth Line	0.07	0.34	0.11	0.21	0.12	0.02	0.98	0.18	0.11	0.89	1.00	0.77	0.18	0.09	0.83
Youth	0.43	0.28	0.01	0.19	0.15	0.01	0.67	0.16	0.38	0.81	0.48	1.00	0.10	0.40	0.90
Bags Under Eyes	0.49	0.36	0.24	0.43	0.23	0.18	0.64	0.18	0.44	0.44	0.65	0.55	1.00	0.47	0.54
Teeth Not Visible	0.42	0.39	0.31	0.29	0.41	0.48	0.26	0.37	0.87	0.66	0.09	0.65	0.13	1.00	0.84
Attractive Woman	0.04	0.34	0.01	0.20	0.01	0.01	0.66	0.16	0.40	0.94	0.46	0.77	0.08	0.42	1.00

Figure 7. The pairwise correlations of 15 attributes.

I. Learning Attribute Correlations. We use the learned SPN to compute the correlation between two attributes as the conditional probability of one when the other is present in a way as described in Sec.3. Note that the correlation between any two sets of attributes can be computed in the same way. The correlation value is in the range of $[0, 1]$. "0", "0.5", and "1" indicate "negative correlation", "independence", and "positive correlation", respectively. Fig.7 plots the pairwise correlations among 15 attributes, where an element in this matrix represents the probability of the attribute at the j-th column when the attribute at the i-th row is present. Some interesting results are observed. For example, "*attractive woman*" has highly positive correlation with "*smiling*" and "*heavy makeup*", but is negatively correlated with "*bald*" and "*mustache*". The experiment results show that attribute correlations can be well discovered by the proposed deep SPN.

II. Attribute Estimation. Our deep SPN is compared

	Male	Asian	White	Black	Baby	Child	Youth	Mid. Aged	Senior	Black Hair	Blond Hair	Brown Hair	Bald	No Eyewear	Eyeglasses	Sunglasses	Mustache	Smiling	Frowning	Chubby	Blurry	Harsh Light.	Flash	Soft Light.	Outdoor
SAC	79	**93**	89	84	83	77	**86**	81	86	82	**81**	73	80	87	90	89	81	**92**	**90**	83	89	79	74	65	**86**
MAC	**92**	**93**	86	90	**95**	81	82	84	**96**	81	68	74	87	**90**	**97**	95	**97**	91	**90**	92	88	**80**	**75**	46	84
Deep SPN	**92**	91	**90**	**98**	94	**86**	83	**96**	95	**86**	78	**76**	**96**	**90**	93	**98**	95	91	**90**	**96**	**97**	77	73	**69**	85
SAC	59	74	62	65	75	61	70	66	77	60	62	55	65	70	74	72	70	78	82	69	77	68	59	50	82
MAC	73	77	**75**	73	76	75	69	**84**	**83**	63	62	**64**	72	72	75	**88**	71	72	75	72	**82**	73	66	**51**	81
Deep SPN	**77**	**82**	**75**	**79**	**82**	**83**	**79**	81	82	**78**	**66**	62	**78**	**79**	**86**	**88**	**90**	**84**	**90**	**86**	80	**82**	**88**	48	**91**
	Curly Hair	Wavy Hair	Straight Hair	Reced. Hairline	Bangs	Sideburns	F. V. Forehead	P. V. Forehead	Obs. Forehead	Bushy Eyebrows	Arch. Eyebrows	Narrow Eyes	Eyes Open	Big Nose	Pointy Nose	Big Lips	Mouth Closed	Mouth S. O.	Mouth W. O.	Teeth N. V.	No Beard	Goatee	Round Jaw	Double Chin	Wearing Hat
SAC	77	69	84	87	85	75	**88**	**76**	82	87	66	74	90	69	**90**	**77**	80	81	**89**	83	80	87	90	86	73
MAC	78	**87**	80	**93**	79	92	73	64	**89**	**90**	76	82	89	**78**	88	70	**88**	**86**	87	89	92	95	**91**	**93**	87
Deep SPN	**79**	**87**	**85**	91	**94**	**97**	**88**	74	88	87	**82**	**86**	**92**	75	87	75	80	84	87	**95**	**94**	**99**	89	90	**95**
SAC	61	56	70	73	74	69	77	**68**	71	73	54	59	68	60	77	61	55	63	70	72	80	78	79	71	61
MAC	60	62	75	**83**	72	66	80	65	72	70	62	66	75	**71**	76	63	**76**	69	71	74	79	74	82	77	69
Deep SPN	**73**	**80**	**79**	82	**85**	**75**	**84**	**68**	**84**	**79**	**70**	**70**	**78**	**71**	**83**	**68**	75	**77**	**79**	**88**	**89**	**91**	**84**	**85**	**79**
	Oval Face	Square Face	Round Face	Color Photo	Posed Photo	Attr. Man	Attr. Woman	Indian	Gray Hair	Bags Un. Eyes	Heavy Makeup	Rosy Cheeks	Shiny Skin	Pale Skin	5 Shadow	Strong N. Lines	Wear. Lipstick	Flushed Face	H. Cheekbones	Brown Eyes	Wear. Earrings	Wear. Necktie	Wear. Necklace		
SAC	83	83	77	87	78	**96**	87	90	85	89	91	**86**	78	**83**	87	83	55	62	**89**	**70**	83	88	83		
MAC	**88**	**98**	**86**	94	**87**	91	93	93	**97**	**92**	91	54	74	68	97	**88**	**96**	42	88	66	88	90	**87**		
Deep SPN	82	92	85	**95**	**87**	92	**94**	**96**	90	87	**97**	80	**79**	80	**98**	82	92	**67**	85	**70**	**98**	**95**	80		
SAC	59	69	70	71	66	72	71	73	69	69	85	70	64	62	65	65	50	60	67	61	68	82	79		
MAC	68	70	75	74	70	68	69	72	77	78	84	54	70	67	77	72	67	41	81	60	81	85	**86**		
Deep SPN	**80**	**88**	**79**	**82**	**81**	**84**	**85**	**88**	**80**	**82**	**92**	**77**	**73**	**82**	**88**	**78**	**84**	**70**	**86**	**78**	**88**	**90**	85		

Table 1. Classification accuracies of deep SPN, SAC [12, 11], and MAC [2]. The first three rows are on the unoccluded test set, and the last three are on the occluded test set.

with the separate attribute classifiers (SAC) [12, 11] and the multilevel attribute classifiers (MAC) [2] on both the unoccluded and occluded testing images. The SAC trains a binary SVM classifier for each attribute. The original MAC is to predict human attribute based on poselets. We apply it to faces by replacing the poselets with our discriminative regions in Sec.2 and then train different level of classifiers as in [2]. Note that MAC can model pairwise attribute correlations.

The results are reported in Table 1. On the unoccluded test, the averaged classification accuracies of SAC, MAC, and deep SPN are 82.1%, 84.4%, and 87.9%, respectively, while the corresponding accuracies on the occluded test set are 68.0%, 72.1%, and 79.8%. The deep SPN generally performs better especially with the presence of occlusions, because it effectively models the high-order correlations between attributes and regional classifiers. On the occluded test set, the deep SPN significantly outperforms SAC and MAC on nearly all the attributes, because SAC assumes independence between attributes, while MAC can only model pairwise correlation. We also examine our SPTs alone. The averaged accuracies are 81.9% and 70.2% of the unoccluded and occluded tests, respectively, which indicate their performances can be significantly improved when organizing them into the deep SPN. Moreover, it takes 2 hours to train our deep SPN on two 3.3 GHz CPUs and 16G RAMs, and 5.3 seconds to classify 73 attributes of an image, which is comparable to 5 seconds of MAC in the same experimental environment.

III. Experiments on synthetic datasets. We evaluate the robustness of deep SPN under different levels of occlusions on a synthetic dataset. We use the same training and unoccluded test sets as in parts I and II. However, for each sample in the test set, a series of corrupted images are generated with the following strategy. We first generate random noise that covers the whole image, and then gradually reduce the coverage of noise towards one of the four directions: "Top to Bottom", "Bottom to Top", "Left to Right", and "Right to Left". An example of generating occluded images along the "Right to Left" direction is shown in Fig.8.

Fig.8 plots the classification accuracies of the deep SPN when the percentage of visible image region is increased from 10% to 80%. At 50% visibility, all the accuracies are over 75% (random guess is 50%). This is because with our deep model, only a few attributes need to be observed in order to infer the others. As plotted at the left of Fig.9, we average the accuracies of the four directions and compare with SAC and MAC. Our method clearly shows superior performance, showing that our method is more robust when large occlusions are present.

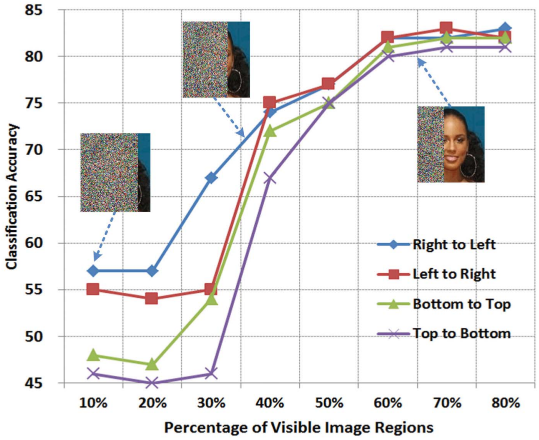


Figure 8. Classification accuracies of deep SPN under different levels of occlusions.

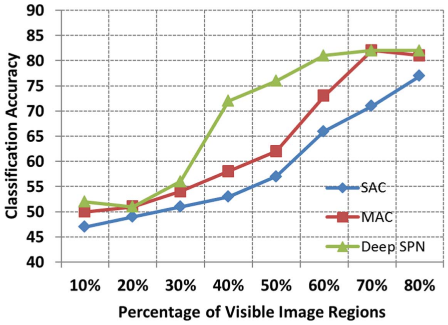


Figure 9. We compare our deep SPN with SAC [12] and MAC [2] under different levels of occlusions.

6. Conclusion

This paper has proposed a facial attribute estimation framework, where attribute estimation is achieved by combining discriminative decision trees with a SPN. For robust attribute estimation, we devise a deep architecture to capture attribute correlations for the SPN, where occlusion handling is casted as marginalizing the variables of the undetected regions. Experimental results on both occluded and unoccluded images show great improvement and indicate that our method is very robust under different levels of occlusions. As face attributes are becoming more important for face recognition, in the future work, we will explore the advantage to combine face attributes with traditional face recognition approaches [23, 24, 3, 26].

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. *ICCV*, 2007.

[2] L. Bourdeva, S. Maji, and J. Malik. Describing people: Poselet-based approach to attribute classification. *ICCV*, 2011.

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. *CVPR*, 2010.

[4] M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *ICML*, 2012.

[5] A. Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM*, 2003.

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010.

[7] R. Gens and P. Domingos. Discriminative learning of sum-product networks. *NIPS*, 2012.

[8] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report 07-49*, 2007.

[10] F. V. Jensena, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 1990.

[11] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A search engine for large collections of images with faces. *ECCV*, pages 340–353, 2008.

[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simle classifers for face verification. *ICCV*, 2009.

[13] Y. LeCun, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[14] P. Luo, X. Wang, and X. Tang. Hierachical face parsing via deep learning. *CVPR*, 2012.

[15] P. Luo, X. Wang, and X. Tang. Pedestrian parsing via deep decompositional network. *ICCV*, 2013.

[16] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. *CVPR*, 2012.

[17] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. *UAI*, 2011.

[18] N. L. Roux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 2011.

[19] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. *CVPR*, 2011.

[20] Y. Tang. Gated boltzmann machine for recognition under occlusion. *NIPS Workshop on Transfer Learning by Learning Rich Generative Models*, 2010.

[21] Y. Tang, R. Salakhutdinov, and G. Hinton. Robust boltzmann machines for recognition and denoising. *CVPR*, 2012.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

[23] X. Wang and X. Tang. A unified framework for subspace face recognition. *TPAMI*, 2004.

[24] X. Wang and X. Tang. Random sampling for subspace face recognition. *IJCV*, 2006.

[25] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. *CVPR*, 2011.

[26] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. *CVPR*, 2011.