Chapter 9
Visual Attributes for Fashion Analytics

Si Liu, Lisa M. Brown, Qiang Chen, Junshi Huang, Luoqi Liu and Shuicheng Yan

Abstract In this chapter, we describe methods that leverage clothing and facial attributes as mid-level features for fashion recommendation and retrieval. We introduce a system called *Magic Closet* for recommending clothing for different occasions, and a system called *Beauty E-Expert* for hairstyle and facial makeup recommendation. For fashion retrieval, we describe a cross-domain clothing retrieval system, which receives as input a user photo of a particular clothing item taken in unconstrained conditions, and retrieves the exact same or similar item from online shopping catalogs. In each of these systems, we show the value of attribute-guided learning and describe approaches to transfer semantic concepts from large-scale uncluttered annotated data to challenging real-world imagery.

S. Liu (✉)
State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, China
e-mail: liusi@iie.ac.cn

L.M. Brown
IBM T. J. Watson Research Center, New York, England
e-mail: lisabr@us.ibm.com

Q. Chen · J. Huang · L. Liu · S. Yan
Qihoo 360 Artificial Intelligence Institute, Beijing, China
e-mail: chenqiang-iri@360.cn

J. Huang
e-mail: huangjunshi@360.cn

L. Liu
e-mail: llq667@gmail.com

S. Yan
e-mail: yanshuicheng@360.cn

R.S. Feris et al. (eds.), *Visual Attributes*, Advances in Computer Vision
and Pattern Recognition, DOI 10.1007/978-3-319-50077-5_9

9.1 Motivation and Related Work

Visual analysis of people, in particular the extraction of facial and clothing attributes [5, 6, 14, 37], is a topic that has received increasing attention in recent years by the computer vision community. The task of predicting fine-grained facial attributes has proven effective in a variety of application domains, such as content-based image retrieval [16], and person search based on textual descriptions [9, 35]. We refer to Chap. 8 for a detailed analysis of methods for processing facial attributes.

Regarding the automated analysis of clothing images, several methods have been proposed for context-aided people identification [10], fashion style recognition [13], occupation recognition [32], and social tribe prediction [17]. Clothing parsing methods, which produce semantic labels for each pixel in the input image, have also received significant attention in the past few years [20, 21, 26, 27, 38]. In the surveillance domain, matching clothing images across cameras is a core subtask for the well-known person reidentification problem [18, 31].

In this chapter, we demonstrate the effectiveness of clothing and facial attributes as mid-level features for fashion analytics and retail applications. This is an important area due to the accelerated growth of e-commerce applications and their enormous financial potential.

Within this application domain, several recent methods have successfully used visual attributes for product retrieval and search. Berg et al. [2] discover attributes of accessories such as shoes and hand bags by mining text and image data from the Internet. Liu et al. [24] describe a system for retrieving clothing items from online shopping catalogs. Kovashka et al. [15] developed a system called "Whittle-Search", which is able to answer queries such as "Show me shoe images like these, but sportier". They used the concept of relative attributes proposed by Parikh and Grauman [29] for relevance feedback. More details about this system is described in Chap. 5. Attributes for clothing have been explored in several recent papers [3–5]. They allow users to search visual content based on fine-grained descriptions, such as a "blue striped polo-style shirt".

Attribute-based representations have also shown compelling results for matching images of people across domains [19, 31]. The work by Donahue and Grauman [7] demonstrates that richer supervision conveying annotator rationales based on visual attributes improves recognition performance. Sharmanska et al. [30] explored attributes and rationales as a form of privileged information [34], considering a learning to rank framework. Along this direction, in one of the applications considered in this chapter, we show that cross-domain image retrieval can benefit from feature learning that simultaneously optimizes a loss function that takes into account visual similarity and attribute classification.

Next, we will describe how visual attributes can serve as a powerful image representation for fashion recommendation and retrieval. We start by describing two systems for clothing and makeup recommendation, respectively, and then show an attribute-guided learning method for cross-domain clothing retrieval.

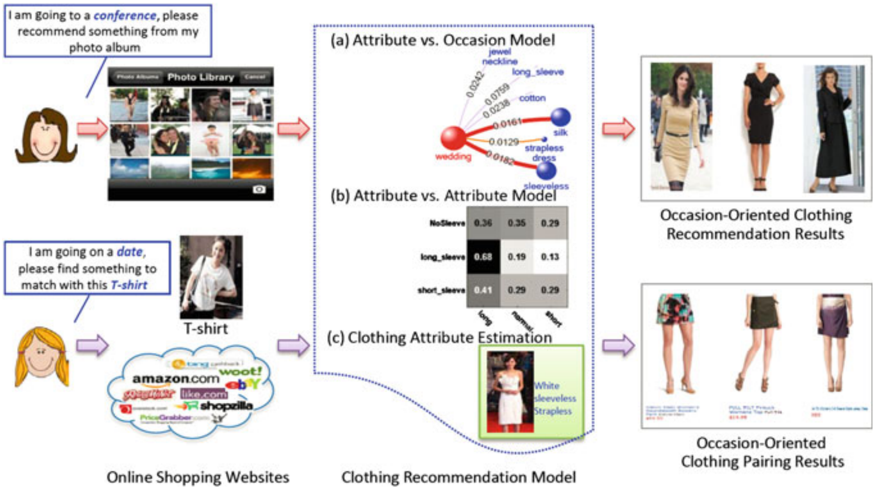

I am going to a conference, please recommend something from my photo album
Photo Albums
Photo Library
Cancel
(a) Attribute vs. Occasion Model
jewel neckline
long_sleeve
cotton
silk
strapless dress
sleeveless
wedding
0.0242
0.0759
0.0238
0.0161
0.0129
0.0182
(b) Attribute vs. Attribute Model
NoSleeve 0.36 0.35 0.29
long_sleeve 0.68 0.19 0.13
short_sleeve 0.41 0.29 0.29
long
normal
short
(c) Clothing Attribute Estimation
White
sleeveless
Strapless
Occasion-Oriented Clothing Recommendation Results
I am going on a date, please find something to match with this T-shirt
T-shirt
amazon.com
woot!
ebY
like.com
shopzilla
PriceGrabber
Online Shopping Websites
Clothing Recommendation Model
Occasion-Oriented Clothing Pairing Results

Fig. 9.1 Two typical clothing recommendation scenarios for Magic Closet. (*Top panel*) Clothing suggestion: given an occasion, the most suitable clothing combinations from the user's photo album are suggested. (*Bottom panel*) Clothing pairing: given an occasion and a reference clothing item, the clothing most suitable for the occasion and most matched with the reference clothing is recommended from online shopping websites

9.2 Recommendation Systems

In this section, we describe two recommendation systems based on attribute prediction. In both cases, we use attributes as an intermediate representation to leverage semantic knowledge from a large expert database. In the first case, we detail a system to recommend clothing from a user's collection for a given special occasion such as a wedding, funeral or conference. We construct a latent SVM model where each potential function in the latent SVM is defined specifically for the clothing recommendation task. We use low-level visual features to predict intermediate clothing attributes such as color, pattern, material, or collar type, which in turn are used to predict the best choice of outfit for the given occasion from the user's closet or from online shopping stores.

In the second case, we develop a system to recommend hairstyle and makeup selections for a user's image without makeup and with either short or bound hair. Again, we use visual features to predict intermediate attribute features for this task. In this scenario, we use a multiple tree-structured super-graphs model to estimate facial/clothing attributes such as a high forehead, flat nose bridge, or collar shape, which in turn are used to predict the most suitable high-level beauty attributes such as hair length or color, lip gloss color or the eye shadow template class.

9.2.1 "Hi, magic closet, tell me what to wear!"

Problem: Only a few existing works target the clothing recommendation task. Some online websites[1] can support the service of recommending the most suitable clothing for an occasion. However, their recommendation tools are mainly based on dress codes and common sense. Magic Closet is the first system to automatically investigate the task of occasion-oriented clothing recommendation and clothing pairing by mining the matching rules among semantic attributes from real images.

Magic Closet mainly addresses two clothing recommendation scenarios. The first scenario is *clothing suggestion*. As shown in the top panel of Fig. 9.1, a user specifies an occasion and the system will suggest the most suitable outfits from the user's own photo album. The second scenario is *clothing pairing*. As shown in the bottom panel of Fig. 9.1, a user inputs an occasion and one reference clothing item (such as a T-shirt the user wants to pair), and then the most matched clothing from the online shopping website is returned (such as a skirt). The returned clothing should aesthetically pair with the reference clothing well and also be suitable for the specified occasion. As a result, the Magic Closet system can serve as a plug-in application in any online shopping website for shopping recommendation.

Two key principles are considered when designing Magic Closet. One is *wearing properly*. Wearing properly means putting on some suitable clothing, which conforms to normative *dress codes*[2] and common sense. The other is *wearing aesthetically*. There are some aesthetic rules which need to be followed when one pairs the upper body clothing and lower body clothing. For example, it looks weird to wear a red coat and a green pants together.

Recommendation Model: In the model learning process, to narrow the semantic gap between the low-level visual features of clothing and the high-level occasion categories, we propose to utilize mid-level clothing attributes. Here 7 multivalue clothing attributes are defined, including the category attribute (e.g., "jeans", "skirts") and detail attributes, describing certain properties of clothing (e.g., color, pattern).

We propose to learn the clothing recommendation model through a unified latent Support Vector Machine (SVM) framework [23]. The model integrates four potentials: (1) visual features versus attribute, (2) visual features versus occasion, (3) attributes versus occasion, and (4) attribute versus attribute. Here the first three potentials relate to clothing-occasion matching and the last one describes the clothing-clothing matching. Embedding these matching rules into the latent SVM model explicitly ensures that the recommended clothing satisfies the requirement of *wearing properly* and *wearing aesthetically* simultaneously.

A training clothing image is denoted as a tuple $(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, \mathbf{o})$. Here $\mathbf{x}$ corresponds to the visual features from the whole body clothing, which is formed by directly concatenating the upper body clothing feature $\mathbf{x}_u$ and lower body clothing feature $\mathbf{x}_l$, namely $\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_l]$. We extract 5 types of features from 20 upper body parts and 10 lower body parts detected using the methodology in [39]. The features include

[1] http://www.dresscodeguide.com/.

[2] Dress codes are written and unwritten rules with regards to clothing.

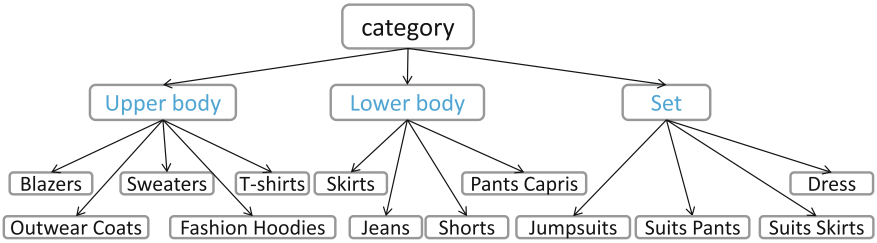
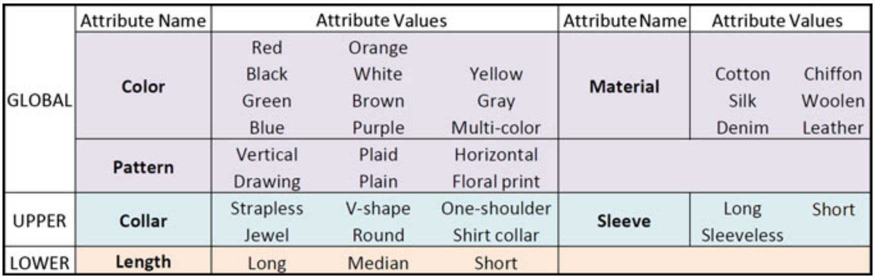


Fig. 9.2 Clothing category attributes. All the attributes are organized in a tree structure and only the leaf nodes are considered in this work



	Attribute Name	Attribute Values			Attribute Name	Attribute Values	
GLOBAL	**Color**	Red Black Green Blue	Orange White Brown Purple	Yellow Gray Multi-color	**Material**	Cotton Silk Denim	Chiffon Woolen Leather
	Pattern	Vertical Drawing	Plaid Plain	Horizontal Floral print			
UPPER	**Collar**	Strapless Jewel	V-shape Round	One-shoulder Shirt collar	**Sleeve**	Long Sleeveless	Short
LOWER	**Length**	Long	Median	Short			

Fig. 9.3 Detail attributes considered in this work

Histograms of Oriented Gradient (HOG), Local Binary Pattern (LBP), color moment, color histogram, and skin descriptor. More specifically, each human part is first partitioned into several smaller, spatially evenly distributed regular blocks. Features extracted from all the blocks are finally concatenated into a 28,770 dimensional feature vector to represent a human part. The block-based features can roughly preserve relative position information inside each human part.

The occasion categories of the clothing are represented by $\mathbf{o} \subset \mathcal{O}$, where $\mathcal{O}$ denotes the finite occasion category set. Note that each clothing may have multiple occasion category labels. The attributes of the upper body clothing are denoted by a vector $\mathbf{a}_u = [a_1^u, \ldots, a_{K_u}^u]^T$, where K_u is the number of attributes considered for the upper body clothing. Each attribute describes certain characteristic of the upper body clothing, e.g., color, collar. Similarly, the attributes of the lower body clothing are denoted as a vector $\mathbf{a}_l = [a_1^l, \ldots, a_{K_l}^l]^T$. All the attributes considered in this work are listed in Figs. 9.2 and 9.3. We denote the attribute set for the upper body and lower body as $\mathcal{A}^u$ and $\mathcal{A}^l$, respectively. Note that each attribute is multivalued and we represent each attribute by a multidimensional binary value vector in the model learning process. For example, the attribute "color" has 11 different values, e.g., red, orange, etc. Then we represent the "color" attribute by an 11-dimensional vector with each element corresponding to one specific type of color.

Given N training examples $\{(\mathbf{x}^{(n)}, \mathbf{a}_u^{(n)}, \mathbf{a}_l^{(n)}, \mathbf{o}^{(n)})\}_{n=1}^N$, our goal is to learn a model that can be used to recommend the most suitable clothing for a given occasion label $o \in \mathcal{O}$, which considers clothing-occasion and clothing–clothing matching simultaneously. Formally speaking, we are interested in learning a scoring function $f_{\mathbf{w}} : \mathcal{X} \times \mathcal{O} \rightarrow \mathbb{R}$, over an image $\mathbf{x}$ and a user specified occasion label o, where $\mathbf{w}$ are the parameters of $f_{\mathbf{w}}$. Here $\mathcal{X}$ denotes the clothing image space. During testing, $f_{\mathbf{w}}$ can be used to suggest the most suitable clothing $\mathbf{x}^*$ from $\mathcal{X}^t$ (candidate clothing repository) for the given occasion o as $\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}^t} f_{\mathbf{w}}(\mathbf{x}, o)$. While for the clothing pairing recommendation, given specified lower body clothing $\mathbf{x}_l$, $f_{\mathbf{w}}$ can select the most suitable upper body clothing $\mathbf{x}_u^*$ as $\mathbf{x}_u^* = \text{argmax}_{\mathbf{x}_u \in \mathcal{X}_u^t} f_{\mathbf{w}} ([\mathbf{x}_u; \mathbf{x}_l], o)$, where $\mathcal{X}_u^t$ denotes the candidate upper body clothing repository. For the lower body clothing pairing, it works similarly.

The recommendation function is defined as follows:

$$\begin{aligned} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, o) = \mathbf{w}_o^T \phi(\mathbf{x}, o) &+ \sum_{j \in \mathcal{A}^u \cup \mathcal{A}^l} \mathbf{w}_{a_j}^T \varphi(\mathbf{x}, a_j) \\ &+ \sum_{j \in \mathcal{A}^u \cup \mathcal{A}^l} \mathbf{w}_{o,a_j}^T \omega(a_j, o) + \sum_{(j,k) \in \mathcal{E}} \mathbf{w}_{j,k}^T \psi(a_j^u, a_k^l). \end{aligned} \tag{9.1}$$

In this model, the parameter vector $\mathbf{w}$ is the concatenation of the parameters in all the factors. $\Phi(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, o)$ is the concatenation of $\phi(\mathbf{x}, o)$, $\varphi(\mathbf{x}, a_j)$, $\omega(a_j, o)$ and $\psi(a_j^u, a_k^l)$. It is a feature vector depending on the images $\mathbf{x}$, the attributes $\mathbf{a}_u$, $\mathbf{a}_l$ and occasion label o. The model presented in Eq. (9.1) simultaneously considers the dependencies among visual features, attributes, and occasions. In particular, its first term predicts occasion from visual features; the second term describes the relationship between visual features and attributes; the third term captures the relationship between attributes and occasion. The last term expresses the dependencies between the attributes of upper and lower body clothing. Instead of predicting the occasion from visual features or attributes directly, we mine much richer matching rules among them explicitly. The impacts of different relationships on the matching score in Eq. (9.1) are automatically determined in the learning process, therefore, the four relationships are not treated equally.

Model Learning and Inference: In this work, we adopt the latent SVM formulation to learn the model as in [8]:

$$\begin{aligned} &\min_{\mathbf{w}, \xi} \beta \|\mathbf{w}\|^2 + \sum_{n=1}^{N} \xi^{(n)} \\ &\text{s.t.} \ \max_{\mathbf{a}_u, \mathbf{a}_l} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{a}_u, \mathbf{a}_l, \mathbf{o}^{(n)}) - \max_{\mathbf{a}_u, \mathbf{a}_l} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{a}_u, \mathbf{a}_l, o) \\ &\quad \geq \Delta(o, \mathbf{o}^{(n)}) - \xi^{(n)}, \quad \forall n, \forall o \in \mathcal{O}, \end{aligned} \tag{9.2}$$

where β is the tradeoff parameter controlling the amount of regularization, and $\xi^{(n)}$ is the slack variable for the n-th training sample to handle the soft margin. Such an objective function requires that the score of clothing for a suitable occasion should be much higher than for a non-suitable occasion. $\Delta(o, \mathbf{o}^{(n)})$ is a loss function defined as

$$\Delta_{0/1}(\mathbf{o}^{(n)}, o) = \begin{cases} 1 & \text{if } o \notin \mathbf{o}^{(n)} \\ 0 & \text{otherwise} \end{cases}$$

In Eq. 9.2, we aim to learn a discriminative occasion-wise scoring function on each pair of clothing (more specifically, on their features and inferred attributes) such that the scoring function can rank clothing correctly by maximizing the score difference between suitable ones and unsuitable ones for the interest occasion.

After learning the model, we can use it to score any image-occasion pair $(\mathbf{x}, o)$. The score is inferred as $f_{\mathbf{w}}(\mathbf{x}, o) = \max_{\mathbf{a}_u,\mathbf{a}_l} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, o)$. Thus after specifying the occasion o, we can obtain a rank of the clothing from the user's clothing photo album. In particular, given the parameter model $\mathbf{w}$, we need to solve the following inference problem during recommendation:

$$\{\mathbf{a}_u^*, \mathbf{a}_l^*\} = \underset{\mathbf{a}_u,\mathbf{a}_l}{\operatorname{argmax}}\ \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, o),$$

which can be solved by linear programming since the attributes form a tree structure [36]. And then the clothing obtaining the highest score will be suggested, namely

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \left\{ \max_{\mathbf{a}_u,\mathbf{a}_l} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{a}_u, \mathbf{a}_l, o) \right\}. \tag{9.3}$$

Similarly, for the clothing pairing recommendation, given a specified upper body clothing $\mathbf{x}_u$ and the occasion o, the most suitable lower body clothing $\mathbf{x}_l^*$ is paired as:

$$\mathbf{x}_l^* = \underset{\mathbf{x}_l}{\operatorname{argmax}} \left\{ \max_{\mathbf{a}_u,\mathbf{a}_l} \mathbf{w}^T \Phi\ ([\mathbf{x}_u; \mathbf{x}_l], \mathbf{a}_u, \mathbf{a}_l, o) \right\}. \tag{9.4}$$

The upper body clothing recommendation for a given lower body clothing is conducted in a similar way.

Evaluation Metric and Baselines: We compare the proposed Magic Closet system with two linear SVM-based models. The first baseline is a feature-occasion multiclass linear SVM which predicts occasion from visual features directly without considering attributes. After training based on $\{\mathbf{x}^{(n)}, \mathbf{o}^{(n)}\}_{n=1}^N$, given an occasion, all the clothing in the repository are ranked according to the output confidence score of the feature-occasion SVM. The second baseline feature-attribute-occasion SVM is composed of a two-layer linear SVM. The first-layer SVM linearly maps visual features to attribute values, which is trained based on $\{\mathbf{x}^{(n)}, \mathbf{a}_u^{(n)}, \mathbf{a}_l^{(n)}\}_{n=1}^N$. Then the visual features are converted into attribute confidence score vectors via such first-layer SVM. The second-layer SVM is trained on these attribute confidence vectors to predict their occasion labels. Similar to feature-occasion SVM, all clothing in the repository are ranked based on the output of the two-layer feature-attribute-occasion SVM. We evaluate their performance via Normalized Discounted Cumulative Gain (NDCG), which is commonly used to evaluate ranking systems.

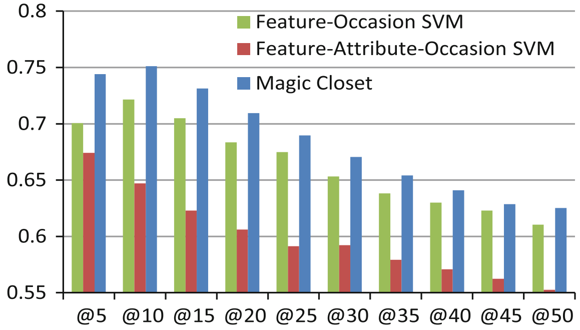
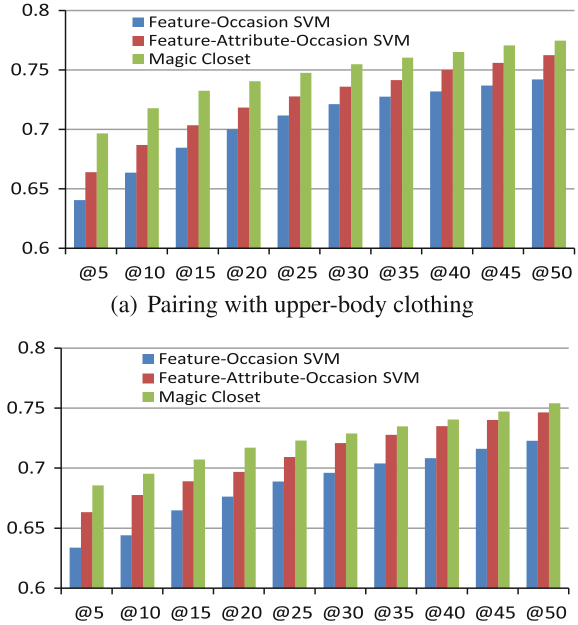


Fig. 9.4 Comparison of Magic Closet with two baselines on the clothing suggestion task (NDCG vs. # returned samples)

Experiment 1: Occasion-Oriented Clothing Suggestion To evaluate the performance of the proposed method, we collect a dataset, which is split into three subsets. The first subset WoW_Full includes 9,469 images containing visible full-body. The second subset, denoted as WoW_Upper, contains 8,421 images with only upper body, such as T-shirts, Fashion hoodies. And the 6,527 images containing lower body clothing, such as Jeans and Skirts, are put into WoW_Lower. According to different sources of data, WoW_Upper is further split into two subsets, one is WoW_Upper_DP where all the images are Daily Photos (DP), which are crawled from popular photo sharing websites, while the other one is WoW_Upper_OS, the photos of which are crawled from Online Shopping (OS) websites. Similarly, both WoW_Lower and WoW_Full subsets are further split into DP and OS subsets in the same way. Though in a practical system all the clothing photos are from the same user, here in order to comprehensively evaluate the Magic Closet system for suggesting clothing with different attributes, we simulate the suggestion scenario on WoW_Full_DP dataset, which contains 6, 661 images from multiple users. We evenly split the WoW_Full_DP subset into two groups. The first half WoW_Full_DP_1 together with WoW_Full_OS (containing 2, 808 images) are used for training the latent SVM-based model embedded in Magic Closet. The second half of WoW_Full_DP_1 is used as testing set. Each set of clothes is annotated with an occasion label, e.g., dating or conference. Given an occasion, the clothing from the set which maximizes the score function in Eq. (9.3) is suggested by Magic Closet.

Quantitative evaluation results of the clothing suggestion are shown in Fig. 9.4. From the results, we can make the following observations. (1) The feature-occasion SVM consistently outperforms the feature-attribute-occasion SVM. This is because the visual features we adopt possess relatively strong discriminative power and their high dimensionality benefits linear classification. We also observe that it is harder to construct a linear relationship between low-dimensional attribute confidence vectors and occasions. (2) The proposed latent SVM model outperforms the two baseline models significantly. This result well demonstrates the effectiveness of the proposed model in mining matching rules among features, attributes, occasions, and utilizing their correlation in occasion-oriented clothing suggestion.

Fig. 9.5 Comparison of Magic Closet with baselines for clothing pairing (NDCG vs. # returned samples)



(a) Pairing with upper-body clothing

(b) Pairing with lower-body clothing

Experiment 2: Occasion-Oriented Clothing Pairing To simulate this scenario, we collect 20 images (10 upper body and 10 lower body) as the queries. Summing up across 8 occasions, the total number of queries is 160. The repository consists of clothing from online shopping dataset, including two subsets WoW_Upper_OS (2,500 images) and WoW_Lower_OS (3,791 images). In clothing pairing, for each query of upper/lower body clothing, we provide the rank of the candidate lower/upperbody clothing in the online shop dataset. The rank is calculated based on the pairs aesthetic score and suitableness for the specified occasion, as evaluated in Eq. 9.4. To obtain the ranking ground truth of the returned clothing, we do not require our labelers (40 people aging from 19 to 40) to score each candidate pair. We adopt the group-wise labeling strategy: given an occasion, we randomly show 8 clothing as a group to the labelers. So, labelers only need to rank the clothing within each group and the final rank is obtained. Such strategy can alleviate the burden of labelers significantly. Each pair is labeled at least 10 times and thus the potential inaccurate rank can be eliminated via averaging.

Figure 9.5 shows the NDCG value w.r.t. the increasing number of returned samples of the baseline models and the Magic Closet system. From the figure, we can have the following observations. (1) For the two baseline methods, the feature-attribute-occasion SVM performs significantly better than the feature-occasion SVM. This is because that the feature-occasion SVM is a linear model. The calculated pairing score equals to $\mathbf{w}^T[\mathbf{x}_u; \mathbf{x}_l] = \mathbf{w}_u^T\mathbf{x}_u + \mathbf{w}_l^T\mathbf{x}_l$. The maximization of this score w.r.t. $\mathbf{x}_l$ is independent of $\mathbf{x}_u$. Therefore, for a specified occasion, for different queries, the returned results are identical. However, due to the good performance of feature-occasion SVM in occasion prediction, it can still return suitable clothing for the occasion. Thus its performance is still acceptable. While for the feature-attribute-occasion SVM, since the features are mapped to the attribute space at first, this issue is alleviated. Moreover, the attribute-based features are more robust to cross-domain variation (DP vs. OS). (2) The proposed Magic Closet outperforms the two baseline models. This result is as expected since Magic Closet can better capture matching rules among attributes and thus recommend more aesthetic clothing pairs.

9.2.2 "Wow You Are so Beautiful Today!"

We have built a system called Beauty e-Experts, a fully automatic system for hairstyle and facial makeup recommendation and synthesis [25]. Given a user-provided frontal face image with short/bound hair and no/light makeup, the Beauty e-Experts system can not only recommend the most suitable hairdo and makeup, but also show the synthetic effects. The interface of the Beauty e-Experts system is shown in Fig. 9.6. The main challenge in this problem is how to model the complex relationships among different beauty and facial/clothing attributes for reliable recommendation and natural synthesis.

To obtain enough knowledge for beauty modeling, we build the Beauty e-Experts Database, which contains 1,505 attractive female photos with a variety of beauty attributes and facial/clothing attributes annotated [25]. Based on this Beauty e-Experts Dataset, two problems are considered for the Beauty e-Experts system: what to recommend and how to wear, which describe a similar process of selecting hairstyle and cosmetics in our daily life. For the what-to-recommend problem, we propose a multiple tree-structured super-graphs model to explore the complex relationships among the high-level beauty attributes, mid-level facial/clothing attributes, and low-level image features, and then based on this model, the most compatible beauty attributes for a given facial image can be efficiently inferred. For the how-to-wear problem, an effective and efficient facial image synthesis module is designed to seamlessly synthesize the recommended hairstyle and makeup into the user facial image.

Beauty attributes, facial/clothing attributes, and features: To obtain beauty knowledge from our dataset, we comprehensively explore different beauty attributes on these images, including various kinds of hairstyles and facial makeups. We carefully organize these beauty attributes and set their attribute values based on some basic observations or preprocessing on the whole dataset. Table 9.1 lists the names and values of all the beauty attributes considered in the work. For the first four beauty attributes in Table 9.1, their values are set intuitively, and for the last five ones, their values are obtained by running the k-means clustering algorithm on the training dataset for the corresponding features. We show the visual examples of specific attribute values for some beauty attributes in Fig. 9.7.

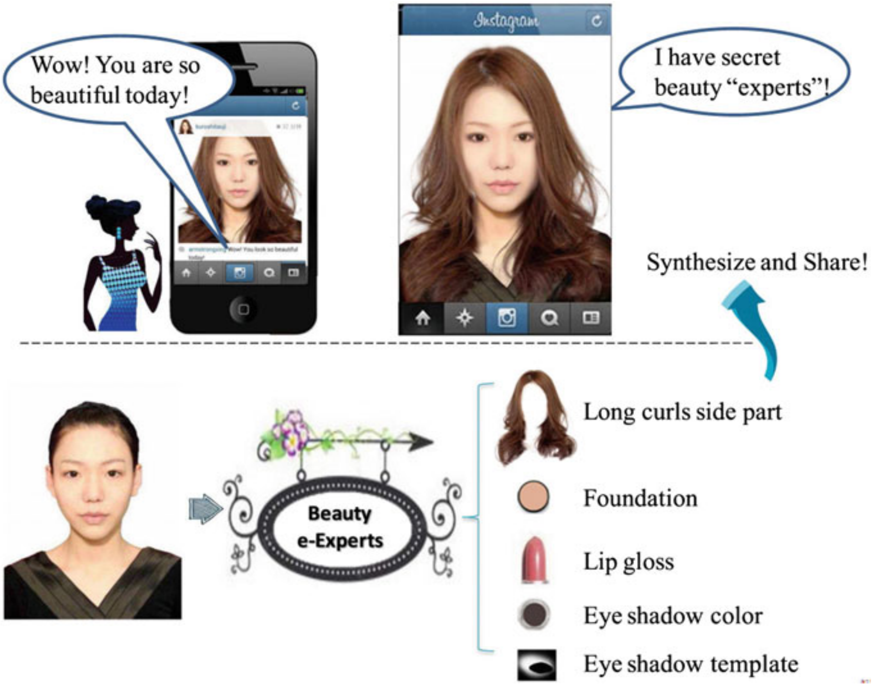


Fig. 9.6 Overall illustration of the proposed Beauty e-Experts system. Based on the user's facial and clothing characteristics, the Beauty e-Experts system automatically recommends the suitable hairstyle and makeup products for the user, and then produces the synthesized visual effects

Table 9.1 A list of the high-level beauty attributes

Name	Values
Hair length	Long, medium, short
Hair shape	Straight, curled, wavy
Hair bangs	Full, slanting, center part, side part
Hair volume	Dense, normal
Hair color	20 classes
Foundation	15 classes
Lip gloss	15 classes
Eye shadow color	15 classes
Eye shadow template	20 classes

We further explore a set of mid-level facial/clothing attributes to narrow the gap between the high-level beauty attributes and the low-level image features. Table 9.2 lists all the mid-level facial/clothing attributes annotated for the dataset. These mid-

Table 9.2 A list of mid-level facial/clothing attributes considered in this work

Names	Values
Forehead	High, normal, low
Eyebrow	Thick, thin
Eyebrow length	Long, short
Eye corner	Upcurved, downcurved, normal
Eye shape	Narrow, normal
Ocular distance	Hypertelorism, normal, hypotelorism
Cheek bone	High, normal
Nose bridge	Prominent, flat
Nose tip	Wide, narrow
Mouth opened	Yes, no
Mouth width	Wide, normal
Smiling	Smiling, neutral
Lip thickness	Thick, normal
Fatness	Fat, normal
Jaw shape	Round, flat, pointed
Face shape	Long, oval, round
Collar shape	Strapless, v-shape, one-shoulder, high-necked, round, shirt collar
Clothing pattern	Vertical, plaid, horizontal, drawing, plain, floral print
Clothing material	Cotton, chiffon, silk, woolen, denim, leather, lace
Clothing color	Red, orange, brown, purple, yellow, green, gray, black, blue, white, pink, multicolor
Race	Asian, Western

level attributes mainly focus on the facial shapes and clothing properties, which are kept fixed during the recommendation and the synthesis process.[3]

After the annotation of the high-level beauty attributes and mid-level facial/clothing attributes, we further extract various types of low-level image features on the clothing and facial regions for each image in the Beauty e-Experts Dataset to facilitate further beauty modeling. The clothing region of an image is automatically determined based on its geometrical relationship with the face region. Specifically, the following features are extracted for image representation:

- RGB color histogram and color moments on the clothing region.
- Histograms of oriented gradients (HOG) and local binary patterns (LBP) features on the clothing region.
- Active shape model [28] based-shape parameters.
- Shape context [1] features extracted at facial points.

[3] Although the clothes of a user can be changed to make one look more beautiful, they are kept fixed in our current Beauty e-Experts system.

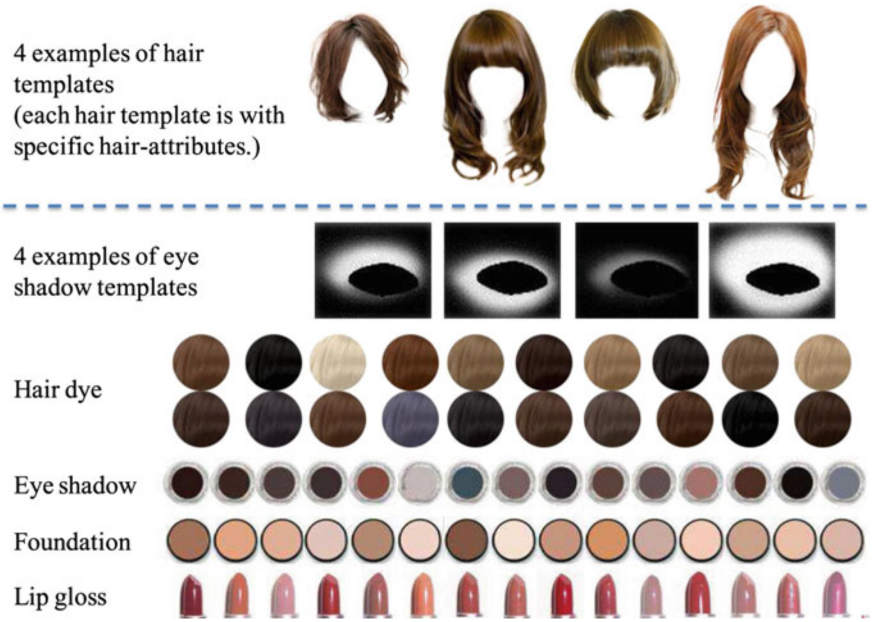

4 examples of hair templates (each hair template is with specific hair-attributes.)
4 examples of eye shadow templates
Hair dye
Eye shadow
Foundation
Lip gloss

Fig. 9.7 Visual examples of the specific values for some beauty attributes

All the above features are concatenated to form a feature vector of 7,109 dimensions, and then Principal Component Analysis (PCA) is performed for dimensionality reduction. The compressed feature vector with 173 dimensions and the annotated attribute values are then fed into an SVM classifier to train a classifier for each attribute.

The Recommendation Model: Based on the beauty attributes and facial/clothing attributes, we propose to learn a multiple tree-structured super-graphs model to explore the complex relationships among these attributes. Based on the recommended results, an effective and efficient facial image synthesis module is designed to seamlessly synthesize the recommended results into the user facial image and show it back to the user. The whole system processing flowchart is illustrated in Fig. 9.8.

A training beauty image is denoted as a tuple $(\langle \mathbf{x}, \mathbf{a}^r \rangle, \mathbf{a}^b)$. Here $\mathbf{x}$ is the image features extracted from the raw image data; $\mathbf{a}^r$ and $\mathbf{a}^b$ denote the set of the facial/clothing attributes and beauty attributes, respectively. Each attribute may have multiple different values, *i.e.,* $a_i \in \{1, \ldots, n_i\}$, where n_i is the number of attribute values for the i-th attribute. The facial/clothing attributes $\mathbf{a}^r$ act as the mid-level cues to narrow the gap between the low-level image features $\mathbf{x}$ and the high-level beauty attributes $\mathbf{a}^b$. The recommendation model needs to uncover the complex relationships among the low-level image features, mid-level facial/clothing attributes and high-level beauty attributes, and make the final recommendation for a given image.

We model the relationships among the low-level image features, the mid-level facial/clothing attributes, and the high-level beauty attributes from a probabilistic

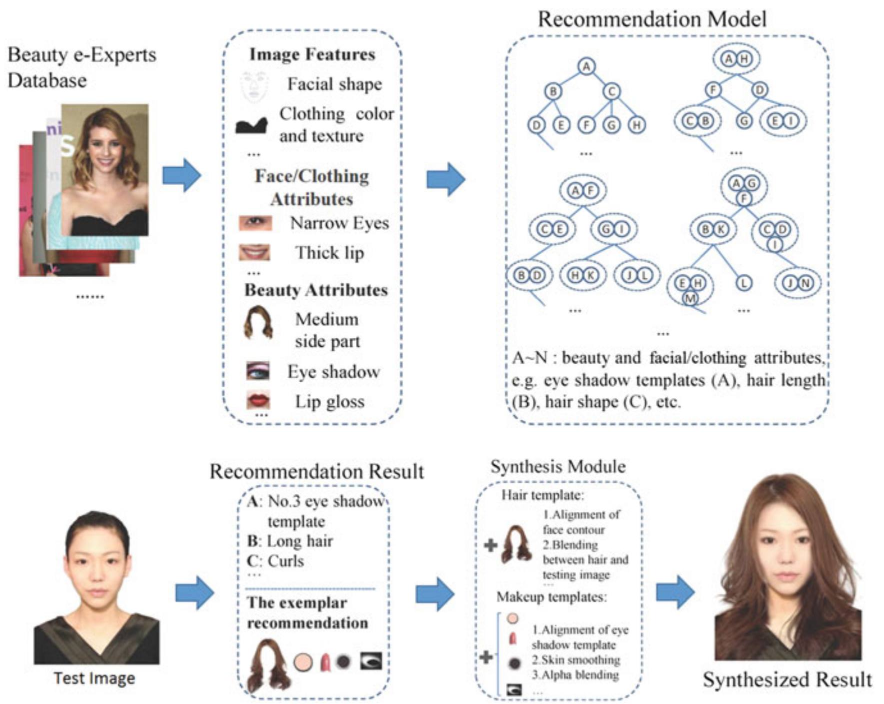

Beauty e-Experts Database
......
Image Features
Facial shape
Clothing color and texture
...
Face/Clothing Attributes
Narrow Eyes
Thick lip
...
Beauty Attributes
Medium side part
Eye shadow
Lip gloss
...
Recommendation Model
A~N : beauty and facial/clothing attributes, e.g. eye shadow templates (A), hair length (B), hair shape (C), etc.
Test Image
Recommendation Result
A: No.3 eye shadow template
B: Long hair
C: Curls
...
The exemplar recommendation
Synthesis Module
Hair template:
1.Alignment of face contour
2.Blending between hair and testing image
...
Makeup templates:
1.Alignment of eye shadow template
2.Skin smoothing
3.Alpha blending
...
Synthesized Result

Fig. 9.8 System processing flowchart. We first collect the Beauty e-Experts Database of 1,505 facial images with different hairstyles and makeup effects. With the extracted facial and clothing features, we propose a multiple tree-structured super-graphs model to express the complex relationships among beauty and facial/clothing attributes. The results from multiple individual super-graphs are fused based on a voting strategy. In the testing stage, the recommended hair and makeup templates for the testing face are then applied to synthesize the final visual effects

perspective. The aim of the recommendation system is to estimate the probability of beauty attributes, together with facial/clothing attributes, given the image features, i.e., $p\left(\mathbf{a}^b, \mathbf{a}^r|\mathbf{x}\right)$, which can be modeled using the Gibbs distribution,

$$p\left(\mathbf{a}^b, \mathbf{a}^r|\mathbf{x}\right) = \frac{1}{Z\left(\mathbf{x}\right)} \exp\left(-E\left(\mathbf{a}^b, \mathbf{a}^r, \mathbf{x}\right)\right), \tag{9.5}$$

where $Z\left(\mathbf{x}\right) = \sum_{\mathbf{a}^b, \mathbf{a}^r} \exp\left(-E\left(\mathbf{a}^b, \mathbf{a}^r, \mathbf{x}\right)\right)$, also known as the partition function, is a normalizing term dependent on the image features, and $E(\mathbf{a}^b, \mathbf{a}^r, \mathbf{x})$ is an energy function measuring the compatibility among the beauty attributes, facial/clothing attributes, and image features. The beauty recommendation results can be obtained by finding the most likely joint beauty attribute state $\hat{\mathbf{a}}^b = \arg\max_{\mathbf{a}^b} \max_{\mathbf{a}^r} p\left(\mathbf{a}^b, \mathbf{a}^r|\mathbf{x}\right)$.

The capacity of this probabilistic model fully depends on the structure of the energy function $E(\mathbf{a}^b, \mathbf{a}^r, \mathbf{x})$. Here we propose to learn a general super-graph structure to build the energy function which can theoretically be used to model any relationships among the low-level image features, mid-level facial/clothing attributes, and high-level beauty attributes. To begin with, we give the definition of a super-graph.

Definition 9.1 Super-graph: a super-graph $\mathcal{G}$ is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is called super-vertexes, consisting of a set of nonempty subsets of a basic node set, and $\mathcal{E}$ is called super-edges, consisting of a set of two-tuples, each of which contains two different elements in $\mathcal{V}$.

It can be seen that a super-graph is actually a generalization of a graph in which a vertex can have multiple basic nodes and an edge can connect any number of basic nodes. When all the super-vertexes only contain one basic node, and each super-edge is forced to connect to only two basic nodes, the super-graph then becomes a traditional graph. A super-graph can be naturally used to model the complex relationships among multiple factors, where the factors are denoted by the vertexes and the relationships are represented by the super-edges.

Definition 9.2 k-order super-graph: for a super-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if the maximal number of vertexes involved by one super-edge in $\mathcal{E}$ is k, $\mathcal{G}$ is said to be a k-order super-graph.

Based on the above definitions, we propose to use the super-graph to characterize the complex relationships among the low-level image features, mid-level facial/clothing attributes, and high-level beauty attributes in our problem. For example, pairwise correlations can be sufficiently represented by a 2-order super-graph (traditional graph), while other more complex relationships, such as one-to-two and two-to-two relationships, can be represented by other higher order super-graphs. Denote the basic node set A as the union of the beauty attributes and facial/clothing attributes, i.e., $A = \{a_i | a_i \in \mathbf{a}^r \cup \mathbf{a}^b\}$. Suppose the underlying relationships among all the attributes are represented by a super-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{a}_i | \mathbf{a}_i \subset A\}$. $\mathbf{a}_i$ is a set of non-empty subsets of A. Note that we use $\mathbf{a}_i$ to denote a non-empty attribute set and a_i to denote a single attribute. $\mathcal{E}$ is the super-edge set that models their relationships, the energy function can then be defined as,

$$E\left(\mathbf{a}^b, \mathbf{a}^r, \mathbf{x}\right) = \sum_{\mathbf{a}_i \in \mathcal{V}} \phi_i\left(\mathbf{a}_i, \mathbf{x}\right) + \sum_{(\mathbf{a}_i, \mathbf{a}_j) \in \mathcal{E}} \phi_{ij}\left(\mathbf{a}_i, \mathbf{a}_j\right). \tag{9.6}$$

The first summation term is called FA (feature to attribute) potential, which is used to model the relationships between the attributes and low-level image features, and the second one is called AA (attribute to attribute) potential and is used to model the complex relationships among different attributes represented by the super-edges. $\phi_i\left(\mathbf{a}_i, \mathbf{x}\right)$ and $\phi_{ij}\left(\mathbf{a}_i, \mathbf{a}_j\right)$ are the potential functions of the corresponding inputs, which can be learned in different ways. Generally, the FA potential $\phi_i\left(\mathbf{a}_i, \mathbf{x}\right)$ is usually modeled as a generalized linear function in the form like

$$\phi_i\left(\mathbf{a}_i = \mathbf{s}_i, \mathbf{x}\right) = \psi_{\mathbf{a}_i}\left(\mathbf{x}\right)^{\top} \mathbf{w}_i^{\mathbf{s}_i}, \tag{9.7}$$

where $\mathbf{s}_i$ is the values for attribute subset $\mathbf{a}_i$, $\psi_{\mathbf{a}_i}(\mathbf{x})$ is a set of feature mapping functions for the attributes in $\mathbf{a}_i$ using SVM on the extracted features, and $\mathbf{w}_i$ is the FA weight parameters to be learned for the model. And the AA potential function $\phi_i\left(\mathbf{a}_i, \mathbf{a}_j\right)$ is defined by a scalar parameter for each joint state of the corresponding super-edge,

$$\phi_{ij}\left(\mathbf{a}_i = \mathbf{s}_i, \mathbf{a}_j = \mathbf{s}_j\right) = w_{i,j}^{\mathbf{s}_i \mathbf{s}_j}, \tag{9.8}$$

where $w_{i,j}^{\mathbf{s}_i \mathbf{s}_j}$ is a scalar parameter for the corresponding joint state of $\mathbf{a}_i$ and $\mathbf{a}_j$ with the specific value $\mathbf{s}_i$ and $\mathbf{s}_j$.

The learning of the super-graph-based energy function includes learning the structure and the parameters in the potential functions.

Model Learning: Structure Learning. For a super-graph built on a basic node set $A = \{a_1, \ldots, a_M\}$ with M elements, we find a k-order tree-structured super-graph for these vertexes. We first calculate the mutual information between each pair of vertexes, and denote the results in the adjacency matrix form, i.e., $W = \{w_{ij}\}_{1 \le i,j \le M}$. Then we propose a two-stage algorithm to find the k-order tree-structured super-graph.

In the *first stage*, we aim to find the candidate set of basic node subsets $\mathcal{V} = \{\mathbf{a}_i | \mathbf{a}_i \subset A\}$, which will be used to form the super-edges. The objective here is to find the set of subsets that has the largest amount of total mutual information in the result k-order super-graph. Here we first define a function that calculates the mutual information of a subset set with a specified mutual information matrix,

$$f(\mathcal{V}, W) = \sum_{|\mathbf{a}_i| \ge 2} \sum_{a_j, a_k \in \mathbf{a}_i} w_{jk}. \tag{9.9}$$

Based on this definition, we formulate the candidate set generation problem as the following optimization problem

$$\begin{aligned} \underset{\mathcal{V}}{\operatorname{argmax}} \quad & f(\mathcal{V}, W), \\ \text{s.t. } |\mathbf{a}_i| \le & \lfloor \frac{k+1}{2} \rfloor, \forall i, \\ |\mathcal{V}| \le & \ m, \end{aligned} \tag{9.10}$$

where the first inequation is from the k-order constraint from the result super-graph, $\lfloor \cdot \rfloor$ is the floor operator, and the parameter m in the second inequation is used to ensure that the generated subsets have a reasonable size to cover all the vertexes and make the inference on the result super-graph more efficient. Specifically, its value can be set as

$$m = \begin{cases} M, & k = 2, \\ 2\lceil M/(k-1) \rceil, & \text{otherwise}, \end{cases} \tag{9.11}$$

where $\lceil \cdot \rceil$ is the ceil operator. To solve this optimization problem, we design a k-means like iterative optimization algorithm to find the solution. The algorithm first initializes some random vertex subsets and then reassigns each vertex to the subsets that result in maximal mutual information increment; if one vertex subset has more than $\lfloor (k+1)/2 \rfloor$ elements, it will be split into two subsets; if the total cardinality of the vertex subset set is larger than $2\lceil M/(k-1) \rceil$, two subsets with the smallest cardinalities will be merged into one subset. This procedure is repeated until convergence.

The *second stage* of the two-stage algorithm first calculates the mutual information between the element pair that satisfies the order restrictions in each vertex subset. The order constraint is that the maximal number of vertexes involved by one super-edge in $\mathcal{E}$ is k. Then it builds a graph by using the calculated mutual information as adjacency matrix, and the maximum spanning tree algorithm is adopted to find its tree-structured approximation.

The above two-stage algorithm is run many times by setting different k values and initializations of subsets, which then generates multiple tree-structured super-graphs with different orders and structures. In order to make the parameter learning tractable, the maximal k-value K is set to be equal to 5.

Model Learning: Parameter Learning. For each particular tree-structured super-graph, its parameter set, including the parameters in the FA potentials and the AA potentials, can be denoted in a whole as $\boldsymbol{\Theta} = \{\mathbf{w}_i^{\mathbf{s}_i}, w_{ij}^{\mathbf{s}_i\mathbf{s}_j}\}$. We adopt the maximal likelihood criterion to learn these parameters. Given N i.i.d. training samples $\mathbf{X} = \{\langle \mathbf{x}_n, \mathbf{a}_n^r \rangle, \mathbf{a}_n^b\}$, we need to minimize the loss function

$$\begin{aligned}\mathcal{L} &= \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n + \frac{1}{2}\lambda\sum_{i,\mathbf{s}_i}\|\mathbf{w}_i^{\mathbf{s}_i}\|_2^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}\left\{-\ln p\left(\mathbf{a}_n^b, \mathbf{a}_n^r|\mathbf{x}_n\right)\right\} + \frac{1}{2}\lambda\sum_{i,\mathbf{s}_i}\|\mathbf{w}_i^{\mathbf{s}_i}\|_2^2, \end{aligned} \tag{9.12}$$

where $\mathcal{L}_n$ is the loss for each sample (expanded in the second line of the equation), λ is the tradeoff parameter between the regularization term and log-likelihood and its value is chosen by k-fold validation on the training set. Since the energy function is linear with respect to the parameters, the log-likelihood function is concave and the parameters can be optimized using gradient-based methods. The gradient of the parameters can be computed by calculating their marginal distributions. Denoting the value of attribute $\mathbf{a}_i$ for training image n as $\hat{\mathbf{a}}_i$, we have

$$\frac{\partial \mathcal{L}_n}{\partial \mathbf{w}_i^{\mathbf{s}_i}} = \left(\left[\hat{\mathbf{a}}_i = \mathbf{s}_i\right] - p\left(\mathbf{a}_i = \mathbf{s}_i|\mathbf{x}_n\right)\right)\psi_{\mathbf{a}_i}\left(\mathbf{x}_n\right), \tag{9.13}$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ij}^{\mathbf{s}_i\mathbf{s}_j}} = \left[\hat{\mathbf{a}}_i = \mathbf{s}_i, \hat{\mathbf{a}}_j = \mathbf{s}_j\right] - p\left(\mathbf{a}_i = \mathbf{s}_i, \mathbf{a}_j = \mathbf{s}_j|\mathbf{x}_n\right), \tag{9.14}$$

where $[\cdot]$ is the Iverson bracket notation, i.e., $[\cdot]$ equals 1 if the expression is true, and 0 otherwise.

Based on the calculation of the gradients, the parameters can be learned from different gradient-based optimization algorithms. In the experiments, we use the implementation by Schmidt[4] to learn these parameters. The learned parameters, together with the corresponding super-graph structures, form the final recommendation model.

Inference: Here each learned tree-structured super-graph model can be seen as a beauty expert. Given an input testing image, the system first extracts the feature vector $\mathbf{x}$, and then each beauty expert makes its recommendation by performing inference on the tree structure to find the maximum posteriori probability of $p\left(\mathbf{a}^b, \mathbf{a}^r|\mathbf{x}\right)$. The recommendation results output by all the Beauty e-Experts are then fused by majority voting to make the final recommendation to the user.

The Synthesis Module: With the beauty attributes recommended by the multiple tree-structured super-graphs model, we further synthesize the final visual effect of hairstyle and makeup for the testing image. To this end, each makeup attribute forms a template which can be directly obtained from a dataset. These obtained hair and makeup templates are then fed into the synthesis process, which mainly has two steps: alignment and alpha blending.

In the alignment step, both of the hairstyle and the makeup templates need to be aligned with the testing image. For hair template alignment, a dual linear transformation procedure is proposed to put the hair template on the target face in the testing image. For the makeup templates alignment, only the eye shadow template needs to be aligned to the eye region in the testing image. Other makeup templates can be directly applied to the corresponding regions based on the face keypoint detection results. In the alpha blending step, the final result is synthesized with hair template, makeup, and the testing face.

Experiments and Results: For the recommendation model in the Beauty e-Experts system, we also implement some alternatives using multiclass SVM, neural network, and latent SVM. Figure 9.9 plots the comparison results of our proposed model and other baselines. The performance is measured by NDCG, which is widely used to evaluate ranking systems. From the results, it is observed that our model and latent SVM significantly outperforms multiclass SVM and neural network. From Fig. 9.9 it can be further found that our model has overall better performance than the latent SVM method, especially in the top 15 recommendations. With higher order relationships embedded, our model can express more complex relationship among different attributes. In addition, by employing multiple tree-structured super-graphs, our model obtains more robust recommendation results.

We then compare the hairstyle and makeup synthesis results with a few commercial systems, including Instant Hair Makeover (IHM),[5] Daily Makeover (DM),[6] and the virtual try-on website (TAAZ).[7] As shown in Fig. 9.10, the first column are the test images, and the other four columns are the results generated by DM, IHM, TAAZ,

[4] http://www.di.ens.fr/~mschmidt/Software/UGM.html.

[5] http://www.realbeauty.com/hair/virtual/hairstyles.

[6] http://www.dailymakeover.com/games-apps/games.

[7] http://www.taaz.com.

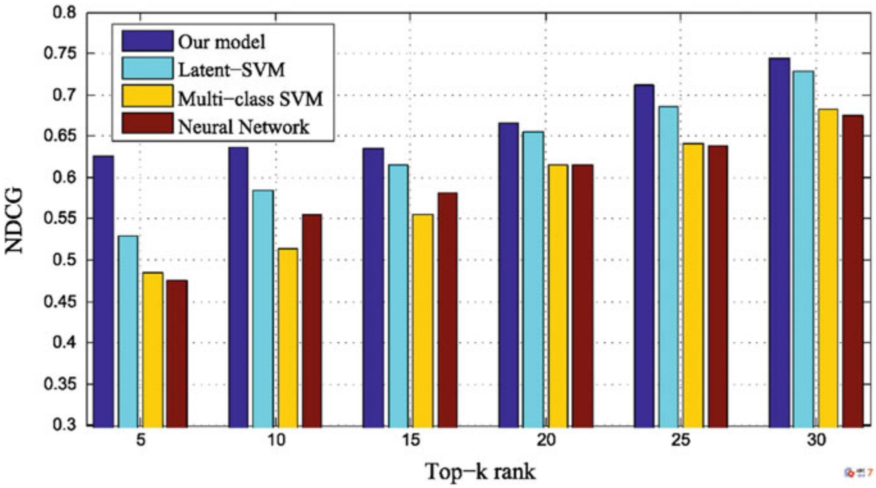


Fig. 9.9 NDCG values of multiple tree-structured super-graphs model and three baselines. The *horizontal axis* is the rank of top-*k* results, while the *vertical axis* is the corresponding NDCG value. Our proposed method achieves better performance than the latent SVM model and other baselines

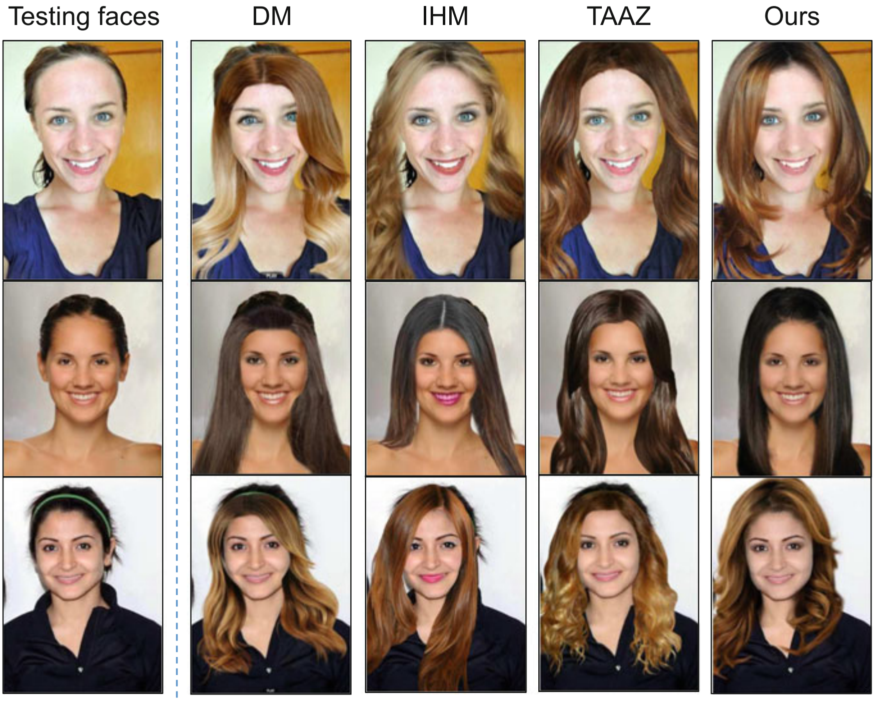


Fig. 9.10 Contrast results of synthesized effect among commercial systems and our paper

and our system, respectively. The reason why we select these three systems is that only these three can synthesize both the hairstyle and makeup effects. The makeup and hairstyle templates used in the synthesis process are selected with some user interactions to ensure that all the four methods share similar makeups and hairstyles. It can be seen that, even after some extra user interactions, the results generated from these three websites have obvious artifacts. The selected hair templates cannot cover the original hair area. IHM cannot even handle the mouth open cases.

9.3 Fine-Grained Clothing Retrieval System

In this section, we describe a fine-grained clothing retrieval system [12]. In a similar fashion to the recommendation work described in the previous section, we use a large-scale annotated dataset with many attributes to transfer knowledge to a noisy real-world domain. In particular, given an offline clothing image from the "street" domain, the goal is to retrieve the same or similar clothing items from a large-scale gallery of professional online shopping images, as illustrated in Fig. 9.11. We propose a Dual Attribute-aware Ranking Network (DARN) consisting of two subnetworks, one for each domain, whose retrieval feature representations are driven by semantic attribute learning.

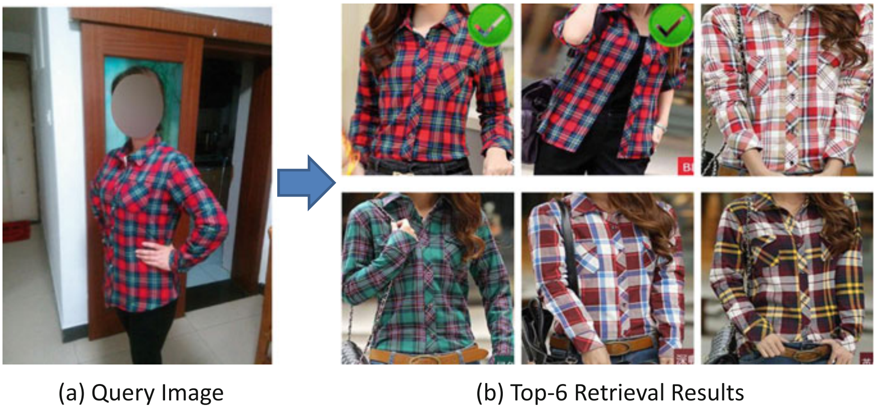


Fig. 9.11 Cross-domain clothing retrieval. **a** Query image from daily photos. **b** Top-6 product retrieval results from the online shopping domain. The proposed system finds the exact same clothing item (first two images) and ranks the ones with similar attributes as top results

9.4 Data Collection

We have collected about 453,983 online upper-clothing images in high-resolution (about 800 × 500 on average) from several online shopping websites. Generally, each image contains a single frontal-view person. From the text surrounding the images, semantic attributes (e.g., clothing color, collar shape, sleeve shape, clothing style) are extracted and parsed into $\langle key, value \rangle$ pairs, where each *key* corresponds to an attribute category (e.g., color), and the *value* is the attribute label (e.g., red, black, white). Then, we manually pruned the noisy labels, merged similar labels based on human perception, and removed those with a small number of samples. After that, 9 categories of clothing attributes are extracted and the total number of attribute values is 179. As an example, there are 56 values for the color attribute.

The specified attribute categories and example attribute values are presented in Table 9.3. This large-scale dataset annotated with fine-grained clothing attributes is used to learn a powerful semantic representation of clothing, as we will describe in the next section.

Recall that the goal of our retrieval problem is to find the online shopping images that correspond to a given query photo in the "street" domain uploaded by the user. To analyze the discrepancy between the images in the shopping scenario (online images) and street scenario (offline images), we collect a large set of offline images with their online counterparts. The key insight to collect this dataset is that there are many customer review websites where users post photos of the clothing they have purchased. As the link to the corresponding clothing images from the shopping store is available, it is possible to collect a large set of online–offline image pairs.

We initially crawled 381,975 online–offline image pairs of different categories from the customer review pages. Then, after a data curation process, where several annotators helped removing unsuitable images, the data was reduced to 91,390 image pairs. For each of these pairs, fine-grained clothing attributes were extracted from the online image descriptions. As can be seen, each pair of images depict the same

Table 9.3 Clothing attribute categories and example values. The number in brackets is the total number of values for each category

Attribute categories	Examples (total number)
Clothes button	Double Breasted, Pullover, … (12)
Clothes category	T-shirt, Skirt, Leather Coat … (20)
Clothes color	Black, White, Red, Blue … (56)
Clothes length	Regular, Long, Short … (6)
Clothes pattern	Pure, Stripe, Lattice, Dot … (27)
Clothes shape	Slim, Straight, Cloak, Loose … (10)
Collar shape	Round, Lapel, V-Neck … (25)
Sleeve length	Long, Three-quarter, Sleeveless … (7)
Sleeve shape	Puff, Raglan, Petal, Pile … (16)

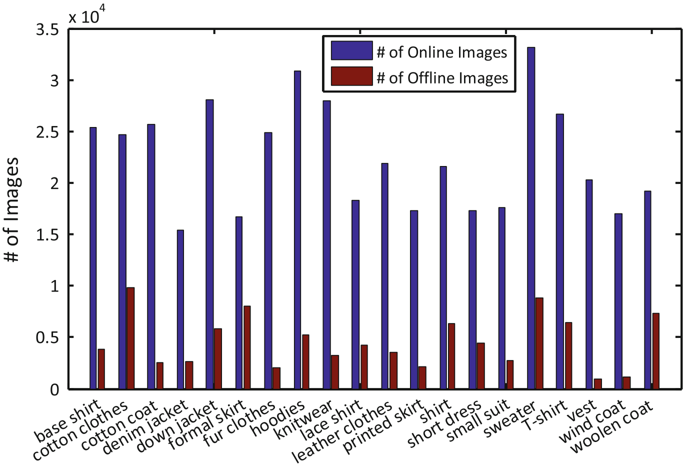


Fig. 9.12 The distribution of online–offline image pairs

clothing, but in different scenarios, exhibiting variations in pose, lighting, and background clutter. The distribution of the collected online–offline images is illustrated in Fig. 9.12. Generally, the number of images of different categories in both scenarios are almost in the same order of magnitude, which is helpful for training the retrieval model.

In summary, our dataset is suitable to the clothing retrieval problem for several reasons. First, the large amount of images enables effective training of retrieval models, especially deep neural network models. Second, the information about fine-grained clothing attributes allows learning of semantic representations of clothing. Last but not least, the online–offline images pairs bridge the gap between the shopping scenario and the street scenario, providing rich information for real-world applications.

9.4.1 Dual Attribute-Aware Ranking Network

In this section, the Dual Attribute-aware Ranking Network (DARN) is introduced for retrieval feature learning. Compared to existing deep features, DARN simultaneously integrates semantic attributes with visual similarity constraints into the feature learning stage, while at the same time modeling the discrepancy between domains.

Network Structure. The structure of DARN is illustrated in Fig. 9.13. Two subnetworks with similar Network-in-Network (NIN) models [22] are constructed as its foundation. During training, the images from the online shopping domain are fed into one subnetwork, and the images from the street domain are fed into the other. Each subnetwork aims to represent the domain-specific information and generate high-level comparable features as output. The NIN model in each subnetwork

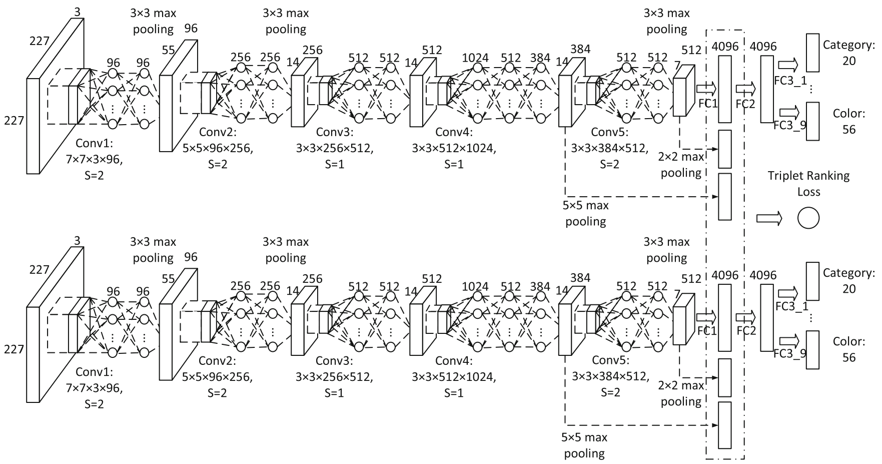


Fig. 9.13 The specific structure of DARN, which consists of two subnetworks for images of the shopping scenario and street scenario, respectively. In each subnetwork, it contains a NIN network, including all the convolutional layers, followed by two fully connected layers. The tree-structure layers are put on top of each network for attribute learning. The output features of each subnetwork, i.e., FC1, Conv4-5, are concatenated and fed into the triplet ranking loss across the two subnetworks

consists of five stacked convolutional layers followed by MLPConv layers as defined in [22], and two fully connected layers (FC1, FC2). To increase the representation capability of the intermediate layer, the fourth layer, named Conv4, is followed by two MLPConv layers.

On top of each subnetwork, we add tree-structured fully connected layers to encode information about semantic attributes. Given the semantic features learned by the two subnetworks, we further impose a triplet-based ranking loss function, which separates the dissimilar images with a fixed margin under the framework of *learning to rank*. The details of semantic information embedding and the ranking loss are introduced next.

Semantic Information Embedding. In the clothing domain, attributes often refer to the specific description of certain parts (e.g., collar shape, sleeve length) or clothing (e.g., clothes color, clothes style). Complementary to the visual appearance, this information can be used to form a powerful semantic representation for the clothing retrieval problem. To represent the clothing in a semantic level, we design tree-structure layers to comprehensively capture the information of attributes and their full relations.

Specifically, we transmit the FC2 response of each subnetwork to several branches, where each branch represents a fully connected network to model each attribute separately. In this tree-structured network, the visual features from the low-level layers are shared among attributes; while the semantic features from the high-level layers are learned separately. The number of neurons in the output-layer of each branch equals the number of corresponding attribute values. Since each attribute has a single value, the cross-entropy loss is used in each branch. Note that the values of some attributes may be missing for some clothing images. In this case, the gradients from the corresponding branches are simply set to zero.

During the training stage, the low-level representation of clothing images is extracted layer by layer. As the activation transfers to the higher layers, the representation becomes more and more abstract. Finally, the distinctive characteristic of each attribute is modeled in each branch. In the back-propagation, the gradient of loss from each attribute w.r.t. the activation of FC2 layer are summed up and transferred back for weight update.

Learning to Rank with Semantic Representation: In addition to encoding the semantic representation, we apply the learning to rank framework on DARN for retrieval feature learning. Specifically, the triplet-based ranking loss is used to constrain the feature similarity of image triplets. Denoting a and b the features of an offline image and its corresponding online image, respectively, the objective function of the triplet ranking loss is:

$$Loss(a, b, c) = max(0, m + dist(a, b) - dist(a, c)), \tag{9.15}$$

where c is the feature of the dissimilar online image, $dist(\cdot, \cdot)$ represents the feature distance, e.g., Euclidean distance, and m is the margin, which is empirically set as 0.3 according to the average feature distance of image pairs. Basically, this loss function imposes that the feature distance between an online–offline clothing pair should be less than that of the offline image and any other dissimilar online image by at least margin m.

In this way, we claim that the triplet ranking loss has two advantages. First and obviously, the desirable ranking ordering can be learned by this loss function. Second, as the features of online and offline images come from two different subnetworks, this loss function can be considered as the constraint to guarantee the comparability of features extracted from those two subnetworks, therefore bridging the gap between the two domains.

We found that the response of FC1 layer, i.e., the first fully connected layer, achieves the best retrieval result. Therefore, the triplet ranking loss is connected to the FC1 layer for feature learning. However, the response from the FC1 layer encodes global features, implying that subtle local information may be lost, which is especially relevant for discriminating clothing images. To handle this problem, we claim that local features captured by convolutions should also be considered. Specifically, the max-pooling layer is used to down-sample the response of the convolutional layers into $3 \times 3 \times f_n$, where f_n is the number of filters in the n-th convolutional layer. Then, the down-sampled response is vectorized and concatenated with the global features. Lastly, the triplet ranking loss is applied on the concatenated features of every triplet. In our implementation, we select the pooled response map of Conv4 and Conv5, i.e., the last two convolutional layers, as local features.

9.4.2 Clothing Detection

As a preprocessing step, the clothing detection component aims to eliminate the impact of cluttered backgrounds by cropping the foreground clothing from images, before feeding them into DARN. Our method is an enhanced version of the R-CNN approach [11], which has recently achieved state-of-the-art results in object detection.

Analogous to the R-CNN framework, clothing proposals are generated by selective search [33], with some unsuitable candidates discarded by constraining the range of size and aspect ratio of the bounding boxes. Similar to Chen et al. [5], we process the region proposals by a NIN model. However, our model differs in the sense that we use the attribute-aware network with tree-structured layers as described in the previous section, in order to embed semantic information as extra knowledge.

Based on the attribute-aware deep features, support vector regression (SVR) is used to predict the intersection over union (IoU) of clothing proposals. In addition, strategies such as the discretization of IoU on training patches, data augmentation, and hard example mining, are used in our training process. As post-processing, bounding box regression is employed to refine the selected proposals with the same features used for detection.

9.4.3 Cross-Domain Clothing Retrieval

We now describe the implementation details of our complete system for cross-domain clothing retrieval.

Training Stage. The training data is comprised of online–offline clothing image pairs with fine-grained clothing attributes. The clothing area is extracted from all images using our clothing detector, and then the cropped images are arranged into triplets.

In each triplet, the first two images are the online–offline pairs, with the third image randomly sampled from the online training pool. As the same clothing images have an unique ID, we sample the third online image until getting a different ID than the online–offline image pair. Several such triplets construct a training batch, and the images in each batch are sequentially fed into their corresponding subnetwork according to their scenarios. We then calculate the gradients for each loss function (cross-entropy loss and triplet ranking loss) w.r.t. each sample, and empirically set the scale of gradients from those loss functions as 1. Lastly, the gradients are back propagated to each individual subnetwork according to the sample domain.

We pre-trained our network as well as the baseline networks used in the experiments on the ImageNet dataset (ILSVRC-2014), as this yields improved retrieval results when compared to random initialization of parameters.

End-to-end Clothing Retrieval. We have set up an end-to-end real-time clothing retrieval demo on our local server. In our retrieval system, 200,000 online clothing images cropped by the clothing detector are used to construct our retrieval gallery. Given the cropped online images, the concatenated responses from FC1 layer, pooled Conv4 layer, and pooled Conv5 layer of one subnetwork of DARN corresponding to shop scenario are used as the representation features. The same processes are operated on the query image, except that the other subnetwork of DARN is used for retrieval feature extraction. We then l_2 normalize the features from different layers for each image. PCA is used to reduce the dimensionality of the normalized features (17,920-D for DARN with Conv4-5) into 4,096-D, which conducts a fair comparison with other deep features using FC1 layer output only. Based on the preprocessed features, the Euclidean distance between query and gallery images is used to rank the images according to the relevance to the query.

9.4.4 Experiments and Results

For the retrieval experiment, about 230,000 online images and 65,000 offline images are sampled for network training. In the training process, each offline image and its online counterpart are collected, with the dissimilar online image randomly sampled from the 230,000 online pool to construct a triplet. To make the retrieval result convincing, the rest 200,000 online images are used as the retrieval gallery.

For clothing retrieval, the approach using Dense-SIFT *(*DSIFT) + fisher vector *(*FV) is selected as traditional baseline. To analyze the retrieval performance of deep features, we compare pretrained networks including AlexNet (*pretrained CNN*) and *pretrained NIN*. We denote the overall solution as Dual Attribute-aware Ranking Network (*DARN*), the solution without dual structure as Attribute-aware Ranking Network (*ARN*), the solution without dual structure and the ranking loss function as Attribute-aware Network (*AN*). We further evaluate the effectiveness of DARN in terms of different configurations w.r.t. the features used, *DARN* using the features obtained from FC1, *DARN with Conv4* using the features from FC1+Conv4, and *DARN with Conv4-5* using the features from FC1+Conv4+Conv5. It is worth noting that the dimension of all features is reduced to 4096 by PCA to have a fair comparison.

Figure 9.14 shows the full detailed top-k retrieval accuracy results for different baselines as well as their proposed methods. We vary *k* as the tuning parameter as it is an important indicator for a real system.

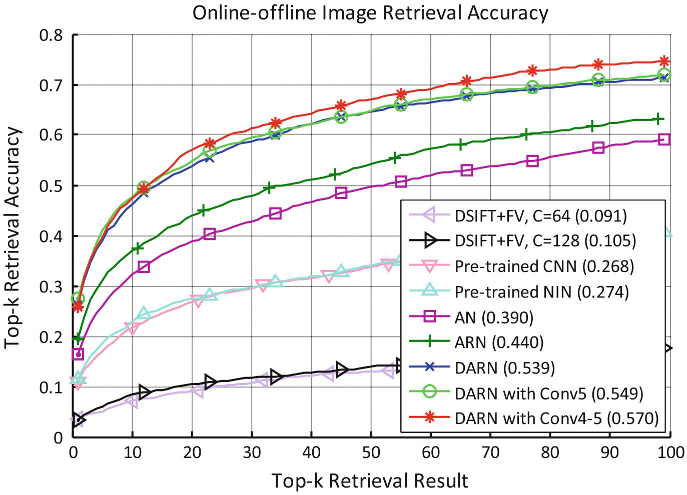


Fig. 9.14 The top-k retrieval accuracy on 200,000 retrieval gallery. The number in the parentheses is the top-20 retrieval accuracy

9.5 Summary

In this chapter, we reviewed fashion attribute prediction and its applications in fashion recommendation and fashion retrieval. We introduced two recommendation systems. The first system is called Beauty E-expert, a fully automatic system for hairstyle and facial makeup recommendation. The second system is called Magic Closet, which is an occasion-oriented clothing recommendation system. For fashion retrieval, a fine-grained clothing retrieval system was developed to retrieve the same or similar clothing items from online shopping stores based on a user clothing photo. In each of these systems, we described an approach to transfer knowledge from a large ground truth dataset to a specific challenging real-world scenario. Visual features were used to learn semantic fashion attributes and their relationships to images from a similar but more challenging user domain. By simultaneously embedding semantic attribute information and visual similarity constraints, we have been able to construct practical real-world systems for fashion analytics.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape context: a new descriptor for shape matching and object recognition. In: Conference on Neural Information Processing Systems (NIPS) (2000)
2. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: European Conference on Computer Vision (ECCV) (2010)
3. Bourdev, L., Maji, S., Malik, J.: Describing people: a poselet-based approach to attribute classification. In: International Conference on Computer Vision (ICCV) (2011)

4. Chen, H., Gallagher, A., Girod, B.: Describing clothing by semantic attributes. In: European Conference on Computer Vision (ECCV) (2012)
5. Chen, Q., Huang, J., Feris, R., Brown, L., Dong, J., Yan, S.: Deep domain adaptation for describing people based on fine-grained clothing attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
6. Datta, A., Feris, R., Vaquero, D.: Hierarchical ranking of facial attributes. In: IEEE International Conference on Automatic Face and Gesture Recognition (FG) (2011)
7. Donahue, J., Grauman, K.: Annotator rationales for visual recognition. In: International Conference on Computer Vision (ICCV) (2011)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
9. Feris, R., Bobbitt, R., Brown, L., Pankanti, S.: Attribute-based people search: lessons learnt from a practical surveillance system. In: International Conference on Multimedia Retrieval (ICMR) (2014)
10. Gallagher, A., Chen, T.: Clothing cosegmentation for recognizing people. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
12. Huang, J., Feris, R.S., Chen, Q., Yan, S.: Cross-domain image retrieval with a dual attribute-aware ranking network. In: International Conference on Computer Vision (ICCV) (2015)
13. Kiapour, M., Yamaguchi, K., Berg, A., Berg, T.: Hipster wars: discovering elements of fashion styles. In: European Conference on Computer Vision (ECCV) (2014)
14. Kiapour, M.H., Han, X., Lazebnik, S., Berg, A.C., Berg, T.L.: Where to buy it: matching street clothing photos in online shops. In: International Conference on Computer Vision (ICCV) (2015)
15. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Image search with relative attribute feedback. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
16. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: International Conference on Computer Vision (ICCV) (2009)
17. Kwak, I., Murillo, A., Belhumeur, P., Kriegman, D., Belongie, S.: From bikers to surfers: visual recognition of urban tribes. In: British Machine Vision Conference (BMVC) (2013)
18. Layne, R., Hospedales, T., Gong, S.: Person re-identification by attributes. In: British Machine Vision Conference (BMVC) (2012)
19. Li, A., Liu, L., Wang, K., Liu, S., Yan, S.: Clothing attributes assisted person re-identification. IEEE Trans. Circ. Syst. Video Technol. (TCSVT) **25**(5), 869–878 (2014)
20. Liang, X., Liu, S., Shen, X., Yang, J., Liu, L., Lin, L., Yan, S.: Deep human parsing with active template regression. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **37**(12), 2402–2414 (2015)
21. Liang, X., Xu, C., Shen, X., Yang, J., Liu, S., Tang, J., Lin, L., Yan, S.: Human parsing with contextualized convolutional neural network. In: International Conference on Computer Vision (ICCV) (2015)
22. Lin, M., Chen, Q., Yan, S.: Network in network. In: International Conference on Learning Representations (ICLR) (2014)
23. Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C., Yan, S.: Hi, magic closet, tell me what to wear! In: ACM Multimedia (ACM MM) (2012)
24. Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-shop: cross-scenario clothing retrieval via parts alignment and auxiliary set. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
25. Liu, L., Xing, J., Liu, S., Xu, H., Zhou, X., Yan, S.: Wow! you are so beautiful today! ACM Trans. Multimedia Comput, Commun. Appl. (TOMM) **11**(1s), 20 (2014)
26. Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., Yan, S.: Fashion parsing with weak color-category labels. IEEE Trans. Multimedia (TMM) **16**(1), 253–265 (2014)

27. Liu, S., Liang, X., Liu, L., Shen, X., Yang, J., Xu, C., Lin, L., Cao, X., Yan, S.: Matching-cnn meets knn: Quasi-parametric human parsing. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
28. Milborrow, S., Nicolls, F.: Locating facial features with an extended active shape model. In: European Conference on Computer Vision (ECCV) (2008)
29. Parikh, D., Grauman, K.: Relative attributes. In: International Conference on Computer Vision (ICCV) (2011)
30. Sharmanska, V., Quadrianto, N., Lampert, C.H.: Learning to rank using privileged information. In: International Conference on Computer Vision (ICCV) (2013)
31. Shi, Z., Hospedales, T., Xiang, T.: Transferring a semantic representation for person re-identification and search. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
32. Song, Z., Wang, M., Hua, X., Yan, S.: Predicting occupation via human clothing and contexts. In: International Conference on Computer Vision (ICCV) (2011)
33. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vision (IJCV) **104**(2), 154–171 (2013)
34. Vapnik, V., Vashist, A.: A new learning paradigm: learning using privileged information. Neural Netw. **22**(5), 544–557 (2009)
35. Vaquero, D., Feris, R., Brown, L., Hampapur, A.: Attribute-based people search in surveillance environments. In: Workshop on Applications of Computer Vision (WACV) (2009)
36. Wang, Y., Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
37. Wang, J., Chen, Y., Feris, R.: Walk and learn: facial attribute representation learning from egocentric video and contextual data. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
38. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Parsing clothing in fashion photographs. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
39. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)