2018 IEEE Winter Conference on Applications of Computer Vision

Efficient Multi-Attribute Similarity Learning *Towards Attribute-based Fashion Search*

Kenan E. Ak,[1,2] Joo Hwee Lim,[2] Jo Yew Tham[3], and Ashraf A. Kassim[1]
[1]National University of Singapore, Singapore
[2]Institute for Infocomm Research, A*STAR, Singapore
[3]ESP xMedia Pte. Ltd., Singapore
emir.ak@u.nus.edu, joohwee@i2r.a-star.edu.sg, thamjy@espxmedia.com, ashraf@nus.edu.sg

Abstract

In this paper, we propose an attribute-based query & retrieval system designed for fashion products. Our system addresses the problem of carrying out fashion searches by the query image and attribute manipulation, e.g. replacing long sleeve attribute of a dress to sleeveless. We present the attributes in two groups: (1) general attributes (category, gender etc.) and (2) special attributes (sleeve length, collar etc.). The special attributes are more suitable for the attribute manipulation and thus conducting searches. In order to solve the mentioned fashion search problem, it is crucial for the deep neural networks to understand attribute similarities. To facilitate more specific similarity learning, clothing items are represented by their structural subcomponents or "parts". The parts are estimated using an unsupervised segmentation method and used inside the proposed Convolutional Neural Network (CNN) as an attention mechanism. Meaning, different parts are connected to the special attributes, e.g. sleeve part is connected with sleeve length attribute. With this mechanism, part-based triplet ranking constraint is applied to learn similarity of each special attribute independently from one another in a single network. In the end, the well-defined features are used to conduct the fashion search. Additionally, an adaptive relevance feedback module is used to personalize the fashion search process with the feature descriptions. For our experiments, a new dataset is constructed containing 101,021 images which consist of pure clothing items. Besides achieving decent retrieval results in our dataset, the experiments show that proposed technique outperforms different baselines and is able to adapt towards user's requests.

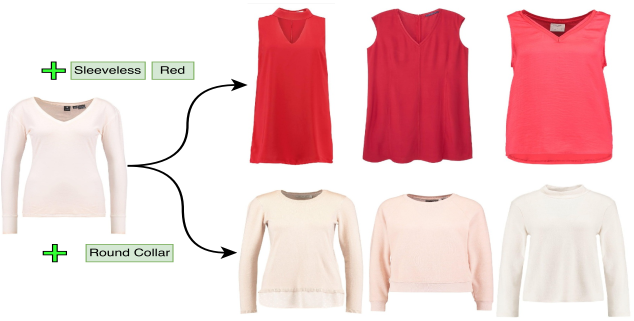


Figure 1: An application example of the proposed work. Given a query image, a list of attributes are created and returned to the user from the classification scores. After applying demonstrated attribute manipulations (denoted with green color), our proposed network finds the top-3 similar products, ranked from left to right.

1. Introduction

Many online shopping and e-commerce systems have been launched in recent years to tap into the insatiable demand for clothing products. However, the enormous variety of clothing products make it very challenging and time-consuming for consumers to find their most preferred items from thousands of images. It is especially challenging for users to keep redefining some attributes in their queries as they repeatedly conduct searches.

Much work has been carried out in the field of clothing recognition [2][3][6][27] and retrieval [9][12][16][17]. However, the only major recent work related to clothing search which allows attribute manipulations is that of [28]. Most methods in the fashion search literature are not sufficient enough to completely describe clothing images and consequently limit the user interaction. To add up, the current query based image retrieval systems mostly aim to find similar images in a 'generic way' while ignoring the individuality of each user. To address these issues, our work considers different sub-components of clothing products to learn more discriminative attribute similarities and the fashion search is personalized by combining relevance feedback

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00186

with the well-defined descriptors.

A use-case of our proposed retrieval/manipulation system which can be applied to any fashion image is depicted in Figure 1. Given a query image with white color, long sleeve and v-neck attributes, two different attribute manipulations are applied (denoted with green color). After conducting the illustrated attribute manipulations, the top-3 retrieved images are listed according to their rankings. Note that all original attributes of the query image remain unchanged (pattern, pocket, clothing category etc.) while changing the undesired attributes.

The proposed network in this work consists of two different paths for part-based attribute similarity learning. The first path is used for general attributes and decides whether to conduct unsupervised part extraction or not (decide if the input image belongs to upper or lower body). The second path combines extracted parts with the special attributes where part-based triplet ranking loss constraint is applied. With this method, we are able to exploit part information and choose triplets of images for each attribute. Triplets of images are used to conduct similarity learning and consist of anchor (1), positive (2) and negative (3) images. The rule of picking triplet images is that the anchor and positive images must share the same attribute label while the negative image must have a different attribute label.

Breaking the tasks into different paths has the following advantages: 1) the part extraction method can be used to attend to the most relevant region of each special attribute and 2) triplets of images can be chosen independently from other attributes which consequently relaxes the conditions of picking triplets of images. As a result, each attribute is represented with a signature vector extracted from a fully connected layer and used to perform the fashion search.

By using the learned representations, fashion search can be conducted and generic images decided by the network is retrieved according to the query image and attribute manipulations. However, the retrieved ”generic image” may not fit every user's preferences. Based on relevant search and learned signature vectors, we propose a personalization approach which incorporates with the individual's perspective. For this purpose, the relevance feedback mechanism [11] is used to find which attribute the user cares the most so that the next fashion search can be tailor-made. Relevant search teaches the network how to engage with users. Hereby, the network redefines some of its components for the future image searches.

Our contributions can be summarized as follows:

1. A new dataset containing 101,021 images consisting of pure clothing items extracted from different e-commerce providers for fashion studies has been developed. Each image in the dataset represented with general and special attributes, where the special attributes are more suitable for attribute manipulation and thus conducting fashion searches.

2. A two-path CNN is introduced for the fashion search. The first path is used for general attributes and whether to extract parts or not. In the second path, part-based triplet ranking constraint for special attributes is added which helps the network to understand attribute similarities. This technique increases the number of triplets of images exponentially; therefore, provides a better learning structure for the network.

3. The relevant search method is utilized to adapt well-trained discriminators with the personalization which brings humans into the search process.

2. Related Work

2.1. Attributes and Recognition

Attributes are used to define mid-level semantic visual concepts [7], such as ”pattern”, ”sleeve”, ”neckline” etc. Attributes can be learned through a combination of image features and classifiers. Attributes are used in different computer vision related problems, such as object detection [26], zero shot learning [21][23], or discover new attributes [1][22]. Attributes also play an important role in fashion related research. In clothing attribute recognition, early works [3][13] relied on handcrafted features such as SIFT [19], HoG [4] and color. Deep learning algorithms, eliminated the need for hand-crafted features which enabled more compact feature representations. The work of Song [25] proposed to use a unified model for multiple object verticals. In [10], spatially-aware fashion concept discovery is made possible. Deep networks such as DeepFashion [18] and MTCT [6], empowered from the large datasets proved themselves to be efficient in attribute recognition. Although the attribute recognition is not the main purpose of our work, the proposed network is closely related with the attributes as we aim to find the attribute similarities.

2.2. Clothing Retrieval

A representative work in clothing retrieval is proposed by [17], where cross scenario match is conducted. A complete clothing pairing system is proposed in [16] with latent SVM's. [13] implemented a clothing product suggestion pipeline by using clothing recognition and segmentation. In [20], local similarity feature is used to merge different items. The joint ranking and classification network is introduced in [24] using weak data for feature extraction. An interesting method is proposed by Kiapour [9] to match a real-world sample with the exact same item in the online shopping dataset. Several networks, such as DeepFashion [18] and DARN [12] are proposed to solve cross-domain retrieval. However, not a single research is introduced to conduct a fashion search after changing an attribute except

Attribute Categories	Attribute Labels (total number)
Category	Shirt, Dress, Trouser, ... (16)
Collar	Polo, Round, Mandarin, ... (17)
Color	Wine, Navy, Blue,... (19)
Fabric	Denim, Jersey, Sweat, ... (14)
Fastening	Zip, Belt, Button, ... (9)
Fit	Skinny, Loose, Straight, ... (15)
Gender	Male, Female (2)
Length	Normal, 3/4, Short, ...(7)
Neckline	Henley, Boat, Envelope, ... (11)
Pattern	Animal, Plain, Photo, ... (15)
Pocket	Side, Flap, Zip, ... (7)
Sleeve length	Long, Short, Sleeveless, ... (9)
Sport	Running, Yoga, Football, ... (15)

Table 1: The full list of attribute categories and labels to show the detail level of the constructed dataset. Special attributes are shown in a bold text which are more intuitive for user manipulations.

AMNet [28]. The focus of AMNet [28] is called "fashion search with attribute manipulation". AMNet [28] is able to manipulate attributes using a memory gate. In contrast to Zhao et. al's work [28], our proposed system follows different solution by choosing triplets of images using a part-based approach and conducts more specific learning. Moreover, we bring in personalization aspect into our system and offer another degree of freedom for the fashion search.

Most of the aforementioned methods benefit from advanced pose estimation or clothing detection algorithms in order to capture the clothing product. However, the captured clothing items may be affected by the background or low image quality. Consequently, this limits the detailed search for a certain clothing item and user involvement is restricted. In order to conduct high-resolution analysis, we directly work on pure in-shop clothing items and try to generate more descriptive learning by considering parts of fashion images.

3. Dataset Construction: Shopping100k

In the fashion related research, many different datasets are available [9][12][14][18]. However, they all consist of a person posing which may not be suitable for high-resolution analysis. Moreover, the posing model may be wearing several clothing items which may create some occlusion problems and it is unclear to focus on which product. In terms of attribute types, our dataset is similar to DARN [12], but their work focuses on cross-scenario image retrieval with real-world images. We want a new dataset which only has pure clothing items with a simple background so that image search can become more detailed for the users.

We first collect around 800,000 images which correspond to 143,021 products (each product have several images) with their meta-data. By using a CNN trained for selecting the image with no human and single-frontal view we get rid of all irrelevant images for each product which leaves 101,021 images in total. Our labeling procedure takes advantage of shopping site where meta-data is available and associated with collected images. We define some keywords for attribute groups (eg. collar - pattern) which correspond to a specific attribute label value (eg. lapel - solid). To get rid of noisy labels, similar labels are merged and labels with small samples are removed manually. To sum up, 12 fashion attributes are created with 151 total number of labels as shown in Table 1. Some examples are given in Figure 2 to demonstrate the detail level of the dataset. The images in the constructed dataset might be simple because of a clean background. However, each image has at least 5-6 attributes wherein DeepFashion [18] dataset, there are around 3-4 attributes per image which may be uninformative sometimes. Moreover, we believe the literature does not have many multi-attribute based datasets.

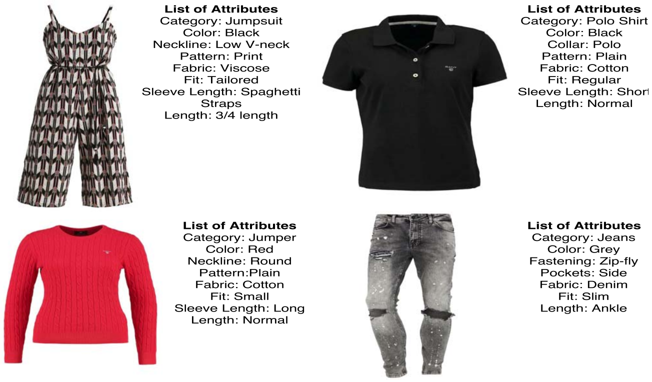


Figure 2: Some example images and their attributes from the constructed dataset.

4. Part Guided Fashion Retrieval Network

4.1. Problem Formulation

We represent attributes in 2 groups: general (category, gender etc.) and special attributes (sleeve length, collar type etc. The reason we divide them is special attributes are much more suitable for the detailed retrieval task in the fashion products and a better learning mechanism should be provided for them.

A query image I_q can be described by its general $(g_{q1}, g_{q2}, ..., g_{qG})$ and special attribute labels $(s_{q1}, s_{q2}, ..., s_{qS})$, where G and S is the number of general and specific attribute respectively. In this work, we focus on 2 different search scenarios: 1) Direct retrieval: where the user wants to find an image with same attributes as I_Q. 2) Retrieval after manipulating some special

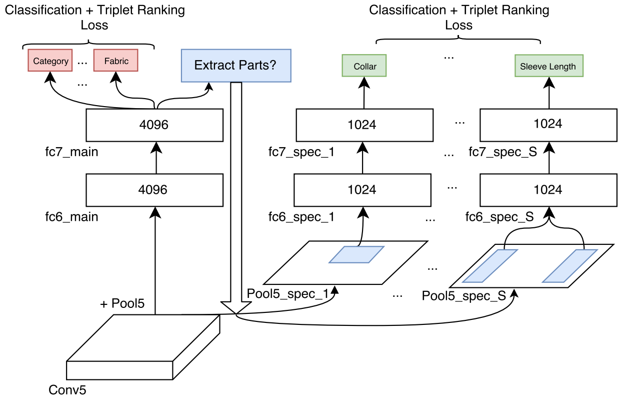


Figure 3: Overall architecture of the proposed system. Red, blue, green colors represent general attributes, part extraction method and special attributes respectively.

attributes. To give an example for the second scenario, the user may want change s_{q2} attribute to s^*_{q2}. We try to solve this problem by directly replacing signature vector representing s_{q2} with a 'generic' representation of s^*_{q2}. This process also keeps all untouched attributes in the query image unchanged.

Currently, it is not clear how to train the deep networks for the multi-attribute similarity. For that purpose, we offer a flexible method to fix this issue with a mechanism to learn attribute similarities independently from one another.

4.2. Architecture Overview

CNN's proved themselves to be very useful in both image recognition and retrieval problems. Here we adopt the well-known CNN model AlexNet [15]. The original network is modified as shown in Figure 3. Following modifications are performed to the original network.

The overall architecture of our work is illustrated in Figure 3. Network architecture from conv-1 to conv-5 layers kept same as the AlexNet [15]. At this point, the network consists of following components: (1) Conv-5 features after the pooling layer are passed through the left-hand side of the network to learn general attributes and two types of loss functions are applied. The first loss function is the attribute classification and the second one is the triplet ranking loss. We discovered using only the ranking loss is not quite sufficient and it must be complemented with classification loss and trained jointly. (2) A fully connected layer called "Extract Parts?" to decide whether the part extraction method is applicable or not by using the rules which are specified in the part extraction subsection. (3) At the right-hand side of the modified network, ROI pooling idea inspired from the [8] is used to feed part-specific features within the network to a set of fully connected layers. For each special attribute, a new branch is used and at the end of the fully connected layers, same loss functions mentioned in the first item are used again. With this setup, the part-specific features of special attributes are passed through the right-hand side of the network and similarity learning is conducted independently from other attributes.

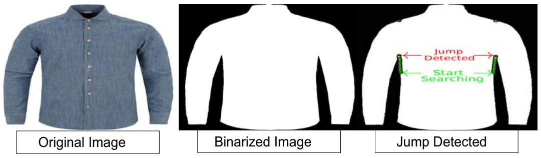


Figure 4: Illustration of the jump detection algorithm. Using binarized image, the discontinuity between sleeve/torso is detected and by using discontinuity points different parts are extracted. Note that a regression method can also be used to detect discontinuity point for more challenging datasets.

4.3. Part Extraction

In order to perform the similarity learning in a more specific space, some kind of part information must be learned. However, we refrain our work to use any supervised detection algorithm which requires bounding-boxes as it would be costly. In this work, the structure of the upper-body clothing items is used to extract different parts with a simple method. As all images in the dataset share similar poses, there is no need to over-engineer this problem with expensive detection methods. For more challenging datasets, our method can be replaced with a regression method to extract different parts.

The first step is to use a straightforward color based segmentation to separate the background and foreground. The obtained binary mask (B) is then improved by using dilation and erosion to remove small gaps and noisy regions. The initial aim is to find key points which will be used to retrieve desired parts from the B. The step detection idea from the signal processing is adopted for this task. All upper images have a discontinuity between torso and sleeves. The point where this discontinuity appears first will give "Jump Location". The detailed visual explanation is illustrated in Figure 4.

The jump detection algorithm is applied in order to find the left and right sleeve parts. After acquiring jump locations, upper jump locations also found and B is approximately segmented into 4 pieces, namely: torso, left/right sleeves, and collar. Moreover, by using the middle of torso region, fastening part is extracted. If there are no sleeves, other parts are approximately estimated as the pose of the input is static.

We plug the part extraction method into the network using a fully connected layer (named: "Extract Parts?") with 3 cases (outputs). In the first case, the part extraction method is deactivated if the input belongs to the lower-body clothing. In the second case, if the sleeve is detected, the part extraction method is activated. The last case covers sleeveless images, where all other parts are extracted except the sleeve region. When the extraction is deactivated for a part, the network does not use ROI pooling and passes the whole feature map of the conv-5th layer after the pooling layer through the fully connected layers of the special attributes.

4.4. Classification Loss Computation

As mentioned in architecture overview section, multiple fully connected layers are added to make predictions of attributes. This helps network to learn about the semantic representation of the fashion images in the feature space. In the each branch, cross entropy loss function is used for attribute prediction.

$$L_{attribute} = -\sum_{i=1}^{N}\sum_{a=1}^{G+S} log(p(g_{ia} = y_{ia}|x_{ia})) \tag{1}$$

$$p(g_{ia} = y_{ia}|x_{ia}) = \frac{e^{x_{ia}^T w_a}}{\sum_{k=1}^{K} e^{x_{ia}^T w_k}} \tag{2}$$

The loss function for attribute recognition is given in Eq. 1, where N is the number of training examples, G is the number of general attributes, S is the number of special attributes and g_{ia} denotes the ground truth of the a'th attribute of i'th image. Posterior probability is estimated as shown in Eq. 2 using the softmax function. x_{ia} represents the feature vector, K is the number of all corresponding possible values for g_{ia}.

4.5. Triplet Ranking Loss Computation

A desired property of the proposed network is to learn special attribute similarities independent from one another. Given example in Figure 5 shows triplets of images choices of anchor $(\hat{i})$, positive (i^+) and negative (i^-) images to be used in collar attribute similarity learning. Although i^- might seem more similar to $\hat{i}$ than i^+, in terms of collar attribute they are quite different. If a pre-trained network without ranking loss is used to check the Euclidean distance of the extracted features, $d(\hat{i}, i^+) > d(\hat{i}, i^-)$ would be the situation. However, we employ part extraction and introduce a more relaxed triplet ranking constraint to make $d(\hat{i}, i^+) < d(\hat{i}, i^-)$. Doing this for each special attribute would result in having a well-designed set of features which would ease the search process.

Triplet ranking loss defined as follows:

$$L_{triplet} = -\sum_{i=1}^{N}\sum_{v=1}^{S+1} max(0, m + d(fc_{7v}(\hat{i}), fc_{7v}(i^+)' \\ -d(fc_{7v}(\hat{i}), fc_{7v}(i^-)') \tag{3}$$

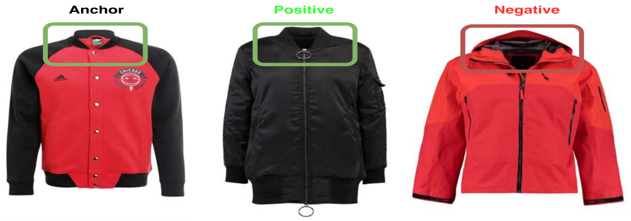


Figure 5: Part based triplet training idea. At first glance, it might seem like the anchor and positive image is not similar to each other. However, when collar parts/attributes are considered they share same attributes. Negative image is randomly chosen from the collar attribute set which does not share the same label as the anchor image.

The aim of Equation 3 is to learn a signature vector (fc_{7v}) for each attribute which minimizes $d(fc_{7v}(\hat{i}), fc_{7v}(i^+)')$, maximizes $d(fc_{7v}(\hat{i}), fc_{7v}(i^-)')$ by a margin m. S+1 different signature vectors are learned which is the number of all special attributes + 1 for general attributes, the signature vectors are represented with v.

Another aim of this part-based ranking training scheme is to make each special attribute invariant from one another. For example, collar attribute similarity should be independent of color or sleeve attributes as shown in Figure 5. Moreover, our method relaxes the conditions by breaking similarity learning into a couple of tasks which effectively increases the number of triplet examples exponentially.

The rule of picking triplets of images for the special attributes is quite simple. $\hat{i}$ and i^+ must have the same special attribute label while i^- is a different label and chosen randomly. The rule of picking triplets for the general attributes is more strict. For this case, $\hat{i}$ and i^+ must have same attribute label for each general attribute and i^- can be chosen randomly again.

4.6. Fashion Search

After training the network, each signature vector representation of the query image can be compared with the images in the retrieval gallery (r) as shown in Eq. 4. D can be used to find the most similar images that minimize the distance.

$$D(i, r) = \sum_{v=1}^{S+1} d(fc_{7v}(i), fc_{7v}(r)') \tag{4}$$

In order to solve attribute manipulation case, we directly replace $fc_{7v}(i)$ representation which corresponds to the undesired attribute of the i and change it with a 'generic' representation of the desired attribute. This generic representation of each attribute is attained by passing images with same attribute label to the fc_{7v} layer and averaging the extracted features. Note that, a normalization operation is used for Eq. 4 to give same importance for each embedding.

5. Personalization of the Fashion Search

The proposed algorithm can produce a learned representation and conduct search with respect to query image and attribute manipulation keywords. However, different people may have quite different opinions when it comes to fashion search. Additionally, users may not even be aware of all attributes. For example, the user may not care about the collar attribute as long as the retrieved image is red colored. There can be infinite examples similar to this. For this purpose, we propose a dynamic relative feedback mechanism on top of the well-defined signature vectors.

$$\sum_{v=1}^{S+1}\left(w_v = \frac{1}{\sum_{k,j} d(fc_{7v}(k), fc_{7v}(j)')}\right) \tag{5}$$

$$D(i,r) = \sum_{v=1}^{S+1} w_v * d(fc_{7v}(i), fc_{7v}(r)') \tag{6}$$

Depending on the picked set of images (k, j) which is decided by the user, Eq. 5 estimates the personalized weights. The idea of Eq. 5 is to check how similar each selected image is in terms of the signature vectors. As each signature vector represents an attribute, the algorithm can now give different importance to the fc_{7v} representations using the estimated weights w_v. By using Eq. 6, the next search can be conducted by considering personal choices (w_v) as well as the features from the fc_{7v} layers. In the relevant search procedure, when only a single image is chosen, the query image is replaced with the chosen image.

6. Implementation Details

Tensorflow software library is used to implement the network. We use the pre-trained AlexNet weights up until conv-5'th layer and reinitialize all other layers. The network is first trained using $L_{attribute}$ loss function for 30 epochs with learning rate 0.01. After that, we use the combination of $L_{attribute} + L_{triplet}$ to train similarity network while keeping the representations that is learned from the first optimization. For the part extraction, tensorflow's `crop_and_resize` function is used to feed cropped regions from conv5 layer to each pooling layer of special attributes. For pattern and color special attributes, chest part of the image is used as it captures most information on those attributes.

As we break triplet similarity learning, selection of $(\hat{i}, i^+, i^-)$ images is easy. For each mini-batch and attribute, the images with the same label are randomly chosen to be $\hat{i}$ and i^+ and i^- is picked from an attribute which does not have the same attribute label. Gradients are calculated for each loss function and joint loss is back propagated to the network. The network is trained with stochastic gradient descent algorithm. We use 80% of the dataset to train the network. Rest images are used to test the network which leaves us with around 21k images. For the retrieval gallery, 19,000 images are reserved and rest 2,000 images are served as the queries.

7. Experiments

Baselines: Since we are using a new dataset, we implement several competing methods along with state-of-the-art AMNET model [28]. We choose to implement 3 more baseline methods for ablation studies. For the first method, attributes of the query image are recognized with classification network. Using the training set, the representation is changed to match query image and the search is performed. We denote this method as "Classic" in the experiments. The second method is the proposed method without the part extraction module (w/o parts) where rather than using ROI pooling whole conv5 feature map is passed through special attribute branches. The last method is same as the proposed work but without using part extraction and triplet ranking loss (w/o parts & ranking).

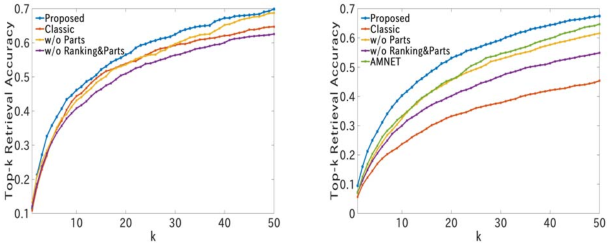
Evaluation Metrics: For retrieval experiments, top-k retrieval accuracy is reported. Top-k retrieval considers the scenario of the algorithm finding an item with exactly the same attributes as demanded by the query and if so it will be a hit (1) otherwise it will be a miss (0). Note that, as attribute labels may be missing some special attributes, we only include results which have a valid attribute label for all special attributes to have realistic results. We do not conduct any recognition experiments as it is not the main task.

Search Strategy: In [5], it is mentioned that searcher (in our case user) is very clear about the search and searcher's session would typically be short, leading to an end-result. Following this, we prepare experiments by allowing maximum 2 attribute manipulations. For each query image, a random special attribute is manipulated and target images in the retrieval gallery are found (if there is any). In some experiments, 2 attribute manipulations are conducted which can be done at once by directly replacing undesired attributes.

7.1. Direct Retrieval

Direct retrieval (search by query) experiments involve extracting signature feature(s) of the query image and comparing it with the retrieval gallery and does not involve any attribute changes. Top-k retrieval accuracy results are given in Figure 6 (a).

The proposed method achieves the best performance, giving 56.5% top-20 retrieval accuracy which is 3% higher



(a) Direct Retrieval

(b) w/ Att. Manipulation

Figure 6: Top-k Retrieval Accuracies of Retrieval (a) and Retrieval After Manipulation (b) Task. For the first task, all methods perform similar to each other as the task is not quite difficult. For the second task, the proposed method shows its strength.

than the nearest competitor. Other competing methods perform similarly to each other, so it is not clear where the main improvement comes from. We will investigate more challenging experiments for the rest of this section.

7.2. Fashion Search with Attribute Manipulation

Retrieval after manipulating some attribute experiments involve replacing signature feature(s) of the query image and comparing it with the retrieval gallery. Top-k retrieval accuracy results are given in Figure 6 (b). The first noticeable finding of this experiment compared to the previous one is the huge accuracy drop in each method which proves difficulty of the task.

The proposed method achieves the best performance again with 53.1% top-20 retrieval accuracy. The second best performing method is AMNET [28] and gives 45.8% top-20 retrieval accuracy. Removing the part extraction from the network decreases top-20 retrieval accuracy by around 7.5% which is a huge deal. AMNET and the method with part extraction perform quite similar to each other. If we were to remove both part extraction and the ranking loss would result in 40.1% top-20 retrieval accuracy which is better than directly using the classic method as it gives 33.2% top-20 retrieval accuracy.

This set of experiments gives us several findings. The first finding is using different layers to train a ranking loss for each attribute does not work so good if there is no part information available however it still works better than classic CNN implementation. Another notable finding is breaking the network into 2 different paths is not quite a good idea without ranking loss and part extraction as it performs worse than current CNN implementations.

Table 2: Top-20 retrieval accuracy performances with different attribute manipulation operations. The proposed method achieves best results for all attributes.

Attribute	Classic	w/o Ranking & Parts	w/o Parts	AMNET [28]	Proposed
Pattern	0.43	0.44	0.48	0.51	**0.54**
Collar	0.35	0.43	0.51	0.39	**0.60**
Fastening	0.14	0.16	0.30	**0.38**	0.35
Sleeve Length	0.23	0.32	0.37	0.37	**0.46**
Color	0.51	0.63	0.67	0.64	**0.72**

In table 2, we also report how much retrieval accuracy is obtained from different special attribute manipulations. The first observation is that our method works quite good for both ”Collar” and ”Sleeve length” attributes. The reason for that is our method can find the exact location of those attributes. Moreover, both of attributes are not disturbed from others in terms of spatial location. For ”Pattern” and ”Color” attributes all methods perform similarly to each other as they are more apparent than the other attributes and easier to learn. Lastly, it is observed that since ”Fastening” attribute is harder to capture all competing methods perform poorly but part extraction has a good effect on this attribute as the spatial location can be captured correctly.

7.3. Visual Examples of Fashion Search

As the attribute manipulation task is more challenging and interesting than the search by query, we provide some attribute manipulation examples in Figure 7. The images with the green bounding box mean that there is an exact match in terms of desired attributes. In the first row, a query image is selected and the user provides some keywords to change ”Hood Collar” attribute to ”Mandarin Collar”. The proposed system finds four images which rank best based on the query image and attribute manipulation. The reason, the second image does not have green bounding-box is because fastening attribute is different than the query image. In the first and third images the undesired attribute is changed while keeping the untouched attributes of the query. Two more examples are provided in the next rows of Figure 7.

7.4. Personalization of the Fashion Search

It is not quite possible to give quantitative results for the personalization of the fashion search. However, with this section, we want to prove that our learned attributes representations are quite discriminative and can be modified just by picking some relevant images from the set of images. Figure 8 is used for this set of experiments.

In the first row of Figure 8, some sample images similar to the query image is provided denoted as ”Retrieved Images”. Followed up by relevance feedback, the user picks the most desired images denoted with the blue bounding box. By using this feedback, the algorithm modifies the parameters of the retrieval formulation and make the search more coherent with the user’s wishes. Lastly, top-3 retrieved images are presented which are in fact in the same direction as user’s opinion. This new search is now tailor-

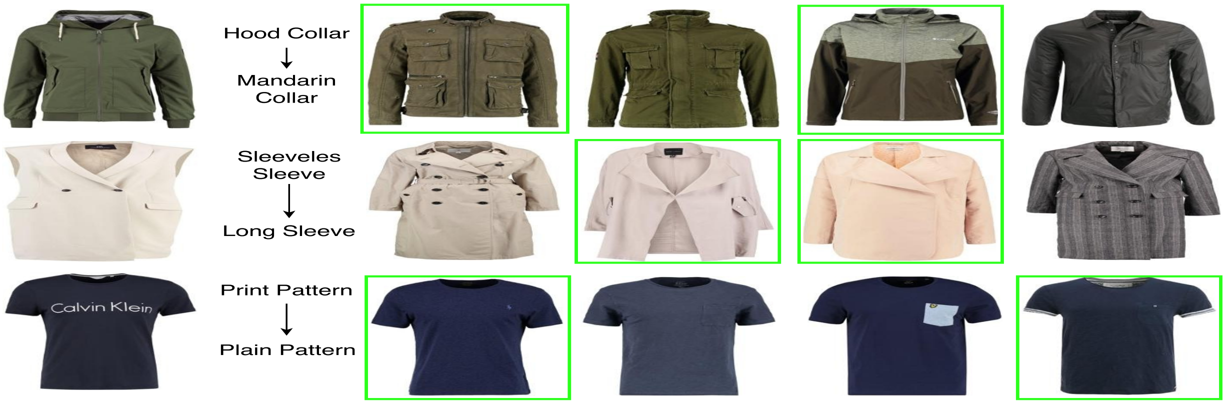


Figure 7: Visual results of attribute manipulations on the Shopping100k dataset. First 2 columns represent query image, attribute manipulation respectively, and the last 4 columns are top-4 retrieved images, ranked from left to right. Correct predictions are shown with the green bounding-box.

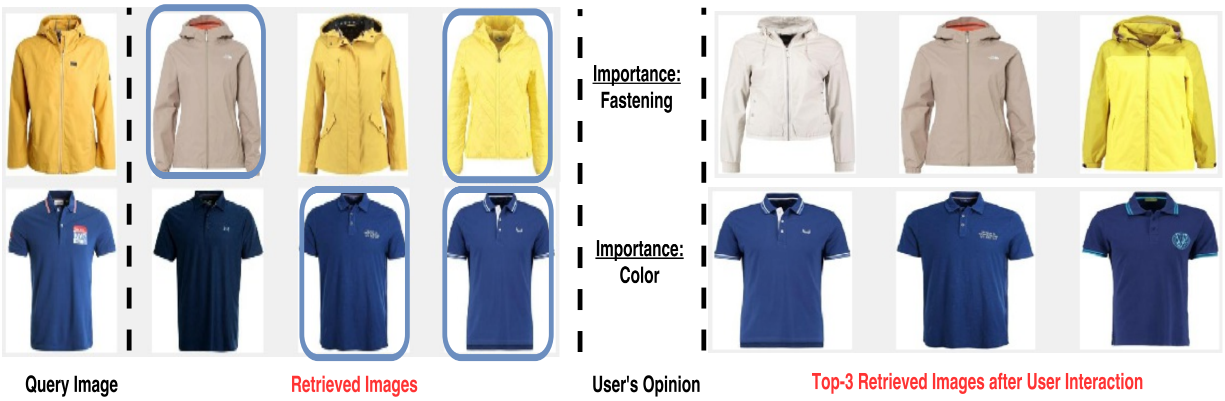


Figure 8: Visual experiments of personalized fashion search. Given a query image, some images are presented to the user (randomly picked from top-10 similar images). In the next step, the user selects some images which he/she has in mind shown in the blue bounding box and the search is conducted again using the acquired information.

made where the user does care about fastening of the item but does not really care about the color similarity. Another example is provided for further investigation.

7.5. Part Extraction

In order to see the effectiveness of the part extraction method. We changed the part extraction method to a regression method which is trained from a small number of images and estimate ”jump detection points”. However, in fashion search proposed and regression-based extraction methods perform identically, therefore we are not reporting any results on different part extraction methods.

7.6. Run Time Performance

Our network is trained on, Intel i7-5820K CPU and 64 GB RAM with GeForce GTX TITAN X GPU. The part extraction method takes about 0.1s per image. In 60 seconds, the network can extract all the necessary features from 7,500 images assuming the part extraction is already performed. The classic method can extract 10,000 images at the same time. The reason for that is the proposed work adds several additional layers on top of the classic implementation, however, the run-time speed is still acceptable.

8. Conclusion

In this paper, we first generated a dataset which was missing from the fashion research. Secondly, we proposed an efficient way to learn attribute similarities from the images with multiple attributes using part extraction method. Our proposed methods enable analysis of dynamic queries and achieve good results in terms of retrieval when applied on our new dataset. Experiments demonstrate that our proposed network outperforms different baselines and is able to adapt towards user’s requests. For our future work, we want to extend our method towards different multi-attribute related computer vision problems.

References

[1] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, pages 663–676. Springer, 2010.

[2] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. In *Asian conference on computer vision (ACCV)*, pages 321–335, 2012.

[3] H. Chen, A. Gallagher, and B. Girod. Describing Clothing by Semantic Attributes. In *European Conference on Computer Vision (ECCV)*, 2012.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference Computer Vision Pattern Recognition (CVPR)*, pages 886–893, 2005.

[5] R. Datta, D. Joshi, J. I. A. Li, and J. Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys (Csur)* , 40(2):1–60, 2008.

[6] Q. Dong, S. Gong, and X. Zhu. Multi-task curriculum transfer deep learning of clothing attributes. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 520–529, 2017.

[7] R. S. Feris, C. Lampert, and D. Parikh. Introduction to visual attributes. In *Visual Attributes*, pages 1–7. Springer, 2017.

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3343–3351, 2015.

[10] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. Automatic spatially-aware fashion concept discovery. *arXiv preprint arXiv:1708.01311*, 2017.

[11] D. Harman. Relevance feedback and other query modification techniques., 1992.

[12] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1062–1070, 2015.

[13] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 105–112, 2013.

[14] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *European conference on computer vision*, pages 472–488, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM international conference on Multimedia*, pages 619–628, 2012.

[17] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *IEEE Conference Computer Vision Pattern Recognition (CVPR)*, pages 3330–3337, 2012.

[18] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *IEEE Conference Computer Vision Pattern Recognition (CVPR)*, pages 1096–1104, 2016.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[20] M. Mizuochi, A. Kanezaki, and T. Harada. Clothing retrieval based on local similarity with multiple images. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1165–1168, 2014.

[21] D. Parikh and K. Grauman. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE, 2011.

[22] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. *Computer Vision–ECCV 2012*, pages 876–889, 2012.

[23] O. Russakovsky and F.-F. Li. Attribute learning in large-scale datasets. In *ECCV Workshops (1)*, volume 6553, pages 1–14, 2010.

[24] E. Simo-Serra and H. Ishikawa. Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction. In *IEEE Conference Computer Vision Pattern Recognition (CVPR)*, pages 298–307, 2016.

[25] Y. Song, Y. Li, B. Wu, C.-Y. Chen, X. Zhang, and H. Adam. Learning unified embedding for apparel recognition. *arXiv preprint arXiv:1707.05929*, 2017.

[26] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. *Computer Vision–ECCV 2010*, pages 155–168, 2010.

[27] M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2937–2940, 2011.

[28] B. Zhao, J. Feng, X. Wu, and S. Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *IEEE Conference Computer Vision Pattern Recognition (CVPR)*, 2017.