Fashion Retrieval via Graph Reasoning Networks on a Similarity Pyramid

Yiming Gao, Zhanghui Kuang, *Member, IEEE,* Guanbin Li, *Member, IEEE,* Ping Luo, *Member, IEEE,* Yimin Chen, Liang Lin, *Senior Member, IEEE,* and Wayne Zhang

Abstract—Matching clothing images from customers and online shopping stores has rich applications in e-commerce. Existing algorithms mostly encode an image as a global feature vector and perform retrieval via global representation matching. However, distinctive local information on clothing is immersed in this global representation, resulting in sub-optimized performance. To address this issue, we propose a novel Graph Reasoning Network (GRNet) on a similarity pyramid, which learns similarities between a query and a gallery cloth by using both initial pairwise multi-scale feature representations and matching propagation for unaligned representations. The query local representations at each scale are aligned with those of the gallery via an adaptive window pooling module. The similarity pyramid is represented by a similarity graph, where nodes represent similarities between clothing components at different scales, and the final matching score is obtained by message propagation along edges. In GRNet, graph reasoning is solved by training a graph convolutional network, enabling the alignment of salient clothing components to improve clothing retrieval. To facilitate future research, we introduce a new benchmark, *i.e.* FindFashion, containing rich annotations of bounding boxes, views, occlusions, and cropping. Extensive experiments show that GRNet obtains new state-of-the-art results on three challenging benchmarks, *e.g.* pushing the accuracy of top-1, top-20, and top-50 on DeepFashion to 27%, 66%, and 75% (*i.e.* 6%, 12%, and 10% absolute improvements), outperforming competitors with large margins. On FindFashion, GRNet achieves considerable improvements on all empirical settings.

Index Terms—Fashion retrieval, graph reasoning, similarity pyramid.

1 Introduction

Fashion image retrieval between customers and online shopping stores has various applications for e-commerce. Given a street snapshot of a clothing image, this task requires searching for the same garment item in an online store. It is a key step for further applications such as generating descriptions of clothing, brands, materials, and styles. While matching clothing across modalities appears effortless for human vision, it is extremely challenging for machine vision. The same clothing may exhibit large variations due to occlusions, cropping, and viewpoint changes. Moreover, garments may even differ in only small local regions, such as logos.

The task of customer-to-shop clothing retrieval has witnessed tremendous progress [1], [2], [3], [4], [5], [6], [7], [8], [9] due to the advancement of convolutional neural networks (CNNs) [10], [11], [12], [13], [14]. Existing methods mostly employ a global similarity pipeline. For example, they first aggregate local features into compact global features and then compute global similarities between query and gallery images by using cosine or Euclidean distance (see Figure 1 (a)). In the procedure of global feature aggregation, the distinctive local regions of clothing would be suppressed. In contrast, human vision verifies whether two clothing items are the same by simultaneously comparing the query and the gallery in terms of both global features such as fabric, colors, textures and categories (*e.g.* "dress" or "t-shirt"), as well as local

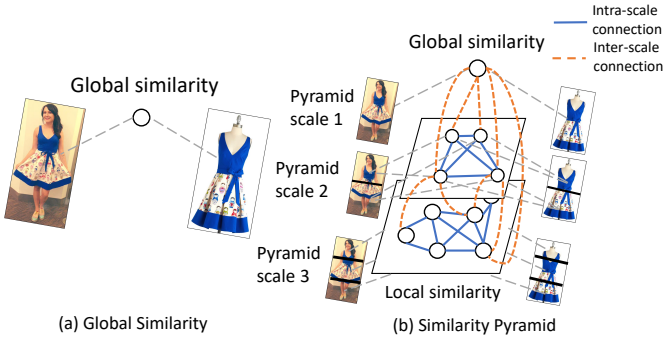


Fig. 1: Comparison between global similarity and a similarity pyramid with graph reasoning. The left illustrates the global similarity. The right panel shows the similarity pyramid with graph reasoning, where scale 1 computes the global similarity, and scales 2 and 3 include local similarities between all possible combinations of local patches from each image pair. The dashed gray line indicates that the similarity is calculated from two image patches. The pyramid similarities (including the global and the local) are reasoned mutually. The blue lines indicate interactions between similarities within the same scale while the red dashed lines indicate those from two different scales (best viewed in color).

- *Yiming Gao, Guanbin Li, and Liang Lin are from the School of Data and Computer Science, Sun Yat-sen University, China; Zhanghui Kuang, Yimin Chen, and Wayne Zhang are from Sensetime Research; Ping Luo is from The University of Hong Kong.*
- *Yiming Gao and Zhanghui Kuang contributed equally to this work.*
- *Corresponding author: Wayne Zhang and Guanbin Li*
  E-mail: wayne.zhang@sensetime.com. liguanbin@mail.sysu.edu.cn

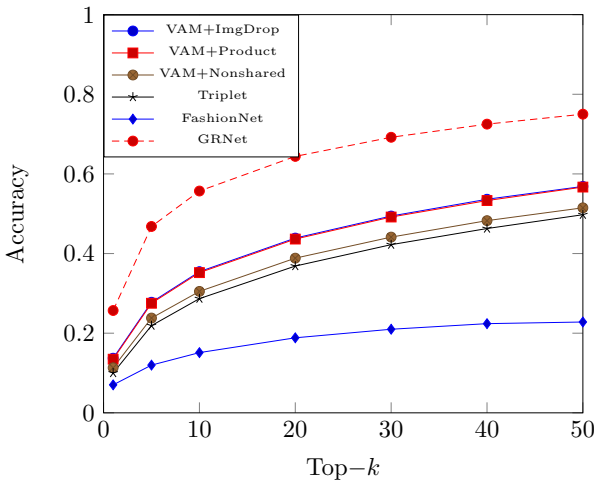


Fig. 2: Comparison with state-of-the-art methods on the DeepFashion consumer-to-shop dataset [3]. ImgDrop+GoogleNet and Product+GoogleNet are the two best results ever reported [15].

features such as sleeve, collar, and logos. Moreover, human vision instinctively focuses on common parts between the query and the gallery, while ignoring those regions that only exist in the query (or the gallery) due to occlusions, cropping or viewpoint changes. We propose that for clothing retrieval and verification, comparing clothing in both global and local patterns is complementary.

Inspired by the procedure above, we design a novel *Graph Reasoning Network (GRNet)* on a *similarity pyramid* to compare a query and a gallery image both globally and locally at different similarity scales. As illustrated in Figure 1 (b), we extract CNN features for all spatial regions at each pyramid scale. A critical issue for matching two clothing is that the local clothing regions are often mismatched. To deal with the misalignment between the query and the gallery, we need to enumerate all the region pairs on the same scale to calculate their similarities. However, as the local regions are not equally important, similarities between aligned regions should be dominant, while those between misaligned pairs should be ignored.

Thus, we construct a pyramid defined by similarities between clothing regions. This *similarity pyramid* can be formulated as a *graph*, where each node of the graph is the similarity between two corresponding clothing regions on the same scale, while each edge is the normalized similarity of its connected nodes. The final similarity (matching score) between a query and a gallery image can be achieved by reasoning on this graph. GRNet contains a key component of a graph convolutional network (GCN), which performs graph reasoning by propagating messages among the nodes.

The proposed GRNet greatly suppresses the performance degradation caused by occlusions, cropping, viewpoint changes and small logos, outperforming existing methods with large margins as shown in Figure 2. Specifically, on the DeepFashion [3] benchmark, GRNet significantly improves the accuracy of top-1, top-20, and top-50 of the best ever reported results (VAM+Product [15]) by 13%, 22% and 18%, and those of our re-implemented state-of-the-art matching method [16] by 6%, 12%, and 10% respectively. On the Street2Shop [2] and DeepFashion2 [17] benchmark, GRNet outperform the previous methods by a huge margin.

Although existing benchmarks such as Street2Shop [2], DARN [1], and DeepFashion [3] have progressed the researches of customer-to-shop clothing retrieval, and the detailed annotations of occlusions, cropping and views are limited, impeding ablation studies of this task. They are also unsuitable for analyzing the influence of image variations on retrieval performance.

Therefore, we build a new customer-to-shop clothing retrieval benchmark, named *FindFashion*, by revisiting existing datasets and annotating attributes in terms of occlusions, cropping, and views. FindFashion allows in-depth analysis of the impacts of possible appearance variation on clothes retrieval. We further introduce four new evaluation protocols of varying difficulties, including *Easy*, *Hard-View*, *Hard-Occlusion*, and *Hard-Cropping*. The training, validation, and test set splits on FindFashion will be released for fair comparisons.

A preliminary version of this work is published in ICCV 2019 [18]. In this work, we inherit the idea of graph reasoning on a similarity pyramid, but extend the conference version in several aspects. *First*, we redesign multi-scale local feature extraction. Instead of fixed spatial windows, the multi-scale local features of queries are extracted with dynamic spatial windows with the help of a tailor-designed adaptive window pooling module. The dynamic spatial windows of the query are adjusted according to the gallery so that local features of the query are aligned with their corresponding ones of the gallery. As validated in our experiments, this strategy consistently improves the final retrieval performance. *Second*, we expand the original *FindFashion* with the recently released DeepFashion2 [17] to a new benchmark, on which we also perform experiments to investigate how cropping, views, and occlusions affect the fashion retrieval performance. *Third*, we conduct additional experiments on the benchmark DeepFahion2 [17] comparing with other state-of-the-art approaches. *Fourth*, we explore the application of GRNet in large scale scenarios. We initially search efficiently the gallery clothes with the global similarity, and obtain a list of candidates, which is re-ranked via our GRNet with negligible runtime and computational overhead. *Finally*, we provide a deeper analyses of the results. *e.g.*, the performances of our GRNet on FindFashion with different subsets of training dataset. We also analyse the underlying reason of performance differences with different settings to facilitate fashion retrieval research in the future.

Our main **contributions** are summarized in three aspects.

- We propose an effective approach for clothing retrieval, *Graph Reasoning Network (GRNet)* on a *similarity pyramid*. GRNet computes similarities between a query and a gallery image at different local clothing regions and scales. GRNet has an important component of the graph convolutional neural network to propagate similarities on the pyramid, performing graph reasoning and producing state-of-the-art performance.
- We validate the effectiveness of GRNet on three popular datasets, DeepFashion, DeepFashion2 and Street2Shop. GRNet outperforms state-of-the-art methods with significantly large margins.
- We annotate different variations and build the new customer-to-shop retrieval benchmarks named FindFashion, which allows the in-depth analysis of the effect of variations on clothing retrieval. Extensive experiments demonstrate that GRNet is more robust against occlusions, cropping, or non-front views than previous methods.

2 Related Work

Datasets	Street2Shop [2]	DeepFashion [3]	DeepFashion2 [17]	**FindFashion**
#images	416,840	239,557	491,895	565,041
#pairs	39,479	195,540	873,234	382,230
Public split	✓	✓	✓	✓
Bbox	✓	✓	✓	✓
View	✗	✗	✓	✓
Occlusion	✗	✗	✓	✓
Cropping	✗	✗	✓	✓

TABLE 1: Comparison of customer-to-shop clothing retrieval datasets.

Clothing retrieval. Pioneer work [7], [19], [20], [21] on clothing retrieval utilized conventional features such as SIFT and semantic preserving visual phrases. Recently, deep neural networks have been widely applied in clothing retrieval and have pushed research into a new phase [1], [2], [3], [4], [5], [6], [8], [9], [17]. These methods usually follow a global similarity computation and matching pipeline, *i.e*. aggregating local features into a single global representation and then performing similarity computation. [1], [3] explored attributes via multi-task learning to learn representations that are related to specific tags such as "crew neck", "short sleeves" and "rectangle-shaped"; [17] made full use of clothing landmark to improve fashion retrieval using the Mask R-CNN [22] framework. [2], [23] investigated different network architectures which are adept at extracting global features for customer-to-shop clothing retrieval. Instead, [6], [9] attempted to train models with weakly or noisy supervised signals to reduce the dependency of data annotation and increase the global feature learning efficiency. Recently, [4] utilized attribute labels to focus on local discriminative regions. Similarly, [15] focused on clothing regions and ignored cluttered backgrounds via a cloth parsing subnetwork. Both works employed attention mechanisms in global feature aggregation to suppress local distractive regions and upweight the discriminative regions to some extent. However, they were highly dependent on explicit knowledge, such as label and clothing parsing category definitions which might be unavailable in real application scenarios. In contrast, we conduct clothing matching computation via pyramid similarity (including both global and local similarities) learning on a relation graph, which can obtain salient component alignment through similarity propagation, and thus achieve more accurate matching. Notably, the proposed approach achieves similarities weighting by end-to-end classification training without any explicit supervised signals. There also exist some variants, such as dialog-based clothes search [24] , video-based clothes retrieval [8], and attribute feedback-based clothes retrieval [25], [26]. Their application scenarios and settings are different from ours.

Customer-to-shop clothing retrieval datasets. Some customer-to-shop clothing retrieval datasets exist, as listed in Table 1. Kiapour *et al*. [2] collected the Street2Shop dataset from a large online retail store. It consists of 78,958 images, 39,479 customer-to-shop pairs, and 396,483 gallery images. Huang *et al*. [1] collected the DARN dataset, which is composed of upper-clothing images. It has 182,780 images, 91,390 pairs, and 91,390 gallery images, in which only query images are bounding boxes. However, the training/testing split is not available and thus prevents other research from making a fair comparison. Liu *et al*. [3] released the DeepFashion dataset. It has 239,557 images, 195,540 customer-to-shop pairs, and 45,392 gallery images. It is later revisited for fine-grained attribution recognition [27]. All the above datasets lack detailed attributes that are most related to clothing retrieval performance. Our benchmark FindFashion contains detailed attribute annotations (*e.g*. views, occlusions and cropping) so that the impacts of attributes on retrieval performance can be analyzed in detail. We noticed that Ge *et al*. [17] recently released the DeepFashion2 dataset with 491,895 images and 873,234 pairs, which is concurrent with our work. We also noticed that there exist other clothing datasets such as [28], [29], [30], [31], [32] and [33]. These datasets mainly target at clothing segmentation, attribution prediction and fashion comments but not customer-to-shop clothing retrieval and lack clothing pairs for evaluation. [25] released Fashion 200k, which aims at attribution discovery and clothing retrieval with attribute manipulation and is very different from our task.

Graph reasoning. Graphs naturally model the dependencies between concepts, which facilitates research on graph reasoning, such as Graph CNN [34], [35], [36], and Gated Graph Neural Network (GGNN) [37]. These graph neural networks have been widely employed in various tasks of computer vision and have made very promising progress, *e.g*. object parsing [38], [39], multi-label image recognition [40], visual grounding [41], [42], [43], social relationship understanding [44], facial action unit recognition [45] and action recognition [46]. These studies create knowledge graph based on the relationship between different entities, *e.g*. images, objects, proposals, and semantic categories. Instead, we are the first to explore the use of a knowledge graph to represent the similarity between different pairs of local regions, and apply it to a new field of customer-to-shop clothing retrieval. It can facilitate the weighting of local region pairs and the enhancement of global matching through the iteration of propagation between pyramid similarities relations, and thus obtain more accurate matching computation.

Image retrieval. Our work is related to image retrieval approaches [47], [48], [49], [50], [51], [52], [53], [54], [55]. They target at retrieving rigid objects such as buildings or scenes, and often aggregate regional features into compact representations to compute global similarities. Different from them, our GRNet aims at retrieving more challenging non-rigid clothes. Moreover, our GRNet captures both local and global similarities and conducts graph reasoning on a similarity pyramid.

Metric learning. Our work is also related to general deep metric learning [56], [57], [58], [59], [60]. However, they only conducted experiments on the InShop clothing retrieval dataset, while our work focuses on customer-to-shop clothing retrieval which is much more challenging as analyzed in [3]. We have also compared the proposed GRNet with the state-of-the-art method [60] in our experiments.

Spatial transformation. Spatial transformation has been widely adopted for geometric matching [61] and facilitating various vision tasks, such as image classification [62], [63], person re-identification [16], [64], and scene text recognition [65]. Geometric matching [61] estimates the spatial transformation with supervised learning, while we do it in a weakly-supervised fashion without direct supervision signals. Compared with previous work [62], [64], [65], our adaptive window pooling differs in design: 1) they learn the spatial transformation of the input image so that the input image is normalized to one hidden canonical image, while we learn to spatially transform the query so that it can be well aligned with the gallery; 2) they take an image as input and then transform it, while we take correlation maps between the query and the gallery as input and perform transformation on the

query only.

Multi-scale representations. Multi-scale features over an image or region are useful for various of computer vision tasks, *e.g.*, image retrieval [47], [66], visual recognition [67], text detection [68] and semantic segmentation [69], [70]. The methods in [66], [67], [69] extract multi-scale features via multi-scale pooling, and then utilize the multi-scale features via concatenation. In contrast, we utilize both multi-scale features and multi-scale similarities. Moreover, we propose the graph reasoning network to capture the relations between the global and local similarities instead of just concatenating multi-scale features.

Image re-ranking. Image re-ranking [71], [72], [73], [74] is widely performed for instance search, which has been investigated for decades. They leverage the distribution of gallery set to re-rank the retrieval results. In contrast, we utilize the information of the query-gallery pair only without the help of the gallery set manifold. Therefore, our proposed GRNet is orthogonal to those re-ranking methods and can be combined with them for further performance improvement.

3 METHODOLOGY

3.1 Motivation

The setup of the customer-to-shop clothing retrieval is as follows. Given one customer clothing image query $\mathbf{x}$ and one shop clothing gallery set $\mathbb{G} = \{\mathbf{y}\}$, it computes the similarities s between $\mathbf{x}$ and $\mathbf{y}$ and ranks them. $\mathbf{x} = \{\mathbf{x}^i\}$ and $\mathbf{y} = \{\mathbf{y}^i\}$, where $\mathbf{x}^i \in \mathbb{R}^{C\times 1}$ and $\mathbf{y}^i \in \mathbb{R}^{C\times 1}$ are local features of the customer clothing image and the shop image, respectively. Previous customer-to-shop clothing retrieval approaches [1], [2], [3], [4], [5], [6], [7], [8], [9] adopt the following global similarity:

$$s_g = S_g(A(\mathbf{x}), A(\mathbf{y})), \tag{1}$$

where $A(\cdot)$ is the aggregation function and $S_g(\cdot,\cdot)$ is the scalar global similarity function. The aggregation function is usually the average pooling or max-pooling operator. The similarity function often adopts the cosine similarity or Euclidean distance. Ordinarily, the global similarity can reliably estimate the similarity between the query and the gallery. However, the aggregation function might aggregate noisy features such as clutter background, other objects, or unique regions that can only be observed in the query or the gallery when existing occlusions, cropping or different views. This undoubtedly greatly degrades clothing retrieval performance.

To address the above issues, [75], [76] computed the similarity between the query and the gallery by summing local similarities between local feature pairs with a greedy strategy as follows:

$$s_l = \sum_{i,j} w_l^{ij} S_l(\mathbf{x}^i, \mathbf{y}^j), \tag{2}$$

where $S_l(\cdot,\cdot)$ is the scalar local similarity function, and w_l^{ij} is the scalar weight of local similarities $S_l(\mathbf{x}^i, \mathbf{y}^j)$, which is given by

$$w_l^{ij} = \begin{cases} 1, & \text{if } j = \text{argmax}_k(S_l(\mathbf{x}^i, \mathbf{y}^k)). \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

However, greedily finding local feature pairs easily leads to misalignment, which accumulates errors in the final estimated similarity s_l.

We attempt to make full use of both the global and local similarities, and learn the importance of different similarities (*i.e.*w_l^{ij}) automatically to mitigate the above issues.

3.2 Graph Reasoning Network

For each query (or gallery) image, instead of extracting local features $\mathbf{x}^i$ (or $\mathbf{y}^i$) and global features $A(\mathbf{x})$ (or $A(\mathbf{y})$), we extract multi-scale features at pyramid spatial windows, and obtains $\{\mathbf{x}_l^i \in \mathbb{R}^{C\times 1}\}$ (or $\{\mathbf{y}_l^i \in \mathbb{R}^{C\times 1}\}$) with $\mathbf{x}_l^i$ (or $\mathbf{y}_l^i$) being the i^{th} local feature for pyramid scale l, where $l \in \{1, \cdots, L\}$ indicates the scale index from the top down. Therefore, $\mathbf{x}_1^1$ and $\mathbf{y}_1^1$ refer to the global feature vector of the query and that of the gallery (*i.e.*, $A(\mathbf{x})$ and $A(\mathbf{y})$) respectively. For each scale l, assuming there exist $R_l \times C_l$ local spatial windows for each image, we have a total of $\sum_l R_l C_l$ pyramid features.

Similarity pyramid graph. We build a similarity pyramid graph with all region pair similarities being the graph nodes, and the relations between two similarities being the edges. Formally, given a pair of local features $\mathbf{x}_l^i$ and $\mathbf{y}_l^j$ from the same pyramid scale l, we compute their similarity vector $\mathbf{s}_l^{ij} \in \mathbb{R}^{D\times 1}$ instead of a similarity scalar in Equation 1 and 2, by a vector similarity function given by

$$S_p(\mathbf{x}_l^i, \mathbf{y}_l^j) = \frac{\mathbf{P}\left|\mathbf{x}_l^i - \mathbf{y}_l^j\right|^2}{\left\|\mathbf{P}\left|\mathbf{x}_l^i - \mathbf{y}_l^j\right|^2\right\|_2}, \tag{4}$$

where $|\cdot|^2$ and $\|\cdot\|_2$ indicate the element-wise square and l_2-norm respectively. $\mathbf{P} \in \mathbb{R}^{D\times C}$ is a projection matrix which projects pyramid feature difference vectors from the C dimension to a lower D dimension. Similarity vectors are guaranteed to have the same magnitude by performing l_2-normalization. For any node pair in the graph $\mathbf{s}_{l_1}^{ij}$ and $\mathbf{s}_{l_2}^{mn}$, we define a scalar edge weight $w_p^{l_1 ij, l_2 mn}$, which is given by

$$w_p^{l_1 ij, l_2 mn} = \frac{\exp((\mathbf{T}_{out}\mathbf{s}_{l_1}^{ij})^\intercal(\mathbf{T}_{in}\mathbf{s}_{l_2}^{mn}))}{\sum_{l,p,q}\exp((\mathbf{T}_{out}\mathbf{s}_{l_1}^{ij})^\intercal(\mathbf{T}_{in}\mathbf{s}_{l}^{pq}))}, \tag{5}$$

where $\mathbf{s}^\intercal$ indicates the transpose of the vector $\mathbf{s}$. $\mathbf{T}_{in} \in \mathbb{R}^{D\times D}$ and $\mathbf{T}_{out} \in \mathbb{R}^{D\times D}$ are the linear transformations of incoming edges and outgoing edges for each graph node respectively. When $l_1 = l_2$, $w_p^{l_1 ij, l_2 mn}$ are intra-scale edges. *i.e.*, their two connected similarity nodes come from the same scale. When $l_1 \neq l_2$, $w_p^{l_1 ij, l_2 mn}$ are inter-scale edges. *i.e.*, their two nodes come from different scales. Inter-scale edges enable similarities with different scales to propagate messages from each other. In this way, the similarity pyramid graph is defined as $G = (\mathbb{N}, \mathbb{E})$, where $\mathbb{N} = \{\mathbf{s}_l^{ij}\}$ and $\mathbb{E} = \{w_p^{l_1 ij, l_2 mn}\}$.

Similarity reasoning. We reason the similarity $\mathbf{s}_l^{ij}$ by conducting a sequence of similarity propagation, linear transformation, and non-linear activation operator. Concretely, similarity is first propagated as

$$\hat{\mathbf{s}}_{l_1}^{ij} = \sum_{l_2,m,n} w_p^{l_1 ij, l_2 mn}\mathbf{s}_{l_2}^{mn} \tag{6}$$

$$= \sum_{l_2,m,n} w_p^{l_1 ij, l_2 mn} S_p(\mathbf{x}_{l_2}^m, \mathbf{y}_{l_2}^n). \tag{7}$$

Then, the linear transformation and the non-linear activation are conducted as

$$\mathbf{h}_{l_1}^{ij} = \text{ReLU}(\mathbf{W}\hat{\mathbf{s}}_{l_1}^{ij}), \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{C'\times D}$ is the learnable parameters. Equations (6) and (8) can be easily implemented by graph convolution [35],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3025062, IEEE Transactions on Pattern Analysis and Machine Intelligence

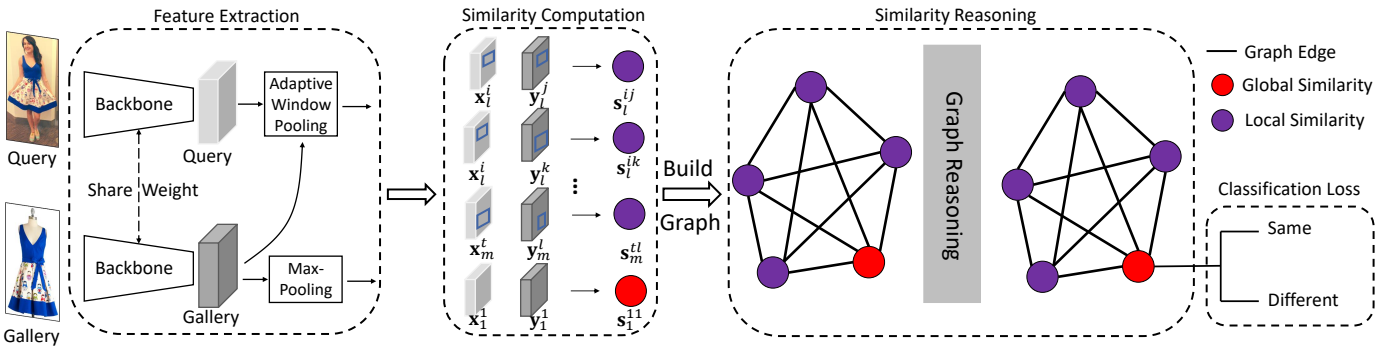


Fig. 3: The overall framework of the proposed GRNet. Given one query and gallery pair, their features extracted by deep convolutional networks are fed into Similarity Computation to build a similarity pyramid graph with all region pair similarities being the graph nodes. In the Feature Extraction, the local features of query are dynamically extracted via the adaptive window pooling based on the features of query and gallery while those of gallery are extracted via max-pooling. In the Similarity Computation, $\mathbf{x}_l^i$ is the i^{th} local feature of the query at scale l while $\mathbf{y}_l^j$ is the j^{th} one of the galleries, and $\mathbf{s}_l^{ij}$ is their similarity vector. Furthermore, the global and local similarities are propagated and updated via Similarity Reasoning. It finally outputs whether the input image pair belongs to the same clothing.

followed by the nonlinear ReLU. We iteratively reason the similarity pyramid T times by setting $\mathbf{s}_{l_2}^{mn}$ on the right-hand side of Equation (6) at the current step to $\mathbf{h}_{l_2}^{mn}$ from the previous step.

3.3 Adaptive Window Pooling

For each scale l, we extract $R_l \times C_l$ local features for each image. The preliminary version of this work [18] divides each image to $R_l \times C_l$ equally and extracts one C dimensional local feature for each window via max-pooling. Therefore, the $R_l \times C_l$ spatial windows are fixed for all images. Although it is simple, local features of the query and the gallery under the same scale fixed windows might be misaligned as the query clothing and the gallery clothing are simply cropped according to bounding boxes without alignment during preprocessing, and the salient components may be divided into different windows. This kind of misalignment could introduce errors when computing local similarities and damage the representation of local features. In this section, we propose adaptive window pooling to mitigate the misalignment so that the spatial windows of the query for local features are adjusted adaptively according to those of the gallery with fixed windows, as illustrated in Figure 4.

To achieve adaptive window pooling, we first compute the correlation map between the feature maps of the query and the gallery, which is used to predict four vertices for each window, and then rectify original feature maps to target feature maps via Thin Plate Spline (TPS) transformation [77]. The local features are finally extracted by max-pooling on the target feature maps. Formally, given the query feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ and the gallery feature amp $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$, we first compute the correlation map $\mathbf{F}_c \in \mathbb{R}^{H\times W\times(H\times W)}$ and predict one normalized vertex coordinate matrix $\mathbf{V}_l \in [0,1]^{2\times N_l}$ with N_l being the number of window vertices for scale l based on $\mathbf{F}_c$. Given the predefined target vertex coordinate matrix $\mathbf{V}_l^t \in \mathbb{R}^{2\times N_l}$, we then estimate one TPS transformation $\mathbf{T}_{tps}^l$, which is used to rectify the original query feature map $\mathbf{X}$ to one target feature map $\mathbf{X}_l^t \in \mathbb{R}^{H\times W\times C}$. The ith local feature at the lth scale $\mathbf{x}_l^i$ is extracted at the ith windows defined by $\mathbf{V}_l^t$ over the feature map $\mathbf{X}_l^t$. To this end, we design one adaptive window pooling module consisting of a correlation unit, vertex prediction unit, TPS transformation unit, and average pooling layer, as shown in Figure 6.

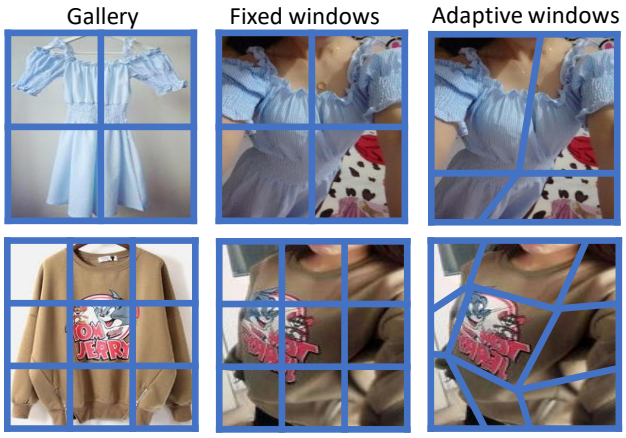


Fig. 4: Illustration of adaptive windows. The first and second rows show examples with the scale of 2×2 and 3×3, respectively. The first column shows the fixed spatial windows of the gallery. The second column shows the fixed spatial windows of the query in [18], while the third's shows the proposed adaptive windows, which are adjusted based on the content of the gallery.

3.3.1 *Correlation Unit*

Since adaptive window pooling aims at adjusting the spatial windows of the query according to the content of the gallery, we embed the correlation between local features of the query and the gallery by calculating spatial position similarities between the query and gallery feature maps inspired by [61]. Compared with element-wise addition or multiplication, correlation can capture the spatial relation between the query and the gallery, as proven in [55].

Mathematically, given the query and gallery feature maps $\mathbf{X}$ and $\mathbf{Y}$, the correlation unit outputs the correlation map $\mathbf{F_c} \in \mathbb{R}^{H\times W\times(H\times W)}$ by calculating the local feature similarity

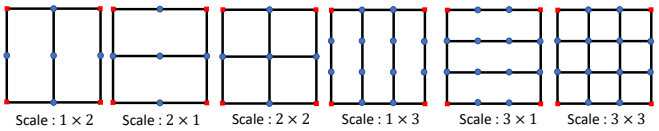


Fig. 5: Vertices which need to be predicted or are fixed for different scales. The blue points are the spatial window vertices on the feature map $\mathbf{X}$ of the query which need to be predicted and thus can be adjusted adaptively. We predict 5, 5, 5, 12, 12, and 12 vertices for scale 1×2, 2×1, 2×2, 1×3, 3×1, and 3×3 respectively. The red points indicate four corners of the query which are fixed for all scales.

Name	Input	Kernel Size	Channels
conv-1	Correlation maps	3×3	32
conv-2	conv-1	3×3	64
maxpool-1	conv-2	2×2	64
conv-3	maxpool-1	3×3	128
conv-4	conv-3	3×3	256
maxpool-2	conv-4	3×3	256
fc-1	maxpool-2	1×1	512
fc-2	fc-1	1×1	$\sum_l N_l \times 2$

TABLE 2: Architecture of the vertex prediction unit.

between the query and the gallery at different position combinations, which is given by

$$f_c^{ijk} = \frac{\mathbf{y}^{ij\intercal}}{\left\|\mathbf{y}^{ij}\right\|^2} \frac{\mathbf{x}^{i_k j_k}}{\left\|\mathbf{x}^{i_k j_k}\right\|^2}, \tag{9}$$

where f_c^{ijk} is the scalar element of $\mathbf{F}_c$ at position (i, j, k). $\mathbf{y}^{ij}$ and $\mathbf{x}^{i_k j_k}$ are the local feature vector of tensor $\mathbf{Y}$ at spatial position (i, j) and that of $\mathbf{X}$ at spatial position (i_k, j_k). $k = Wi_k + j_k$ is an auxiliary indexing variable for (i_k, j_k). Thus, the scalar f_c^{ijk} represents the similarity between the local feature of the gallery at position (i, j) and that of the query at position (i_k, j_k).

3.3.2 Vertex Prediction Unit

Given $\mathbf{F}_c$ as input, the vertex prediction unit predicts $\mathbf{V}_l \in [0, 1]^{2 \times N_l}$ for the lth scale with N_l being the number of vertices predicted. In total, it outputs $\sum_l N_l$ normalized 2-dimensional coordinates. Figure 5 shows the detailed configuration of the vertices that need to be predicted.

The vertex prediction unit is a lightweight convolutional neural subnetwork. It consists of four convolutional layers, each of which is followed by one ReLU, two max-pooling layers with stride 2, one global max-pooling layer, and two full-connected layers with one ReLU between them. The detailed architecture of the vertex prediction unit is listed in Table 2.

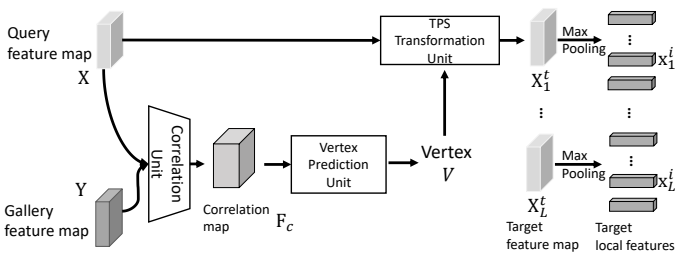


Fig. 6: Architecture of the adaptive window pooling module.

3.3.3 TPS Transformation Unit

For each scale l, we rectify the original query feature map $\mathbf{X}$ to one target feature map $\mathbf{X}_l^t$, where the max pooling can be conducted to extract local features with the help of the TPS Transformation Unit.

For each vertex in $\mathbf{V}_l$, we would like to transform it to its corresponding vertex in one target vertex matrix $\mathbf{V}_l^t$, which are set to vertices of fixed spatial windows of the gallery at scale l. In addition, the coordinates of four corner vertices of the query (*i.e.*, $[0, 0]^\intercal$, $[0, 1]^\intercal$, $[1, 0]^\intercal$ and $[1, 1]^\intercal$) remain unchanged after transformation as shown in Figure 5. Concatenating the four corner vertices with $\mathbf{V}_l$ and $\mathbf{V}_l^t$, we obtain augmented predicted vertex matrix and augmented target vertex matrix $\overline{\mathbf{V}}_l$ and $\overline{\mathbf{V}}_l^t$ respectively. In this way, the tuple $\langle \overline{\mathbf{V}}_l, \overline{\mathbf{V}}_l^t \rangle$ forms the control points of the TPS transformation.

As performed in [65], [77], we parameterize the TPS transformation as:

$$\mathbf{z}_{li} = \mathbf{T}_l \begin{bmatrix} 1 \\ \mathbf{z}_i^t \\ \phi(||\mathbf{z}_i^t - \overline{\mathbf{v}}_{l1}^t||_2) \\ \vdots \\ \phi(||\mathbf{z}_i^t - \overline{\mathbf{v}}_{l(N_l+4)}^t||_2) \end{bmatrix}, \tag{10}$$

where $\mathbf{z}_i^t \in [0, 1]^{2 \times 1}$ is one sampling 2-dimensional position coordinate of the target feature map, $\phi(x) = x^2 \log(x)$ is the radial basis kernel. $\mathbf{T}_l \in \mathbb{R}^{2 \times (N_l + 7)}$ is the TPS transformation matrix for scale l. $\mathbf{z}_{li} \in [0, 1]^{2 \times 1}$ is the 2-dimensional coordinate of the position of the source feature map, which corresponds to $\mathbf{z}_i^t$. Inspired by [65], we have one close-form solution of $\mathbf{T}_l$ by substituting for all vertex pairs of $\langle \overline{\mathbf{V}}_l, \overline{\mathbf{V}}_l^t \rangle$ in Equation (10) and solving a linear system.

Once we have $\mathbf{T}_l$ in Equation (10), we compute $\mathbf{X}_l^t$ via bilinear interpolation as follows:

$$\mathbf{X}_l^t(\mathbf{z}_i^t) = B(\mathbf{X}, \mathbf{z}_{li}), \tag{11}$$

where B is the bilinear interpolation and $\mathbf{X}_l^t(\mathbf{z}_i^t)$ indicates the local feature vector of $\mathbf{X}_l^t$ at position $\mathbf{z}_i^t$.

The bilinear interpolation, the TPS solving and the transformation process are differentiable [62]. Therefore, the TPS transformation unit can be inserted into deep neural networks that can be trained end-to-end.

3.4 Network training

Network architecture. Figure 3 illustrates the overall framework of the proposed graph reasoning network. It consists of four modules: feature extraction, similarity computation, similarity reasoning and classification loss. In the feature extraction module, we employ CNN (*e.g.*, GoogleNet [78] and ResNet [13]) as the backbone, and extract pyramid features on its last convolution activation. The query pyramid features are pooled over adaptive windows while the gallery ones over predefined spatial windows with different pyramid sizes. Both the query and gallery share the same feature extraction backbone. In the similarity computation module, we compute the similarity between all possible local feature combinations between the query and the gallery at the same pyramid scale. In the similarity reasoning module, we employ a stack of graph convolution and ReLU operators.

End-to-end training. We use the cross-entropy loss over the final reasoned global similarity vector (*i.e.*, $\mathbf{h}_1^{11}$) and the ground

Setups	*E*	*HO*	*HC*	*HV*
#Validation	125863	4920	15883	47164
#Test	30746	1250	3883	11383

TABLE 3: Statistics of query pairs of four evaluation setups on FindFashion.

truth $\bar{s}$ corresponding to the query and the gallery $(\mathbf{x}, \mathbf{y})$. We train the whole network including the adaptive window pooling module end-to-end with the cross-entropy loss. In this way, adaptive windows of queries, similarities, and the importance of each local region are jointly learned.

3.5 Large Scale Fashion Retrieval

In large-scale scenarios, retrieval systems typically contain the modules of initial search with a hashing-based Approximated Nearest Neighbor (ANN) search [79] or fast global similarity computation such as Euclidean distance, and re-ranking. In this case, we can first use global features (*i.e.*, global-pooled 1x1 features before graph reasoning module) to conduct the initial search before applying GRNet to re-rank the resulted short list. We empirically found that this strategy can scale up the application of GRNet without accuracy drop. Moreover, our method can be accelerated by neural network quantization techniques such as binary networks [80] or neural network compression such as pruning [81]. In our application, *i.e.*, fashion retrieval, we can accelerate it by predicting the query category and filtering out gallery clothing with unrelated tags before computing their similarities.

4 FindFashion

We build a new benchmark named FindFashion by revisiting the publicly available datasets. *i.e.*, Street2Shop [2], and DeepFashion [3]. We label 3 attributes (*i.e.*, occlusions, views, and cropping) which mostly affect clothes retrieval performance. According to the attributes of the query, we divide the benchmark into 4 subsets with different difficulty levels. *i.e.*, *Easy*, *Hard-Cropping*, *Hard-Occlusion*, and *Hard-View*.

We adopt the same evaluation measure, *i.e.*, top-k accuracy, to evaluate the performance as in [2], [3].

Data collection and cleaning. We first merged the two existing datasets (*i.e.*, Street2Shop [2], DeepFashion [3]), and formed a large dataset containing 382,230 image pairs and 565,041 images, and then we asked the annotators to screen out the image pairs that are clearly not of the same clothing.

Annotations. Gallery images from Street2Shop have no clothing bounding boxes, we first train a Faster RCNN [14] detector over DeepFashion to detect their bounding boxes, and then manually correct them. We annotate three attributes (*i.e.*, views, occlusions and cropping) for all images. For views, we labeled each clothes images as front, side, or back. Clothing with the yaw angle in $[-45°, 45°]$ is labeled as front, those with yaw angle in $(45°, 135°)$ or $(-135°, -45°)$ is labeled as side while $[135°, 225°]$ as back. For occlusions, clothing with more than 30% occluded by other things such as other clothes, mobile phone or belt is labeled as occluded otherwise as un-occluded. For cropping, clothing with more than 30% cropped is labeled as cropped otherwise as un-cropped.

Images in FindFahsion are of large variance in terms of views, cropping, and occlusions. 8% of images are cropped. 3% of them are occluded. Front view, side view, and back view account for 74%, 20%, and 6% respectively.

Evaluation protocol. As done in [3], we report top-k accuracy to evaluate the retrieval performance. It reflects the quality of the results of a search engine as they would be visually inspected by a user. Four evaluation setups of different difficulty levels are defined according to the query attribute while keeping the gallery unchanged in the test set:

(1) *Easy (E)*, queries are captured from the front view without cropping or occlusion.

(2) *Hard-Cropping (HC)*, queries are with cropping.

(3) *Hard-Occlusion (HO)*, queries are occluded.

(4) *Hard-View (HV)*, queries are of non-frontal view. Namely, side or back view.

We do not split training dataset according to the above four evaluation setups as we found using maximum training data can achieve better results in all the setups. The detailed statistics of our evaluation protocols are listed in Table 3.

5 Experiments

5.1 Implementation Details

Our implementation on customer-to-shop clothing retrieval follows the practice in [3]. We train our models with PyTorch. We perform standard data augmentation with random horizontal flipping. All cropped images are resized to 224×224 before being fed into the networks. Optimization is performed using synchronous SGD with momentum 0.9, and weight decay 0.0005 on servers with 8 GPUs. The initial learning rate is set to 0.01 and decreased by a factor of 10 every 20 epochs. All compared models including ours, are trained using the same training set for 60 epochs. Following previous works [2], [3], [17], the evaluation metric is top-k accuracy.

We set the batch size to 64 during training. Each batch consists of 32 clothing with 2 images per clothes. The query and gallery pairs of the same clothing construct positive training samples while other combinations negative ones.

In the local feature extraction module, we have a total of 7 scales including the global scale (*i.e.*, $L = 7$). The whole spatial window of images is divided into 1×1, 1×2, 2×1, 2×2, 1×3, 3×1 and 3×3 from scales 1 to 7, respectively. In the similarity reasoning module, we use three (*i.e.*, $T = 3$) graph convolution layers with channel number $C^{'}$ set to 128. The projection dimension (*i.e.*, D) is set to 512. The feature extractor is initialized with its pre-trained model on ImageNet while the similarity computation module and the similarity reasoning module are randomly initialized as in [82]. For fair comparison, we use GoogleNet [78] as backbone on the DeepFashion, Street2Shop and FindFashion datasets, and ResNet-50 [13] as backbone on the DeepFashion2 dataset following [17].

As shown in Figure 5, we set $N_l = 5$ when scale l is 1×2, 2×1 and 2×2, and $N_l = 12$ when scale l is 1×3, 3×1 and 3×3. The weights of all convolutional layers in adaptive window pooling are initialized as with [82]. Since the output of vertex prediction unit represents the adaptive spatial windows, we stabilize the training process following [62]. Specifically, the weights in the last fully-connected layer of the vertex prediction unit, are initialized to zeros, and bias is set to the values so that the predicted vertices are equally distributed and identical to their corresponding target vertices in $\mathbf{V}_l^t$.

Methods	Top-1	Top-20	Top-50
FashionNet [3]	7.0	18.8	22.8
Triplet [15]	10.0	37.0	49.9
VAM+Nonshared [15]	11.3	38.8	51.5
VAM+Product [15]	13.4	43.6	56.7
VAM+ImgDrop [15]	13.7	43.9	56.9
DREML(192,48) [60]	18.6	51.0	59.1
KPM [16]	21.3	54.1	65.2
GRNet w/o AWP	25.7	64.4	75.0
GRNet	**26.8**	**65.6**	**75.2**

TABLE 4: Comparison with state-of-the-art methods on DeepFashion consumer-to-shop benchmark [3]. GRNet w/o AWP indicates GRNet without adaptive window pooling.

Method	Tops	Dresses	Skirts	Pants	Outerwear
Kiapour *et al*. [2]	38.1	37.1	54.6	29.2	21.0
VAM+ImgDrop [15]	52.3	62.1	70.9	–	–
Trip. [15]	44.9	56.0	69.0	–	–
Trip.+Partial [15]	47.0	58.3	72.3	–	–
GRNet w/o AWP	58.3	64.2	**72.5**	48.5	38.6
GRNet	**58.6**	**64.5**	72.1	**49.4**	**39.4**

TABLE 5: Comparison with state-of-the-art methods on Street2Shop [2] in terms of top-20 accuracy. GRNet w/o AWP indicates GRNet without adaptive window pooling.

5.2 Comparison with State-of-the-art Methods

Results on DeepFashion. Table 4 compares the proposed GRNet with state-of-the-art methods, including FashionNet [3], triplet-based metric learning approach, and Visual Attention Model (VAM) and its variants (VAM+ImgDrop, VAM+Product, and VAM+Nonshared) [15], on DeepFashion [3]. Except FashionNet, all counterparts use the same backbone GoogleNet [78]. The proposed GRNet outperforms the existing methods with an impressive margin. Specifically, it obtains an accuracy of 26.8, 65.6 and 75.2, and absolutely improves the best ever reported results (VAM+Product) by 13%, 22% and 19% respectively. Notably, VAM [15] uses an attention sub-network which needs clothes segmentation annotations for training, while our GRNet is trained with only query-gallery image pairs, thus it is more practical. We also compare GRNet with recent DREML [60], which achieves state-of-the-art performance on multiple general metric learning benchmarks, including Inshop [3]. We train the DREML model on DeepFashion training set using its open source code with 192 recommended meta classes and 48 ensemble models, as in Table 2 of DREML [60]. Our GRNet is remarkably superior than DREML although DREML employs 48 models for ensembles. The above previous works aim to learn a more discriminative features representation for a single image relying on training with detection and landmark [3], attention mask based on clothing segmentation [15], and network ensemble [60]. Our GRNet outperforms the performance by learning similarities between a query and a gallery using similarity reasoning between local and global similarity at different scales. Moreover, we also compare GRNet with KPM [16], which achieves state-of-the-art performance on multiple person re-identification benchmarks relying on comparing similarity between a query and a gallery using kronechker-product to match the feature maps. Again, our GRNet outperforms KPM remarkably, which demonstrates the effectiveness of our GRNet. Further, it has been shown that adaptive window pooling improves the top-1 accuracy of GRNet by 1.1%.

Results on Street2Shop. We compare the proposed GRNet with state-of-the-art customer-to-shop clothes retrieval methods on Street2Shop dataset [2] in Table 5. It has been shown that it outperform performance to state-of-the-art methods relying on attention mask [15]. It has been shown that it achieves the best results on the tops, dresses, skirts, pants and outerwear categories of Street2Shop. Particularly, it absolutely improves the best ever reported results by 11.6% and 6.2% for tops and dresses categories respectively. Agian, we can observe that adaptive window pooling can improve the retrieval performance of GRNet.

Methods	Top-1	Top-5	Top-10	Top-15	Top-20
class [17]	7.9	19.8	27.3	32.9	36.6
pose [17]	18.2	32.6	41.6	46.9	51.0
class + pose [17]	19.2	34.5	43.5	48.8	52.4
GRNet w/o AWP	25.6	40.1	50.3	55.7	58.5
GRNet	**26.7**	**41.2**	**51.0**	**56.9**	**60.1**

TABLE 6: Comparison with state-of-the-art fashion retrieval methods on the DeepFashion2 [17].

Results on DeepFashion2. We also evaluate the proposed GRNet on the recent challenging DeepFashion2 [17] retrieval benchmark with 292k images and 487k pairs, of which 192k/33k/67k images and 337k/50k/100k pairs are for training, validation and testing, respectively. Previous work [17] improves the performance by training with the supervision of landmarks (*i.e*., "pose") and category labels (*i.e*., "class"). As shown in Table 6, we clearly observe the advantage of the GRNet with or without adaptive window pooling. For fair comparison, our method employs the same feature extractor as [17]. Our proposed GRNet achieves the best results on this challenging dataset with 7.5%, and 7.7% absolute improvement in terms of top-1 and top-20 accuracies respectively.

Results on FindFashion. We evaluate the proposed GRNet on our annotated benchmark FindFashion with four evaluation protocols. Namely, *Easy*, Hard-View, Hard-Cropping, and Hard-Occlusion. We also compare it with DREML [60], KPM [16] and our baseline in Table 7. We utilize the siamese-structure with the classification loss for the global similarity as our baseline. Specifically on FindFashion, our GRNet improves the results of the top-20 accuracy up to 65.9 on *Easy*, 58.5 on Hard-View, 36.2 on Hard-Occlusion and 49.2 on Hard-Cropping. Comparing with the results of KPM [16] which uses the same backbone as ours, GRNet acquires more improvement on the evaluation protocols of Easy, Hard-View, Hard-Occlusion and Hard-Cropping. It demonstrates the proposed method's superiority and capability to take full advantages of different scales information to boost the retrieval performance.

We further enlarge the FindFashion benchmark by merging the recently released dataset, DeepFashion2 [17]. DeepFashion2 annotates the attributes of occlusion, zoom-in, and viewpoint, thus we transform its medium and heavy occlusion labels to our defined occlusion ones, and its medium and large zoom-in labels to our defined cropping ones. We name this kind of extended FindFashion by FindFashion-Ext. It has 125863, 4920, 15883, and 47161 validation pairs, and 41580, 2266, 6021, and 13253 test pairs with the *Easy*, *Hard-Cropping*, *Hard-Occlusion* and *Hard-View* evaluation setup respectively. As shown in Table 8, GRNet with or without adaptive window pooling consistently outperforms its counterparts such as siamese network, DREML, and KPM.

In Table 7 and 8, we have observed that GRNet performs worst on hard-occlusion, and best on hard-view among three non-easy setups. We believe the underlying reasons are as follows: 1) For

Methods	Easy			Hard-View			Hard-Occlusion			Hard-Cropping		
	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50
Baseline	16.9	53.6	67.6	10.4	37.8	53.2	4.5	25.3	35.8	7.3	35.4	49.9
DREML(192,48) [60]	20.7	54.2	68.2	17.2	44.3	54.0	6.3	31.3	43.8	10.6	43.4	55.2
KPM [16]	22.9	56.2	69.2	18.3	45.8	55.8	5.8	25.5	35.4	9.7	34.8	46.7
GRNet w/o AWP	27.1	65.1	**75.2**	23.3	57.9	69.6	7.8	35.0	45.0	14.9	48.4	61.1
GRNet	**28.0**	**65.9**	**75.2**	**23.9**	**58.5**	**70.3**	**8.5**	**36.2**	**45.8**	**15.8**	**49.2**	**61.9**

TABLE 7: Comparison with state-of-the-art methods on FindFashion.

Methods	Easy			Hard-View			Hard-Occlusion			Hard-Cropping		
	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50	Top-1	Top-20	Top-50
Baseline	14.2	46.1	53.0	8.9	34.5	45.6	4.6	18.6	28.2	8.8	28.5	41.5
DREML(192,48) [60]	17.0	45.1	56.5	14.0	39.6	47.8	7.8	24.7	33.7	9.8	32.6	44.6
KPM [16]	19.4	47.3	57.1	14.5	41.4	49.4	7.2	27.0	32.8	10.2	30.6	42.2
GRNet w/o AWP	22.3	55.3	62.3	17.8	47.6	55.5	8.3	28.6	38.1	12.9	39.8	50.6
GRNet	**22.6**	**56.1**	**62.8**	**18.1**	**48.2**	**55.8**	**9.4**	**29.2**	**38.9**	**13.5**	**40.4**	**51.3**

TABLE 8: Comparison with state-of-the-art methods on FindFashion-Ext.

#	Local similarity						Intra-scale connection	Inter-scale connection	Accuracy		
	1×2	2×1	2×2	3×1	1×3	3×3			top-1	top-20	top-50
1	-	-	-	-	-	-	-	-	14.06	47.60	60.62
2	✓	✓	✓	-	-	-	✓	✓	22.60	62.71	73.25
3	-	-	-	✓	✓	✓	✓	✓	23.96	64.48	74.32
4	✓	✓	✓	✓	✓	✓	-	-	24.48	63.85	74.17
5	✓	✓	✓	✓	✓	✓	-	✓	24.79	64.17	74.27
6	✓	✓	✓	✓	✓	✓	✓	-	24.58	63.85	73.44
7	✓	✓	✓	✓	✓	✓	✓	✓	**25.73**	**64.38**	**75.00**

TABLE 9: Ablation experiments of GRNet on DeepFashion [3].

hard-view setup, the queries with side view account for about 77% and those with back view account for about 23% only as described in Section 4. As images with side view and those with front views have much overlap, thus hard-view setup is one task with moderate difficulty, as shown in Figure 7 (a). 2) For hard-cropping setup, most of cropped regions are marginal parts of clothes with little discriminative information, and discriminative regions might be kept, as shown in Figure 7 (b). 3) For hard-occlusion setup, most of discriminative regions are occluded, which results in great difficulty for fashion retrieval, as shown in Figure 7 (c).

Fig. 7: Some examples of query images in the setting of hard-view (a), hard-cropping (b) and hard-occlusion (c), respectively.

5.3 Ablation Study

We investigate the effectiveness of each component in the proposed GRNet by conducting the following ablation studies on the DeepFashion dataset [3], shown in Table 9. For simplicity, we perform ablation experiments with GRNet without adaptive window sampling except otherwise specified.

Graph reasoning. To validate the effectiveness of graph reasoning, we utilize a GRNet without graph reasoning as our baseline(#1), which computes the global similarity between global features. Comparing #1 and #7, our graph reasoning acquires 11.6% improvement on the top-1 accuracy.

Inter-scale connections. Comparing #6 and #7, it can be observed that the proposed GRNet can achieve 1.15% performance gain on the top-1 accuracy by adding the inter-scale connections (Noted that #6 and #4 retain the connections between the global similarity and the local similarities but remove the connections between different scales. In other words, #6 and #4 just add the connections between the global similarity and the local similarities on the baseline #1).

Intra-scale connections. As reported in Table 9, by propagating similarities at the same scale, our intra-scale connections acquire 0.9% improvement on the top-1 accuracy (#5 vs #7). It shows that the local similarities are also refined by their interactions at the same scale.

Multi-scale similarities. Comparing #1, #2, #3 and # 7, we observe that the performance is consistently improved when using more scale similarities. Specifically, the accuracy is improved from 14%, 47% and 60% to 22%, 62% and 73% at top-1, top-20, and top-50 after adding 2×1, 1×2, and 2×2. They are improved slightly by further adding 1×3, 3×1 and 3×3 similarities. Moreover, we compare the results of different scale levels of local similarity. Comparing #2 and #3, the fine scale brings very subtle improvement. The result shows that the multi-scale similarities can enhance the global similarity representation.

Number of graph convolution layers. We conduct experiments with different numbers of graph convolutional layers. The top-1 accuracy increases from 16.8%, 22.8%, to 25.7% when the number of graph convolutional layer is set to 1, 2, and 3. We observe a performance drop if the layer number is increased

Training set	# ins	Easy		Hard-View		Hard-Occlusion		Hard-Cropping	
		top-1	top-20	top-1	top-20	top-1	top-20	top-1	top-20
All	**109810**	**27.1**	**65.1**	**23.3**	**57.9**	**7.8**	**35.0**	**14.9**	**48.4**
Easy	70715	26.4	65.0	20.4	54.2	6.3	34.3	10.0	38.3
Non-front	26825	18.6	54.4	14.7	48.0	5.0	30.2	9.7	38.6
Occlusion	3991	6.4	28.6	4.2	23.3	2.0	11.7	2.3	17.9
Cropping	8279	11.0	41.2	7.6	33.8	2.3	19.4	6.0	32.5

TABLE 10: Comparison between GRNets trained on different training subsets of FindFashion in terms of top-1 and top-20 accuracy. # ins indicates the number of training query instances.

Methods	Inference time (s/query img)			Accuracy		
	Features extraction	Similarity computation	Total	Top-1	Top-20	Top-50
Euclidean distance	0.0070	0.9226	0.9296	12.5	43.5	50.4
DREML (192,48) [60]	0.3360	0.9346	1.2706	15.0	42.4	54.2
KPM [16] + top-500	0.0070	1.0051	1.0121	17.1	45.0	55.5
GRNet w/o AWP + top-500	0.0070	1.0317	1.0387	20.4	53.8	59.5
GRNet + top-500	0.0070	1.0753	1.0823	**20.6**	**54.2**	**59.6**

TABLE 11: Comparison between Our GRNet and triplet-based method in terms of running time and accuracy on FindFashion-Ext.

Projection dim. D	Channel num. $C^{'}$	Accuracy		
		Top-1	Top-20	Top-50
512	128	25.73	64.38	75.00
512	256	25.52	64.50	74.43
512	512	25.92	64.75	75.54
256	128	24.06	63.02	73.33
256	256	25.10	64.48	74.17
128	128	24.69	63.64	74.38

TABLE 12: Impacts of Dimensions of GRNet w/o AWP on the DeepFashion [3].

Methods	FLOPs	Parameters
Euclidean distance	1.586G	5.974M
Basesline	1.585G	5.978M
DREML(192,48) [60]	76.074G	291.177M
KPM [16]	1.595G	5.979M
GRNet w/o AWP	1.763G	6.696M
GRNet	1.776G	7.099M

TABLE 13: The computational cost of our GRNet and other state-of-the-art methods. FLOPs indicates the number of floating point operations, which contains the operations of the features extractor and the similarity computation.

further due to over-fitting. Thus, we fix the number of graph convolutional layers to 3 , which achieves the best performance.

Projection dimension and channel number in graph CNN. Table 12 evaluates GRNet with different projection dimensions D and channel numbers $C^{'}$. It has been observed that GRNet is insensitive to projection dimension and channel number. Except $D = 128$, there is no obvious performance drop. We fix $D = 512$ and channel number $C^{'}$ to 128 in all our experiments except otherwise noted.

Adaptive window pooling. Table 4, 5 6, and 7 have been shown that adaptive window pooling can consistently improve the performance of fashion retrieval w.r.t. fixed windows. These results demonstrate the effectiveness of adaptive window pooling. As visualized in Figure 10, we believe that the improvement is achieved via the alignment between local features of the query with those of the gallery and comparing the similar components at different scales.

Training data of FindFashion. In our proposed benchmark FindFashion, we do not split the training data according the four evaluation setups. Table 10 shows the performance of GRNets when trained on the training subsets with easy, non-front views, occlusions, or cropping only. As shown in the table, the performance drops greatly compared with the model trained on all the training set. We believe that the reason lies in that the subsets with easy, non-front views, occlusions, or cropping are with inadequate training samples. Comparing the performance trained on only easy subset, the model trained on all the training set obtains the comparable performance on easy subset but better performance on other subsets, *e.g.*, non-front views, occlusions and cropping. It demonstrates that training with hard samples can improve the generalization ability of model when inference with hard samples.

5.4 Results on Large Scale Fashion Retrieval

To evaluate the time and accuracy of the proposed GRNet in large-scale scenarios, we conduct one initial search with the global average-pooled feature-based Euclidean distance, which returns the top-500 clothes, and then re-rank the initial short list via GRNet on the largest benchmark, FindFashion-Ext. To better simulate the large-scale scenarios, we use the whole FindFashion-Ext in this experiment. All experiments of large-scale scenario run on a server with 128G RAM using one GTX1080 Ti. Table 11 compares the inference time of GRNet with the initial search, triplet-based method, and other state-of-the-art methods, such as KPM [16] and DREML [60]. To fair comparison with KPM [16], which also computes similarity between a query and a gallery, we report the time of KPM with the initial search. It has been shown that GRNet w/o AWP greatly outperforms its competitor triplet-based method in terms of accuracy at the cost of 0.11s and about 12% runtime. It has also shown that our lightweight adaptive window pooling can further improve the accuracy of GRNet at the cost of 0.04s and about 4% runtime overhead, which could be negligible. Thus, our proposed GRNet with initial search can achieve a better performance with an acceptable cost of time on large-scale scenarios.

Besides the inference time, we also report the number of parameters and computational complexity in terms of FLOPs, as shown in Table 13. Except for the DREML [60] ensembling 48 models, the parameters and FLOPs of other methods are comparable. Compared with Euclidean distance and baseline, our graph reasoning network and adaptive window pooling cost more 0.722M parameters with 0.177G FLOPs and 0.403M parameters with 0.013G FLOPs respectively. Though a different graph is built for each pair of images, the additional parameters and FLOPs

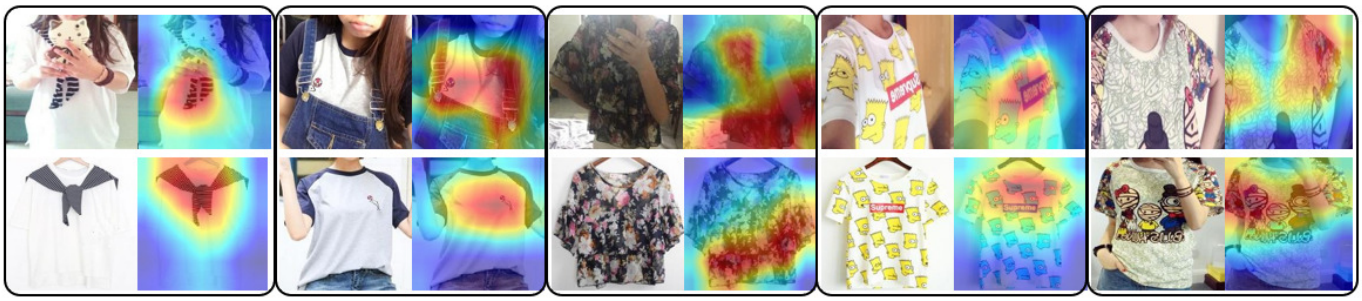


Fig. 8: Visualization of important regions in the query and the gallery images. Each 2×2 images in one rectangle show one query-gallery image pair and their corresponding highlights, in which the top-left, the top-right, the bottom-left, and the bottom-right are the query, the query highlights, the gallery, and the gallery highlights respectively. Query 1 and 3 are occluded by hands; query 2 is occluded by trousers; query 4 is side view while its gallery front; query 5 is cropped.

Fig. 9: Examples of the up-weighted nodes in our similarity pyramid graph. Each node represents one similarity of the local patch (indicated by red rectangles) pair from the query (the top row) and the gallery (the bottom row). Each 2×2 images in one black rectangle show one query-gallery image pair and their up-weighted local patch pairs, where the left column shows the most important node before the similarity reasoning and the right shows it after the similarity propagation. GRNet can up-weight the similarity between aligned salient clothing components (*e.g.*, logo) after graph reasoning.

could be negligible comparing with those of features extractors, such as GoogleNet (5.974M parameters and 1.586G FLOPs) and ResNet-50 (25.56M parameters and 4.14G FLOPs).

5.5 Visualization

To investigate why the GRNet works effectively, we firstly analysis the GRNet without adaptive window pooling to investigate the effectiveness of graph reasoning. We employ Grad-CAM [83] to visualize the important regions in the query and the gallery images for predicting whether they belong to the same clothing in Figure 8. It has been shown that GRNet automatically focuses on local discriminative regions (*e.g.*, scarf, and logo) and shared regions which can be observed in both the query and the gallery while ignoring non-discriminative regions (*e.g.*, non-texture regions), occlusions (*e.g.*, hand) or unique regions which can be observed only on one side due to different views or cropping. We visualize the similarity node which contributes most to the final classification by selecting the one whose edge outgoing to the global similarity node has the largest weight, as shown in Figure 9. It has been shown that our GRNet can focus on aligned salient clothing components (*e.g.*, logo).

To investigate why the adaptive window pooling improves performance, we visualize the rectified query in Figure 10. Each row is a query-gallery image pair with predicted vertex for query image at different scales, where except for the last row is a negative pair, the other rows are positive pairs. It has been shown that the adaptive window pooling can locate the representative regions or discriminative components by predicting vertices according to the gallery at different scales. Even without direct supervision of target windows, the vertex prediction unit can learn to place the vertex around the regions which are similarity to galleries or discriminative at same scale. Thus the adaptive windows in the query are aligned to the same scale windows in the gallery. Moreover, in negative pairs, the adaptive window pooling still learns to find the similarity components based on gallery, or nearly predict the default windows to distinguish the pair whether the same or not.

We present qualitative retrieval results on the DeepFashion dataset [3] and the Street2Shop dataset [2] in Figure 12 and Figure 13, respectively. Benefiting from the Graph Reasoning on the similarity pyramid, the global similarity is effectively refined by multi-scale local similarities. Furthermore, we also show some representative negative results in Figure 11. We have observed that our method might fail when the query can be differentiate from negative galleries with very fine-grained features only (the first row) or has no discriminative features (the second, the third, and the bottom rows). Particularly, for the first row, the query is a T-shirt with text logo. Our method only retrieves clothes with similar logo, but can not recognize the text to retrieve the clothing with the same text logo. Comparing with the second row in the Figure 12, our method are strong in retrieving clothing with similar local regions, but difficult to distinguish the similar regions with different texts.

6 Conclusions

In this paper, we focus on a real-world application task of customer-to-shop clothing retrieval and have proposed a Graph Reasoning Network (GRNet). The proposed GRNet first represents the multi-scale regional similarities and their relationships as a graph and then performs graph CNN based reasoning over the graph to adaptively adjust both the local and global similarities. Further, we propose the adaptive window pooling module to align

Fig. 10: Visualization of the predicted adaptive pooling windows of query images at different scales. Each row is a pair of query-gallery images. The first four rows are positive pairs while the last row negative pair. In each row, the first column shows the gallery image while other columns the query images with predicted adaptive pooling windows at different scales superposing on them. The blue round points are predicted by vertex prediction unit while the red square points are fixed for all scales (best viewed in color).

queries with galleries. GRNet achieves more precise matching of salient clothing components through information propagation among nodes of similarities. To facilitate future research, we have also introduced a new benchmark called FindFashion, which contains rich annotations of clothing including bounding boxes, views, occlusions, and cropping. We also provide a deeper analysis on the experimental results to explain why hard-occlusion performs worst on the non-easy setting. Moreover, we extend FindFashion with the recent released dataset DeepFashion2 to conduct large scale experiments. To facilitate the application of real scenarios, which has large-scale images and requires efficient search, we propose a large-scale fashion retrieval system to accelerate the search process. Extensive experimental results show that our proposed method obtains new state-of-the-art results on both the existing datasets and FindFashion.

ACKNOWLEDGMENTS

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2020B1515020048, Beijing Municipal Science and Technology Commission (Grant No. Z181100008918004), and National Natural Science Foundation of China (Grant No.61702565, No.61976250 and No.U1811463).

REFERENCES

[1] J. Huang, R. Feris, Q. Chen, and S. Yan, "Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network," in *ICCV*, 2015, pp. 1062–1070.

[2] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to Buy It: Matching Street Clothing Photos in Online Shops," in *ICCV*, 2015, pp. 3343–3351.

[3] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," in *CVPR*, 2016, pp. 1096–1104.

[4] X. Ji, W. Wang, M. Zhang, and Y. Yang, "Cross-Domain Image Retrieval with Attention Modeling," in *ACM MM*, 2017, pp. 1654–1662.

[5] Y. Song, Y. Li, B. Wu, C. Y. Chen, X. Zhang, and H. Adam, "Learning Unified Embedding for Apparel Recognition," in *ICCVW*, 2017, pp. 2243–2246.

[6] C. Corbière, H. Ben-Younes, A. Ramé, and C. Ollion, "Leveraging Weakly Annotated Data for Fashion Image Retrieval and Label Prediction," in *ICCV Workshop*, 2017.

[7] N. Garcia and G. Vogiatzis, "Dress Like a Star: Retrieving Fashion Products from Videos," in *ICCVW*, 2017, pp. 2293–2299.

[8] Z. Q. Cheng, X. Wu, Y. Liu, and X. S. Hua, "Video2Shop: Exact Matching Clothes in Videos to Online Shopping Images," in *CVPR*, 2017, pp. 4169–4177.

[9] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin, "Visual Search at Alibaba," in *ACM SIGKDD*, 2018, pp. 993–1001.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012, pp. 1097–1105.

[11] K. He and R. Girshick, "Mask R-CNN," in *arXiv preprint arXiv:1703.06870*, 2017.

[12] R. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440–1448.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016, pp. 770–778.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards Real-time Object Detection with Region Proposal Networks," in *NIPS*, 2015, pp. 91–99.

[15] Z. Wang, Y. Gu, Y. Zhang, J. Zhou, and X. Gu, "Clothing Retrieval with Visual Attention Model," in *IEEE Visual Communications and Image Processing*, 2017.

[16] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "End-to-end deep kronecker-product matching for person re-identification," in *CVPR*, June 2018.

[17] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] Z. Kuang, Y. Gao, G. Li, P. Luo, Y. Chen, L. Lin, and W. Zhang, "Fashion retrieval via graph reasoning networks on a similarity pyramid," in *ICCV*, 2019.

[19] X. Wang and T. Zhang, "Clothes Search in Consumer Photos via Color Matching and Attribute Learning," in *ACM MM*, 2011.

[20] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan, "Style Finder: Fine-Grained Clothing Style Detection and Retrieval," in *CVPRW*, 2013.

[21] J. Fu, J. Wang, Z. Li, M. Xu, and H. Lu, "Efficient Clothing Retrieval with Semantic-preserving Visual Phrases," in *ACCV*, 2012, pp. 420–431.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988.

[23] Y. H. Kuo and W. H. Hsu, "Feature Learning with Rank-Based Candidate Selection for Product Search," in *ICCVW*, 2017, pp. 298–307.

[24] X. Guo, H. Wu, Y. Cheng, S. Rennie, and R. S. Feris, "Dialog-based Interactive Image Retrieval," in *NIPS*, 2018, pp. 1–15.

[25] 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," in *ICCV*, 2017, pp. 1472–1480.

[26] B. Zhao, J. Feng, X. Wu, and S. Yan, "Memory-augmented Attribute Manipulation Networks for Interactive Fashion Search," *CVPR*, pp. 6156–6164, 2017.

[27] R. Zakizadeh, M. Sasdelli, Y. Qian, and E. Vazquez, "Improving the Annotation of DeepFashion Images for Fine-grained Attribute Recognition," in *arXiv preprint*, 2018.

[28] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel Classification with Style," in *ACCV*, 2012, pp. 321–335.

[29] H. Chen, A. Gallagher, and B. Girod, "Describing Clothing by Semantic Attributes," in *ECCV*, 2012.

[30] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations," in *ACM MM*, 2018, pp. 22–26.

[31] N. Inoue, E. Simo-Serra, T. Yamasaki, and H. Ishikawa, "Multi-label Fashion Image Classification with Minimal Human Supervision," in *ICCVW*, 2017, pp. 2261–2267.

[32] W. H. Lin, K.-T. Chen, H. Y. Chiang, and W. Hsu, "Netizen-Style Commenting on Fashion Photos: Dataset and Diversity Measures," in *arXiv preprint*, 2018.

[33] "Fashionai Dataset. http://fashionai.alibaba.com/datasets." [Online]. Available: http://fashionai.alibaba.com/datasets

[34] D. Duvenaud, D. Maclaurin, J. Aguilera-iparraguirre, G. Rafael, T. Hirzel, and R. P. Adams, "Convolutional Networks on Graphs for Learning Molecular Fingerprints," in *NIPS*, 2015.

[35] T. N. Kipf and M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," in *ICLR*, 2017, pp. 1–14.

[36] M. Schlichtkrull, T. N. Kipf, P. Bloem, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," in *European Semantic Web Conference*, 2018.

[37] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated Graph Sequence Neural Networks," in *ICLR*, 2016.

[38] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan, and E. P. Xing, "Interpretable Structure-Evolving LSTM," in *CVPR*, 2017.

[39] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic Object Parsing with Graph LSTM," in *ECCV*, 2016.

[40] Z. Wang, T. Chen, R. Xu, and L. Lin, "Multi-label Image Recognition by Recurrently Discovering Attentional Regions," in *ICCV*, 2017.

[41] S. Yang, G. Li, and Y. Yu, "Graph-structured referring expression reasoning in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] S. Yang, G. Li, and Y. Yu, "Dynamic graph attention for referring expression comprehension," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4644–4653.

[43] S. Yang, G. Li, and Y. Yu, "Relationship-embedded representation learning for grounding referring expressions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[44] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin, "Deep Reasoning with Knowledge Graph for Social Relationship Understanding," in *IJCAI*, 2018.

[45] G. Li, X. Zhu, Y. Zeng, Q. Wang, and L. Lin, "Semantic relationships guided representation learning for facial action unit recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8594–8601.

[46] X. Wang and A. Gupta, "Videos as Space-Time Region Graphs," in *ECCV*, 2018.

[47] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-Scale Image Retrieval with Attentive Deep Local Features," in *ICCV*, 2017.

[48] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end Learning of Deep Visual Representations for Image Retrieval," *IJCV*, vol. 124, no. 2, pp. 237–254, 2017.

[49] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for Weakly Supervised Place Recognition," in *CVPR*, 2016.

[50] F. Radenović, G. Tolias, and O. Chum, "CNN Image Retrieval Learns from BoW: Unsupervised Fine-tuning with Hard Examples," in *ECCV*, 2016, pp. 3–20.

[51] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep Image Retrieval: Learning Global Representations for Image Search," in *ECCV*, 2016, pp. 241–257.

[52] G. Tolias, R. Sicre, and H. Jégou, "Particular Object Retrieval with Integral Max-pooling of CNN Activations," in *ICLR*, 2016.

[53] A. B. Yandex and V. Lempitsky, "Aggregating Local Deep Features for Image Retrieval," in *ICCV*, 2015.

[54] Z. Chen, Z. Kuang, K.-Y. K. Wong, and W. Zhang, "Aggregated Deep Feature from Activation Clusters for Particular Object Retrieval," in *ACM MM Thematic Workshops*, 2019.

[55] Z. Chen, Z. Kuang, W. Zhang, and K.-Y. K. Wong, "Learning Local Similarity with Spatial Relations for Object Retrieval," in *ACM MM*, 2019.

[56] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "BIER - Boosting Independent Embeddings Robustly," in *ICCV*, vol. 2017, 2017, pp. 5199–5208.

[57] Y. Yuan, K. Yang, and C. Zhang, "Hard-Aware Deeply Cascaded Embedding," in *ICCV*, 2017, pp. 814–823.

[58] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon, "Attention-based Ensemble for Deep Metric Learning," in *CVPR*, 2018.

[59] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "BIER: Boosting Independent Embeddings Robustly," in *ICCV*, 2017.

[60] H. Xuan, R. Souvenir, and R. Pless, "Deep Randomized Ensembles for Metric Learning," in *ECCV*, 2018, pp. 1–12.

[61] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6148–6157.

[62] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[63] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 464–472.

[64] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2285–2294.

[65] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[66] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European conference on computer vision*. Springer, 2014, pp. 392–407.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[68] Y. Xu, J. Duan, Z. Kuang, X. Yue, H. Sun, Y. Guan, and W. Zhang, "Geometry normalization networks for accurate scene text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[69] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
[70] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.
[71] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 5, pp. 1213–1226, 2018.
[72] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Advances in neural information processing systems*, 2004, pp. 169–176.
[73] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR 2011*. IEEE, 2011, pp. 777–784.
[74] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3013–3020.
[75] C. Wallraven and B. Caputo, "Recognition with Local Features: the Kernel Recipe," in *ICCV*, 2003.
[76] S. Boughorbel and J.-p. Tarel, "Non-Mercer Kernels for SVM Object Recognition," in *BMVC*, 2014.
[77] F. L. Bookstein, "Principal warps: Thin-Plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, p. 567–585, 1989.
[78] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," in *CVPR*, 2015.
[79] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.
[80] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," 2016.
[81] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *ICLR*, 2016.
[82] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification," in *ICCV*, 2015.
[83] R. R. Selvaraju, M. Cogswell, A. Das, D. Vedantam, Ramakrishna Parikh, and D. Batra, "Visual Explanations from Deep Networks via Gradient-based Localization," in *ICCV*, 2017.

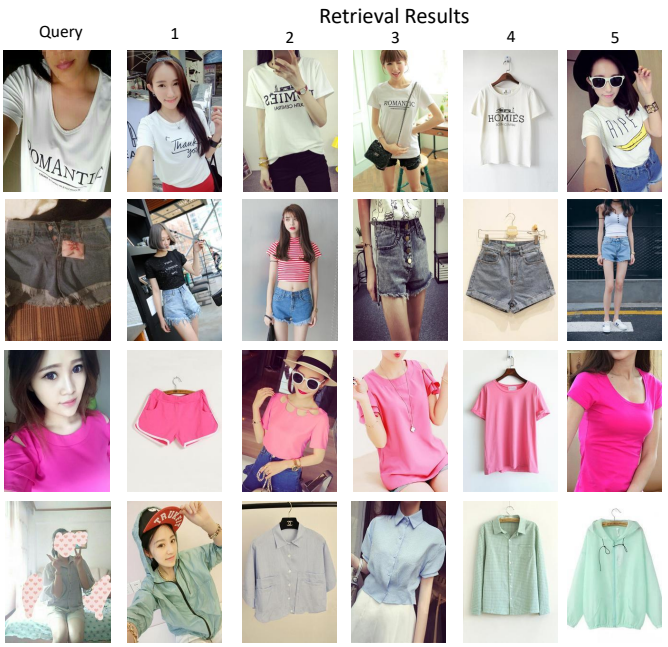


Fig. 11: Representative negative results on the Deepfashion [3]. We show the top 5 most similar images to the query image.

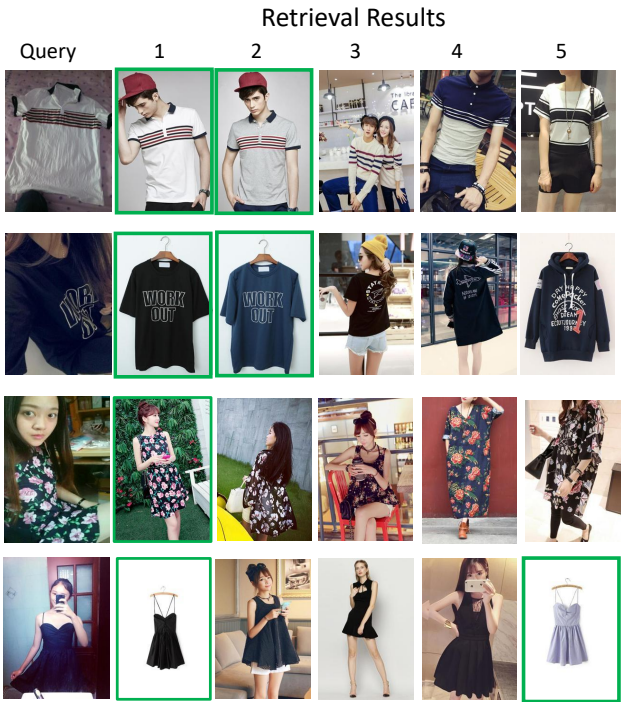


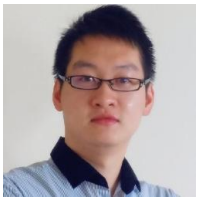
Fig. 12: Qualitative results on the Deepfashion [3]. We show the top 5 most similar images to the query image. Correct results are highlighted in green.

Fig. 13: Qualitative results on the Street2Shop [2]. We show the top 5 most similar images to the query image. Correct results are highlighted in green.

Yiming Gao received his B.S. degree in the School of Mathematics from South China University, China. He is currently pursuing his Master's Degree at the School of Data and Computer Science at Sun Yat-Sen University. His current research interests include computer vision (*e.g.*, image retrieval and human-centric tasks) and machine learning.

Zhanghui Kuang received his B.S. degree from Sun Yat-Sen University, Guangzhou, China, in 2009, and Ph.D. degree from The University of Hong Kong in 2014. He is currently a Research Director in SenseTime Group Limited. His research interests include deep learning and computer vision. He has authorized and co-authorized on more than 10 papers in top-tier academic journals and conferences.

Guanbin Li (M'15) is currently an associate professor in School of Data and Computer Science, Sun Yat-sen University. He received his PhD degree from the University of Hong Kong in 2016. His current research interests include computer vision, image processing, and deep learning. He is a recipient of ICCV 2019 Best Paper Nomination Award. He has authorized and co-authorized on more than 40 papers in top-tier academic journals and conferences. He serves as an area chair for the conference of VISAPP. He has been serving as a reviewer for numerous academic journals and conferences such as TPAMI, TIP, TMM, TC, CVPR, AAAI and IJCAI.



Ping Luo is an Assistant Professor in the department of computer science, The University of Hong Kong (HKU). He received his PhD degree in 2014 from Information Engineering, the Chinese University of Hong Kong (CUHK), supervised by Prof. Xiaoou Tang and Prof. Xiaogang Wang. He was a Postdoctoral Fellow in CUHK from 2014 to 2016. He joined SenseTime Research as a Principal Research Scientist from 2017 to 2018. His research interests are machine learning and computer vision. He has published 70+ peer-reviewed articles in top-tier conferences and journals such as TPAMI, IJCV, ICML, ICLR, CVPR, and NIPS. His work has high impact with 7,000 citations according to Google Scholar. He has won a number of competitions and awards such as the first runner up in 2014 ImageNet ILSVRC Challenge, the first place in 2017 DAVIS Challenge on Video Object Segmentation, Gold medal in 2017 Youtube 8M Video Classification Challenge, the first place in 2018 Drivable Area Segmentation Challenge for Autonomous Driving, 2011 HK PhD Fellow Award, and 2013 Microsoft Research Fellow Award (ten PhDs in Asia)

Yimin Chen is currently a senior researcher in SenseTime. He received his bachelor and master degrees in biomedical engineering from Huazhong University of Science and Technology in 2009 and 2012 respectively, his PhD in electronic engineering from City University of Hong Kong in 2016. His research focuses on computer vision and pattern recognition.

Liang Lin (M'09, SM'15) is a Full Professor at Sun Yat-sen University, and CEO of DarkerMatter AI. He worked as the Executive Director of the SenseTime Group from 2016 to 2018, leading the R&D teams in developing cutting-edge, deliverable solutions in computer vision, data analysis and mining, and intelligent robotic systems. He has authored or co-authored more than 200 papers in leading academic journals and conferences. He is an associate editor of IEEE Trans. Human-Machine Systems and IET Computer Vision, and he served as the area/session chair for numerous conferences, such as CVPR, ICME, ICCV. He was the recipient of Annual Best Paper Award by Pattern Recognition (Elsevier) in 2018, Dimond Award for best paper in IEEE ICME in 2017, ACM NPAR Best Paper Runners-Up Award in 2010, Google Faculty Award in 2012, award for the best student paper in IEEE ICME in 2014, and Hong Kong Scholars Award in 2014. He is a Fellow of IET.

Wayne Zhang received the B.Eng. degree in electronic engineering from the Tsinghua University, Beijing, China, in 2007, the M.Phil. degree in 2009, and Ph.D. degree in 2012, both in information engineering from The Chinese University of Hong Kong. He is currently a Senior Research Director in SenseTime Group Limited. He serves as an EXCO member of AI Specialist Group of Hong Kong Computer Society. His research interests include deep learning and computer vision.