Feature Learning with Rank-Based Candidate Selection for Product Search

Yin-Hsi Kuo and Winston H. Hsu
National Taiwan University, Taipei, Taiwan

Abstract

*Nowadays, more and more people buy products via e-commerce websites. We can not only compare prices from different online retailers but also obtain useful review comments from other customers. Especially, people tend to search for visually similar products when they are looking for possible candidates. The need for product search is emerging. To tackle the problem, recent works integrate different additional information (*e.g.*, attributes, image pairs, category) with deep convolutional neural networks (CNNs) for solving cross-domain image retrieval and product search. Based on the state-of-the-art approaches, we propose a rank-based candidate selection for feature learning. Given a query image, we attempt to push hard negative (irrelevant) images away from queries and make ambiguous positive (relevant) images close to queries. We investigate the effects of global and attention-based local features on the proposed method, and achieve 15.8% relative gain for product search.*

1. Introduction

Due to the rapid growth of e-commerce websites (*e.g.*, Amazon, eBay, Alibaba, Rakuten), it influences customers' shopping behavior. The statistics shows that more than 40% Internet users (1 billion+) have online shopping experience.[1] For the biggest online shopping event in the world (November 11th, Singles Day), the total gross merchandise volume (GMV) on Alibaba e-commerce is around 14.3 and 17.8 billion (USD) for just 24 hours in 2015 and 2016, respectively.[2] In 2016, it achieves 1 billion in the first 5 minutes. These facts demonstrate online shopping becomes a part of our life. It is not only convenient but also comparable for customers when they surf the Internet. Customers can search for similar products or review comments before purchases. It is essential to provide relevant product information and recommendation; hence, plenty of department stores and retailers (*e.g.*, Macy's and Target) incorporate with visual search and recognition companies/technologies

Figure 1. We attempt to leverage e-commence data for achieving better product search results. We propose a rank-based candidate selection for deciding hard negative (irrelevant) images and ambiguous positive (relevant) images in the offline learning process. Hence, we can learn better features and may be able to ignore cluttered background for image matching.

(*e.g.*, Cortexica, Slyce, and ViSenze) for better online shopping experience.

With the shift of online shopping, it also motivates social media (*e.g.*, Twitter, Facebook) to integrate buyable buttons for providing seamless shopping experience (*e.g.*, 'see now, buy now' in fashion show). As reported by Pinterest,[3] they have partnered with 20,000 merchants (over 10 million unique products) and provided visual search [17] since 2015. In 2016, Instagram also announces they will provide seamless mobile shopping experience in their application.[4] eBay and Alibaba further launch virtual reality shops for instant purchases. Therefore, an efficient and effective visual (product) search engine becomes one of the most critical and emerging trends in e-commerce websites and mobile applications (*e.g.*, recommendation [48], visual discovery [47], online video advertising [7], Amazon's Firefly, Alibaba's Pailitao[5], and Google Lens).

Users and retailers can manually annotate regions of interest (ROIs) or bounding boxes for products, and associate them with online shopping stores or e-commerce websites.

[1] Statistics and market data about e-commerce (Statista)
[2] Live updates: Alibaba's 11.11 global shopping festival (Alibaba)
[3] New ways to shop with Pinterest (Pinterest)
[4] Shopping coming to Instagram (Instagram)
[5] Pailitao (Taobao, Alibaba)

It would be more smart and effortless processes to rely on robust product search systems, and could bring more potential revenues for e-commerce. The fundamental of product search [13, 17] is related to content-based image retrieval (CBIR) and mobile visual search (MVS) [6, 10]. It is a more challenging problem for product search. In real applications, consumer photos are quite different from online shopping stores (cross-domain image retrieval) as examples shown in Figure 2. These photos usually contain cluttered background with various lighting conditions. To tackle this problem, the state-of-the-art approaches propose to utilize deep convolutional neural networks (CNNs) [22] for better feature representations. CNN features have been demonstrated promising results in image classification and retrieval [1, 31]. To deal with object queries or small targets, recent works further utilize bounding box or landmark information in the learning process [14, 21, 24], and extract features from the learned region proposals. However, for large-scale datasets, it is time-consuming and infeasible to manually annotate bounding boxes for training.

Instead of utilizing manual annotation, we attempt to leverage freely available product (item) information from e-commerce data in our learning process. We propose to utilize the ranking information from relevant (positive) and irrelevant (negative) products for learning better feature representations (*e.g.*, ignoring cluttered background). We investigate the effects of rank-based candidate selection on global and attention-based local features. Experiment results show that we can achieve better retrieval accuracy as shown in the bottom row of Figure 1. The primary contributions of this paper include,

- Proposing rank-based candidate selection (hard negatives and ambiguous positives) in an end-to-end feature learning framework (Section 3).
- Investigating the effects of global and attention-based local features on the proposed method, and demonstrating the improvement of retrieval accuracy (Section 5).

2. Related works

To achieve effective and efficient image search results, the state-of-the-art approaches usually extract bag-of-words (BoW) model [38], vector of locally aggregated descriptors (VLAD) [16], or binary representations [39]. Nowadays, deep learning becomes promising and robust representations for image classification [22]. Instead of designing hand-crafted features, deep convolutional neural networks (CNNs) learn weights and intermediate features directly from large-scale datasets [4]. As demonstrated in [32], we can directly utilize a pre-trained model on ImageNet (*e.g.*, AlexNet [22]) to achieve the state-of-the-art performance on

Figure 2. Sampled examples for (a) query and (b) database images from AlibabaS (Section 4.3). Images in the same column represent the same product (target item). For e-commerce data, images usually contain cluttered background.

image retrieval tasks. The learning process of deep learning can also learn mid- and high-level representations in different layers [46].

For off-the-shelf networks [32] with fixed input size, we usually extract the last convolutional layers (*e.g.*, Conv5 on AlexNet [22], or Conv5_3 on VGGNet [37]) or fully-connected (FC) layers (*e.g.*, FC6 or FC7 on AlexNet). When adopting to another datasets, we extract features from mid-level layers (*e.g.*, Conv5 or FC6) [2, 45]. Babenko *et al.* [4] propose to collect a large amount of relevant data (*i.e.*, a landmark dataset for building retrieval) for fine-tuning, and investigate the effects of CNN features on different layers.

Because the image resolution is important for image retrieval, Razavian *et al.* [33] observe that we can have a huge performance gain while using convolutional (Conv) features on the original image resolution. It not only reduces the model size (*i.e.*, without fully-connected layers) but also releases the constraint of fixed image size. However, the dimensions of output features will be different for Conv features. It is hard to calculate distance/similarity under different dimensions. Recent works propose to pool or aggregate Conv features to fixed feature dimensions such as max-pooling [40], sum-pooling [3], spatial pyramid pooling (SPP) [19], BoW-like [26], and VLAD-like [27] methods.

To deal with object-level image retrieval, we extract features from possible regions such as different sizes of grids (*e.g.*, 1x1, 2x2, 3x3 grids) [33] or sliding window approaches [11]. Motivated by the success of object detection, Ren *et al.* propose faster R-CNN (region-based CNN) with region proposal networks (RPNs) [34]. RPN learning has shown promising results for image retrieval tasks [12, 35]; however, we observe that the learning process requires bounding box information. For real applications and datasets, it might be hard to obtain these information or corresponding features from spatial verification on non-rigid objects. Noh *et al.* [28] propose to utilize attention-based local features for object queries. For e-commerce data, we only have a small amount of annotated data, and it is not suitable to fine-tune on categories (items). Rather than directly fine-tuning the network with standard cross-entropy

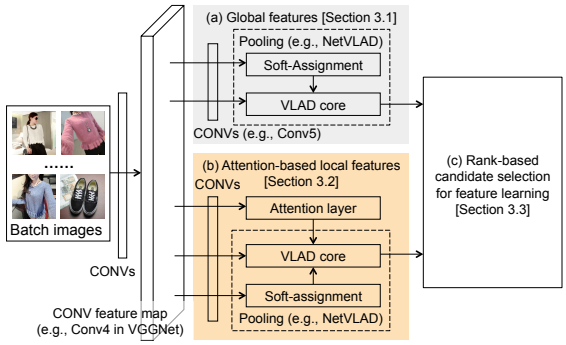


Figure 3. The network structure of the proposed feature learning with rank-based candidate selection. (a) The state-of-the-art methods usually adopt a pooling layer after the last Conv layer (*e.g.*, Conv5 + NetVLAD [1]), and we view the extracted features as global features. (b) We further add an attention layer for extracting local features. (c) Finally, we calculate the loss from those selected ambiguous positive and hard negative candidates.

loss (*e.g.*, landmark recognition), we further consider a loss function based on the ranking information.

There exist lots of challenges for product search such as cross-domain image retrieval, noisy tags, cluttered background [14]. To tackle these problems, recent works propose end-to-end learning by utilizing different information in the learning process [1, 30, 31, 42, 44]. For fashion retrieval, we can utilize additional category and attribute information for multi-task learning [21, 24]. For cross-domain learning, we can utilize siamese [23], triplet [24], or rank-based networks [1] to learn better feature representations. It is also essential to mine hard examples in the learning process [31]. In this work, we propose a rank-based candidate selection for feature learning. Motivated by NetVLAD [1], we further consider the loss function for both positive and negative images, and conduct experiments on both global and attention-based local features.

3. Rank-based feature learning

For achieving better (object-level) product search, we might need to estimate possible target objects and then extract features on these regions. We observe that images in e-commerce websites usually do not contain bounding boxes or regions of interest (ROIs). For e-commerce data, images are very diverse because retailers can upload any kinds of images. Especially, in consumer to consumer (C2C) markets, sellers would like to add promotion information or apply visually appealing features for their product images (*e.g.*, wording of discount). As shown in Figure 2, both query and database images contain not only products but also cluttered background. If we can learn more about product features and contextual information, we can have better retrieval accuracy for product search.

Based on the promising results in deep learning, we attempt to utilize robust CNN features and propose a rank-based candidate selection for feature learning. The proposed network structure is shown in Figure 3. We extract both global (Section 3.1) and attention-based local (Section 3.2) features. Different from prior works, we attempt to learn essential features without utilizing bounding box information. By considering the ranking results in the training, we propose to select hard negatives and ambiguous positives for learning in Section 3.3. Note that we can further apply efficient search tools (*e.g.*, [18]) for fast retrieval.

3.1. Global feature extraction from ConvNets

For global features, we follow the state-of-the-art approaches to extract features from deep convolutional neural networks (ConvNets). Inspired by VLAD [16], NetVLAD [1] is a trainable end-to-end deep structure. It collects all the local features ($x_i \in \mathbb{R}^d$) quantized into the same VLAD center (c_k, k centers), and aggregates the difference between features and the center ($x_i - c_k$). To integrate with deep learning, Arandjelovic *et al*. [1] propose to utilize soft assignment ($\alpha_i \in \mathbb{R}^k$) with the existing network layers (Conv and softmax layers). The extracted global features (NetVLAD) are defined as

$$Global = NetVLAD_k = \sum_{i=1}^{N} \alpha_i^k (x_i - c_k), \qquad (1)$$

where α_i^k is the k-th dimension of soft assignment scores for the feature (x_i). N is the total number of features from convolutional layers, and equivalent to the size of extracted feature maps. $N = H * W$, where H and W are the height and width of the feature map. For a 320x320 image, the Conv5_3 feature map of VGGNet is 20x20, so the number of (local) features is 400 ($N = 20*20$). Note that those centers (c_k) can be learned simultaneously. For off-the-shelf features (with L2 normalization), the centers are calculated from the training data (*i.e.*, k-means without learning).

3.2. Attention-based local feature extraction

Although NetVLAD and other pooling methods can achieve better retrieval accuracy, we find that few of them focus on selecting essential features. For product search, images may contain cluttered background or irrelevant features. Meanwhile, those pre-trained models mainly focus on learning general features (*e.g.*, suit, cup, purse) without learning specific characteristics of products (*e.g.*, styles, patterns). A straightforward solution is to design weights for features (*e.g.*, Gaussian weights for center objects, or based on feature maps [20, 49]). Motivated by object detection, segmentation, and attention-based approaches [5, 8, 25, 28, 29, 34, 36, 43], we utilize an attention layer for extracting local features as shown in Figure 3(b). Based

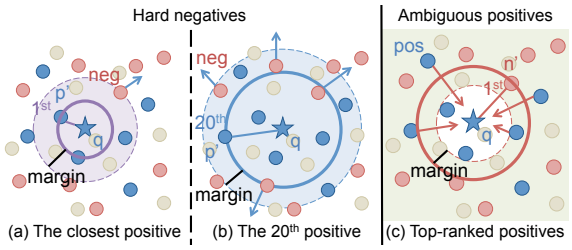


Figure 4. Our proposed rank-based candidate selection for feature learning. The blue star represents the query, blue circles are positives (relevant products), and red circles are negatives (irrelevant products). We attempt to push hard negatives away from the query, and make ambiguous positives close to the query for retrieving more relevant images in the top-ranked results. For hard negatives, we can select them based on (a) the closest positive or (b) a selected positive (*e.g.*, 20th in our experiments). (c) For ambiguous positives, based on the closest negative, we select the top-ranked (violated) positives.

on Eq. (1), local features are defined as

$$Local = f_k = \sum_{i=1}^{N} w_i[\alpha_i^k(x_i - c_k)], \tag{2}$$

where w_i is a weight value from attention layer for each feature (x_i). In this work, we apply a 1x1 convolutional layer with ReLU for the attention layer. Note that ReLU and sigmoid activations perform similar retrieval accuracy in our experiments, and we can also increase hidden layers in the attention layer. We only update the last few layers (*i.e.*, gray and orange regions in Figure 3) and retrain the remaining layers (*i.e.*, fixed layers before Conv5) with their original parameters trained by ImageNet for generalization. We can also utilize L2-norm scoring on the feature map [20, 28] for the weights (w_i). We will conduct this setting in our experiments.

3.3. Learning by rank-based candidate selection

Due to the large variety of consumer and shop photos, it is essential to utilize pairwise, triplet, or ranking information to mitigate the gap between cross-domain matching. For product search, we would be able to obtain a user query and a list of relevant products (positives). By aggregating all the queries and products, we can form a training set for learning better features. We attempt to leverage ranking results (*i.e.*, L2 distance) for selecting hard negatives (other products) and ambiguous positives in the training set. Given a query (q), we select a positive (p') image from the ground truth (target/relevant products), and the remaining irrelevant images as negatives. Motivated by NetVLAD [1], the loss of hard negative learning is formulated as

$$min \sum_{neg \in negatives} max(0, D(q, p') + margin - D(q, neg)), \tag{3}$$

where $D()$ is L2 distance function. As Figure 4(a) shows, it attempts to push those irrelevant (negative) images away from the relevant (positive, p') image. For training the first epoch, features are extracted from pre-trained model (*e.g.*, off-the-shelf NetVLAD). They will be updated by the learned network for later epochs (*e.g.*, trained NetVLAD or local features). For the same query in different epochs, the selected positives and negatives may be changed. It is time-consuming to extract all features in training set for every query; hence, we only extract new features after a predefined number of queries (*e.g.*, 1,000).

We observe that it is essential to select a suitable positive (p') image in the learning process. We select the positive image based on different tasks. As Figure 4(a) shows, the original NetVLAD sets the closest positive as p', because each query image only has one or few ground truth images (*e.g.*, DeepFashion). If it contains many relevant images (*e.g.*, Alibaba), we should set the positive image to a certain rank level as shown in Figure 4(b). It enforces top-ranked results with more relevant images. We also adopt hard negative mining by selecting those negatives near the query.

There are large numbers of negative images (irrelevant products). We usually sample a few negatives (*e.g.*, 1,000) for ranking in the training. It is possible that there are no hard negatives near the query. We further propose to pull ambiguous positives away from a negative (n'). Followed by Eq. (3), the loss of ambiguous positive learning is formulated as

$$min \sum_{pos \in positives} max(0, D(q, pos) + margin - D(q, n')), \tag{4}$$

where n' is the closest negative in our experiments. As Figure 4(c) shows, we selet top-ranked (violated) positives (*i.e.*, the green region in Figure 4(c)) in the learning. We will investigate the effects of these selection criteria in Section 5.

In this work, we focus on feature learning by leveraging ranking information for cross-domain product search. We can further improve the feature representation by incorporating other attributes or class information for multi-task learning. For both global and attention-based local features, we utilize the same loss functions in the learning process. However, we do not train them at the same time. This is because we attempt to fine-tune the network by different selected candidates (ranking results).

4. Experiment setup

4.1. Parameter setting and evaluation

Based on NetVLAD which is provided by the authors [1], we modify it for our proposed rank-based candidate selection. It is implemented in the MatConvNet (Convolutional Neural Networks for MATLAB) [41]. We choose VGG-16 as the pre-trained model for the ConvNets and L2

Table 1. The statistics of the consumer-to-shop benchmark in DeepFashion dataset. 'M' represents a subset of the original data. We randomly sample 1,000 items for training (T) and validation (V) sets in our experiments

	#Items	#Queries	#Database
Training	16,940	98,204	22,723
DeepFashionM-T	1,000	5,780	1,324
Validation	8,470	48,527	11,357
DeepFashionM-V	1,000	5,516	1,338
Testing	8,471	47,434	11,312

distance for CNN features. Followed by NetVLAD [1], we set the learning rate to be 0.001 and $margin = 0.1$, and recompute new features after 1,000 queries. For every 5 epochs, we half the learning rate and re-computation frequency as the same setting as NetVLAD [1]. We do not apply PCA with whitening (PCAW) for the final features, because we would like to evaluate the original effects on the proposed method. Therefore, for both global (NetVLAD) and attention-based local features, the dimension of features is 32,768 (with 64 VLAD centers). We calculate mean average precision (MAP) and retrieval accuracy at 20 (ACC@20) as our evaluation metric.

4.2. Initial setting for the attention layer

It is essential to set proper initial values for the attention layer (Section 3.2). We assume the original global (NetVLAD) features are good enough to represent images, so the goal of the initial weights is to be 1 for w_i (*i.e.*, selecting this feature, Eq. (2)). Based on the original setting of NetVLAD, each Conv feature (x_i) is L2 normalized before aggregation. The ideal vector (v, 1x1 convolution) of the weights would be the same vector as the Conv feature (*i.e.*, $x_i^T v = x_i^T x_i = 1$). We average all of the training features and apply L2 normalization for initializing v, so that we will have higher chance to generate 1 for w_i.

4.3. DeepFashion and Alibaba datasets

DeepFashion dataset [24] is the largest public available fashion dataset. It contains more than 800,000 images with label information (*e.g.*, categories, attributes, landmarks, bounding boxes). It further provides benchmarks for evaluation; hence, we conduct experiments on the provided "Consumer-to-Shop Clothes Retrieval" benchmark. We randomly sample 1,000 items (products) to form a small subset (**DeepFashionM**) for training and validation, and utilize the original testing set (8,471 items with 47K queries and 11K database images) in the experiments. Table 1 shows the statistics of the number of queries, database images, and items.

Table 2. The statistics of Alibaba dataset. 'M' and 'S' represent subsets of the original data. We only focus on the queries and their corresponding ground truth images for training (T) and validation (V) sets in our experiments

	#Queries	#Database
Training (with labels)	-	1,950,998
Testing (including val.)	4,984 (1,417)	3,195,334
AlibabaM-T	473	31,001
AlibabaM-V	472	32,055
AlibabaM (testing)	472	29,516
AlibabaM (testing) + 3M	472	3,195,334
AlibabaS	20	400

Alibaba dataset[6] is provided by Alibaba large-scale image search challenge (ALISC). It contains 4,984 query images (including 1,417 validation images), and 3.2 million database images for the final competition. They also provide 2 million images with category information (*e.g.*, clothes, snacks, beauty, furniture) for training. We do not have the ground truth for testing set (3,567 queries); hence, we split the validation set (1,417 queries) into training, validation, and testing sets. We collect the corresponding ground truth images to form **AlibabaM** as shown in Table 2. For AlibabaM (testing), we further conduct experiments on the whole database images (+3M) for large-scale experiments. As Figure 2 shows, **AlibabaS** contains 20 queries and 400 database images (*i.e.*, 20 ground truth images per product) with our manually annotated bounding boxes for evaluating object-level product search.

5. Experiments and discussions

5.1. Accuracy on global and local features

First, we evaluate the effects of global and local features on DeepFashionM-T (train) and DeepFashionM-V (val). Based on the NetVLAD learning framework, the results of each epoch are shown in Figure 5. We train on DeepFashionM-T with 5,780 queries but only evaluate accuracy on 1,000 (sampled) queries for both training and validation sets. For attention-based local features, we initialize the parameters by averaging all the training vectors as mentioned in Section 4.2. Hence, the result of pre-trained model (ep0) is slightly different from global features. It is still similar because we attempt to generate 1 for initialization. Although the final accuracy on the validation set are similar for two features, the learned features are complementary for product search. We will demonstrate the concatenated features can further improve the search results in next section. Meanwhile, we can obtain essential features based on the learned attention layer.

[6] Alibaba Large-scale Image Search Challenge (Alibaba)

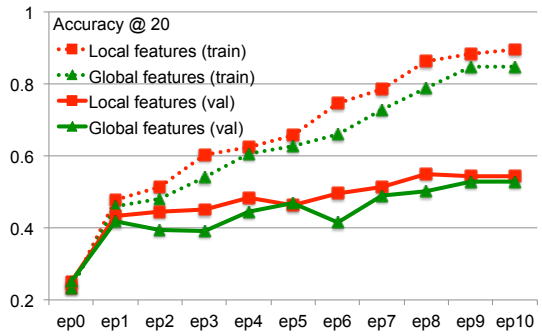


Figure 5. Feature learning results on DeepFashionM dataset. It shows that local features are slightly better than global features. The learned features are complementary, and we can further improve the results as shown in Figure 6.

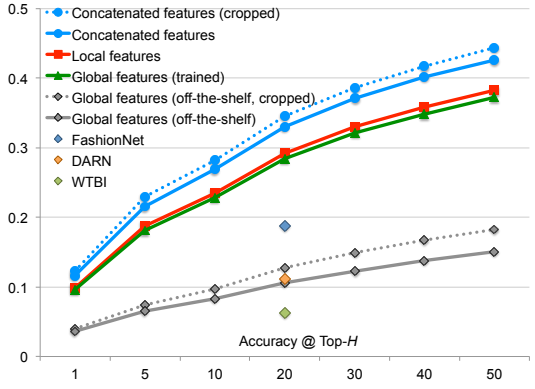


Figure 6. Overall comparison on DeepFashion testing set. The learned features can achieve better retrieval accuracy than other methods. Note that the retrieval accuracy of WTBI [21], DARN [14], and FashionNet [24] is reported from DeepFashion [24].

5.2. Combined features on DeepFashion testing

Based on the learned models from the training set (DeepFashionM-T), we choose the best ACC@20 in the validation set (DeepFashionM-V, Figure 5) as the final models for global features (ep9, 52.9%) and local features (ep8, 54.9%). We also fine-tune the network based on the L2-norm scoring for w_i in Eq. (2), and the best ACC@20 is 54% (ep9) in our experiment. The accuracy of attention-based local features is slightly better (54.9%) in DeepFashionM-V. For off-the-shelf features (without training), we only utilize sampled features from the training set to initialize the required parameters. We concatenated two learned features, and evaluate these features on DeepFashion testing.

As shown in Figure 6, we find that the original global feature (*i.e.*, the gray line, off-the-shelf NetVLAD in our experiments) is a strong baseline. After the learning process, both global and attention-based local features can beat the state-of-the-art methods (*i.e.*, WTBI [21], DARN [14], and FashionNet [24]). The accuracy is reported from the DeepFashion paper [24]. We can have huge accuracy gains for global features (from 10.6% to 28.4% on ACC@20). In DeepFashion (consumer-to-shop) dataset, the query (consumer) images are quite different from those online (shop) images, Hence, it is essential to utilize ranking information to learn better feature representations for cross-domain image retrieval.

It is worth noting that we only utilize positive (relevant products) and negative (other products) information from the training set and achieve the best results. The learning process of other approaches might contain additional information (*e.g.*, attribute, category, or landmark information) [24]. We think this is because the help of hard negative mining is very critical in the learning process for pushing negatives away from the query and positives. Although the learned global and local features achieve similar retrieval accuracy, the concatenated features can further improve the performance (*e.g.*, 33% on ACC@20). The learned features are complementary to each other.

DeepFashion dataset also provides bounding box information for each image. We perform product search on the whole and cropped (object) images. As shown in Figure 6, those dotted lines are the cropped retrieval results. The overall comparison is summarized in the upper half of Table 3. We can improve ACC@20 from 28.4% (WQ-WD) to 30.5% (CQ-CD) when we focus on the target objects (cropped images). The learning process can reduce the gap of retrieval accuracy between whole (WQ-WD) and cropped (CQ-CD) images (*e.g.*, 0.106 to 0.127 [+20%] versus 0.284 to 0.305 [+7%]).

5.3. Experiments on AlibabaS

To further demonstrate the generalization of the learned models from DeepFashionM (Section 5.1), we conduct experiments on AlibabaS dataset. Because the dataset do not provide the bounding box information, we manually annotate them (20 queries and 400 database images) for the experiments. As shown in the bottom half of Table 3, it shows that those learned features can still improve the retrieval accuracy (*e.g.*, from 0.610 to 0.813) on other datasets. When focusing on target objects (cropped images), we can improve MAP from 81.3% (WQ-WD) to 85.8% (CQ-CD). Local features can achieve better results on the three different settings (*i.e.*, WQ-WD, CQ-WD, and CQ-CD). This is because images in Alibaba dataset may contain more cluttered background. Attention-based local features can focus on those essential and relevant features for product search.

5.4. Learning with different hard negatives

We also apply the feature learning process on AlibabaM dataset. Similarly, we train features on AlibabaM-T with

Table 3. Retrieval accuracy on DeepFashion testing and AlibabaS datasets. The concatenated features can further improve the retrieval accuracy. Note that the models are learned from DeepFashionM-T in Section 5.1. 'W' and 'C' represent whole and cropped (object) images, and 'Q' and 'D' stand for query and database images

	WQ-WD	CQ-WD	CQ-CD
ACC@20	DeepFashion testing		
Global (off-the-shelf)	0.106	0.110	0.127
Global features	0.284	0.288	0.305
Local features	0.292	0.293	0.307
Concatenated features	0.330	0.330	0.345
MAP	AlibabaS		
Global (off-the-shelf)	0.610	0.672	0.732
Global features	0.813	0.818	0.858
Local features	0.836	0.854	0.901
Concatenated features	0.861	0.869	0.901

10 epochs, and select the best models from AlibabaM-V. As Table 4 shows, we can greatly improve the retrieval accuracy (*e.g.*, from 37.1% to 56.1%). Note that we follow the same evaluation measurement (MAP@20) as stated in ALISC (Alibaba dataset). If query images contain more than 20 ground truth images, we will set them as 20 when measuring MAP@20. The goal of this evaluation is to retrieve 20 relevant images in the top-ranked results (*i.e.*, the ideal MAP@20 will be 1). We adjust the closest positive images to the 20th positive image for p' in Eq. (3). With more ground truth images in the training set (database), the learning process will enforce the top-ranked results have more relevant products. It further improves the results on the validation set (from 56.1% to 61.8%). Attention-based local features achieve better retrieval accuracy than global features (61.8% versus 60.3%).

5.5. Learning with ambiguous (hard) positives

We further integrate hard negative learning (*i.e.*, 20th in Table 4) with ambiguous positive learning (+ Pos10) as mentioned in Eq. (4). For balancing the number of selected negative and positive candidates, we choose the same number for them (*i.e.*, 10 hard negatives and 10 ambiguous positives) in our experiments. As Table 4 shows, we can improve the retrieval accuracy when we pull ambiguous positives away from the closest negative. Especially, for global features, the MAP increases from 60.3% to 63.1%. Our proposed rank-based candidate selection can achieve the best accuracy (+15.8%, from 0.545 to 0.631) on global features. For attention-based local features, although the MAP continues increasing in the training set, it slightly increases in the validation set. This might be because those ambiguous positive images have similar local patterns to the closest negative image.

Table 4. Hard negative and ambiguous positive learning on AlibabaM dataset. We adjust the closest positive image to the 20th positive image for retrieving more relevant products in the top-ranked results, and achieve higher accuracy when considering ambiguous positives (+ Pos10)

MAP@20	Pre-trained	Closest	20th	+ Pos10
	Global features			
AlibabaM-T	0.316	0.595	0.781	0.846
AlibabaM-V	0.372	0.545	0.603	**0.631**
	Attention-based local features			
AlibabaM-T	0.315	0.614	0.807	0.838
AlibabaM-V	0.371	0.561	0.618	**0.623**

5.6. Experiments on AlibabaM (testing)

Based on the above trained models (DeepFashionM-T and AlibabaM-T), we conduct experiments on AlibabaM (testing), and utilize the best model from AlibabaM-V (*i.e.*, 20th + Pos10 in Table 4). As Table 5 shows, we can improve the retrieval results when training on the target dataset (DeepFashionM-T [L]: 52.2% vs. AlibabaM-T [L]: 60.9%). In the testing set, although the accuracy of off-the-shelf features on AlibabaM-O is worse than DeepFashionM-O (33.6% versus 36.3%), the learned features can achieve better accuracy (*i.e.*, larger performance gains). The concatenated features [L+G] achieve the best accuracy on MAP@20. It is true that doubling the feature dimensions can improve the accuracy; nevertheless, the improvement is larger when we concatenate different types of features (*i.e.*, [G+G][7]: 0.620 versus [G+L]: 0.630 from [G]: 0.612).

5.7. Experiments on AlibabaM (testing) + 3M

We further conduct experiments on large-scale image database (+3M). As shown in the bottom half of Table 5, these learned features still perform well on the large dataset. Because we do not have the ground truth of the original Alibaba testing set, we cannot compare with other works directly. The following comparison is just for references. Although the query set is different in our experiments and [42], we have almost the same number of queries in testing. We find that the ACC@20 is quite similar as reported in [42] (71.9%). However, they further consider classification loss in the learning process. That is to say, we can have further improvement when we integrate with other information. Without considering classification error, their method achieves 69.3% accuracy. Based on the similar learning information, our features can achieve better accuracy (73.1%). Besides, they utilize more training data (2

[7] We train additional global features based on the same training process, and concatenate the two global features for the experiments.

Table 5. Retrieval accuracy on AlibabaM testing set. We investigate the effects on learned local and global features from different training sets (models). We further evaluate on large-scale dataset (+ 3M images). It shows that features learned from DeepFashionM can provide a good retrieval accuracy. When training on a more relevant dataset (AlibabaM-T), we can further improve the results. 'CI' represents we utilize the given class information in the final ranking. 'O' stands for off-the-shelf features. 'L' and 'G' represent local and global features

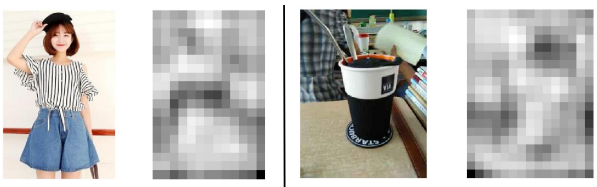
	MAP@20	ACC@20
Models	AlibabaM (testing)	
DeepFashionM-O	0.363	0.801
DeepFashionM-T [L]	0.522	0.900
AlibabaM-O	0.336	0.769
AlibabaM-T [L]	0.609	0.930
AlibabaM-T [G]	0.612	0.913
AlibabaM-T [L+G]	0.630	0.922
	AlibabaM (testing) + 3M	
DeepFashionM-T	0.234	0.659
AlibabaM-T [L]	0.291	0.695
AlibabaM-T [G]	0.299	0.731
AlibabaM-T [L+G]	0.309	0.729
AlibabaM-T [L+G] + CI	0.326	0.768

times than ours, *e.g.*, AlibabaM-T + AlibabaM-V) in the learning process. With more training data (*i.e.*, 473 to 945 queries [+ AlibabaM-V] and related ground truth images), we can improve the MAP@20 from 61% to 65% on AlibabaM (testing), and achieve 0.325 (from 0.291) on AlibabaM (testing) + 3M.

The concatenated features achieve 0.309 on MAP@20 for the large-scale dataset. If we can know the coarse category information (*e.g.*, clothes, beauty) for both query and database images, we can focus on those related categories. We utilize the given 10 class information from Alibaba dataset, calculate ranking scores on those products within the same class, and achieve the best retrieval accuracy (76.8%). Note that we can also learn to classify these 10 classes if the class information is not given.

5.8. Dimension reduction results

For a practical system, we can apply dimension reduction (*e.g.*, PCAW, PCA with whitening) for obtaining low-dimensional features (*e.g.*, 4,096 or 256). We learn the projection matrix on AlibabaM-T, and conduct experiments on AlibabaM (testing). For 4,096-d features, we only slightly decrease the retrieval accuracy on both global (0.612 to 0.585) and local (0.609 to 0.588) features. For 256-d features, we can still retain reasonable accuracy ([G]: 0.548 and [L]: 0.545) in our experiments. With low-dimensional features, we only require few milliseconds (*e.g.*, 11ms for 4,096-d, 1ms for 256-d) for computing L2 distance with around 30K database images. Note that we can further improve the accuracy by integrating the dimension reduction as an additional layer in the end-to-end learning as [12], and utilize existing tool for faster search [18].



(a) Target object: shorts (b) Target object: cup

Figure 7. The weights of local features for two examples. We find that the extracted features may be able to ignore irrelevant parts (*e.g.*, hands or face). We normalize the values to [0 (black), 255 (white)] for visualization.

5.9. Visualization on the attention layer

We visualize the weights of local features in Figure 7. For the target 'shorts' in Figure 7(a), we could focus on the lower body and ignore irrelevant parts such as hands and face. Hence, we might be able to generate better feature representations by learning from those important objects/parts (*e.g.*, logo) and ignoring cluttered background (*e.g.*, words). The purpose of the attention layer is to learn essential features from training data. Hence, it indeed learns a saliency model that is good for retrieval.

6. Conclusions and future works

For product search, we observe that we can utilize large-scale e-commerce data for learning better feature representations. Based on the item/product information, we propose a rank-based candidate selection for feature learning, and investigate the effects on both global and attention-based local features. Experiment results show that we can improve the retrieval accuracy by leveraging hard negatives and ambiguous positives in the learning process. In the future, we will investigate different loss functions and attention layers for learning better features, and learn on larger training sets. We will also integrate additional information in the learning process (*e.g.*, category) as adopted in [9] for segmentation, or apply spatial transformer networks [15] for solving rotation issues.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 104-2622-8-002-002 and MOST 105-2218-E-002-032, and in part by MediaTek Inc. and grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer. The Alibaba dataset is provided by Alibaba Group. The work of Y.-H. Kuo was supported by the Microsoft Research Asia Fellowship.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPR Workshops*, 2015.

[3] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, 2015.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014.

[5] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):185–207, Jan. 2013.

[6] D. M. Chen and B. Girod. Memory-efficient image databases for mobile visual search. *IEEE MultiMedia*, 21(1):14–23, 2014.

[7] Z.-Q. Cheng, Y. Liu, X. Wu, and X.-S. Hua. Video ecommerce: Towards online video advertising. In *ACM Multimedia*, pages 1365–1374, 2016.

[8] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015.

[9] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[10] B. Girod, V. Chandrasekhar, D. M. Chen, N. Cheung, R. Grzeszczuk, Y. A. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham. Mobile visual search. *IEEE Signal Process. Mag.*, 28(4):61–76, 2011.

[11] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407, 2014.

[12] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, pages 241–257, 2016.

[13] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang. Mobile product search with bag of hash bits and boundary reranking. In *CVPR*, 2012.

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

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015.

[16] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[17] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *KDD*, pages 1889–1898, 2015.

[18] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734, 2017.

[19] H. Kaiming, Z. Xiangyu, R. Shaoqing, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[20] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCV Workshops*, pages 685–701, 2016.

[21] 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.

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

[23] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, 2015.

[24] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[25] A. Mahendran and A. Vedaldi. Salient deconvolutional networks. In *European Conference on Computer Vision*, 2016.

[26] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *ICMR*, pages 327–331, 2016.

[27] J. Y.-H. Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval. In *CVPR Workshops*, 2015.

[28] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017.

[29] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015.

[30] S. Qi, K. Zawlin, H. Zhang, X. Wang, K. Gao, L. Yao, and T. seng Chua. Saliency meets spatial quantization: A practical framework for large scale product search. In *IEEE ICME Workshops*, 2016.

[31] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

[32] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, pages 512–519, 2014.

[33] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *ICLR Workshop*, 2015.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[35] A. Salvador, X. Giro-i Nieto, F. Marques, and S. Satoh. Faster r-cnn features for instance search. In *CVPR Workshops*, 2016.

[36] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[38] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[39] V. L. T. Trzcinski, M. Christoudias and P. Fua. Boosting Binary Keypoint Descriptors. In *CVPR*, 2013.

[40] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*, 2016.

[41] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *ACM Multimedia*, 2015.

[42] X. Wang, Z. Sun, W. Zhang, Y. Zhou, and Y.-G. Jiang. Matching user photos to online products with robust deep features. In *ACM ICMR*, 2016.

[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[44] H.-F. Yang, K. Lin, and C.-S. Chen. Cross-batch reference learning for deep classification and retrieval. In *ACM Multimedia*, pages 1237–1246, 2016.

[45] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014.

[46] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[47] A. Zhai, D. Kislyuk, Y. Jing, M. Feng, E. Tzeng, J. Donahue, Y. L. Du, and T. Darrell. Visual discovery at pinterest. In *WWW*, 2017.

[48] N. Zhang, T. Mei, X.-S. Hua, L. Guan, and S. Li. Taptell: Interactive visual search for mobile task recommendation. *JVCI*, May 2015.

[49] S. Zhao, Y. Xu, and Y. Han. Large-scale e-commerce image retrieval with top-weighted convolutional neural networks. In *ICMR*, pages 285–288, 2016.