Attributes and Categories for Generic Instance Search from One Example

Ran Tao[1], Arnold W.M. Smeulders[1], Shih-Fu Chang[2]
[1]ISLA, Informatics Institute, University of Amsterdam, The Netherlands
[2]Department of Electrical Engineering, Columbia University, USA

Abstract

This paper aims for generic instance search from one example where the instance can be an arbitrary 3D object like shoes, not just near-planar and one-sided instances like buildings and logos. Firstly, we evaluate state-of-the-art instance search methods on this problem. We observe that what works for buildings loses its generality on shoes. Secondly, we propose to use automatically learned category-specific attributes to address the large appearance variations present in generic instance search. On the problem of searching among instances from the same category as the query, the category-specific attributes outperform existing approaches by a large margin. On a shoe dataset containing 6624 shoe images recorded from all viewing angles, we improve the performance from 36.73 to 56.56 using category-specific attributes. Thirdly, we extend our methods to search objects without restricting to the specifically known category. We show the combination of category-level information and the category-specific attributes is superior to combining category-level information with low-level features such as Fisher vector.

1. Introduction

In instance search, the objective is to retrieve all images of a specific object given a few query examples of that object [3, 16, 25, 31]. We consider the challenging case of only 1 query image and admitting large differences in the imaging angle and other imaging conditions between the query image and the target images. A very hard case is a query specified in frontal view while the relevant images in the search set show a view from the back which has never been seen before. Humans solve the search task by employing two types of general knowledge. First, when the query instance is a certain class, say a female, answers should be restricted to be from the same class. And, queries in the frontal view showing one attribute, say brown hair, will limit answers to show the same or no such attribute, even when the viewpoint is from the back. In this paper, we use and evaluate these two types of general knowledge to handle a wide variety of circumstances for instance search.

In instance search, excellent results have been achieved by restricting the search to buildings [3, 4, 29]. Searching buildings can be used in location recognition and 3D reconstruction. Another set of good results has been achieved in searching for logos [20, 34, 40] for the estimation of brand exposure. And, [44] searches for book and magazine covers. All these cases of instance search show good results for near-planar, and one-sided objects. In this work, we aim for broader classes of query instances. We aim to perform generic instance search from 1 example. *Generic* implies we consider arbitrary objects, and not just planar objects. And, *generic* implies we aim to use one approach not optimized for a certain type of instances, such as RANSAC matching the planarity of the objects. In our case, instances can be buildings and logos, but also shoes, clothes and other objects.

The challenge in instance search is to represent the query image invariant to the (unknown) appearance variations of the query while maintaining a sufficiently rich representation to permit distinction from other, similar instances. To solve this, most existing approaches in instance search match the appearance of spots in the potential target to the query [16, 19, 29, 40, 41]. The quality of match in these approaches between two images is the sum of similarities over all local descriptor pairs. The difference between the cited approaches lies in the way local descriptors are encoded and in the computation of the similarity. Good performance has been achieved by this paradigm on buildings, logos and scenes from a distance. However, when searching for an arbitrary thing with wider range of viewpoint variability, more sides, and possibly having self-occlusion and non-rigid deformation, these methods are likely to fail as local descriptor matching becomes unreliable in these cases. For instance search from only one example, more information on the target object is always needed, coupled to a representation robust against appearance variations to successfully address generic instance search.

In this paper we propose to use attribute representation [10, 23] to handle a wide range of visual appearances. In this way, we aim to be more robust against ap-

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

pearance variations than the low-level image representation, like the bag-of-words histogram [39] and Fisher vector [28]. We rely on attribute representation as it has been shown advantageous in classification when training examples are insufficiently covering all variations in low-level feature space [10, 45], surely present in the one-example challenging case. We propose to learn automatically a list of category-specific and non-semantic attributes, which are discriminative among instances of the same category. An instance can be represented as a specific combination of the attributes, and instance search boils down to finding the most similar combinations of attributes.

In order to address the possible confusion of the query with instances from other categories, we propose to first search at the concept-level and then zoom in to search within the category. In this way, we are able to reduce the search space of all pixel configurations tremendously while still being reasonably sure that we do not lose the target.

It is advantageous when there is only 1 query image, to use slightly more user provided information. In addition to the interactive specification of the object region in the query image, in the paper we require the specification of the category the query instance belongs to.

2. Related work

Most approaches in instance search rely on gathering matches of local image descriptors [39, 29, 16, 41, 40, 19], where the differences reside in the way the local descriptors are encoded and the matching score of two descriptors is evaluated. Bag-of-words (BoW) [39, 29] encodes a local descriptor by the index of the nearest visual word. Hamming embedding [16] improves upon BoW by adding an extra binary code to better describe the position of the local descriptor in space. The matching score of a pair of descriptors is 1 if they are in the same word and the Hamming distance between binary signatures is smaller than a certain threshold. VLAD [17] and Fisher vector [27] improve over BoW by representing the local descriptor with an extra residual vector, obtained by subtracting the mean of the visual word or the Gaussian component respectively. In VLAD and Fisher vector, the score of two descriptors is the dot product of the residuals when they are in the same word, and 0 otherwise. [41, 40] improve VLAD and Fisher vector by replacing the dot product by a thresholded polynomial similarity and an exponential similarity respectively to give more credits to closer descriptor pairs. [19] encodes a local descriptor by only considering the directions to the visual word centers, not the magnitudes, outperforming Fisher vector on instance search. With these methods, good performance has been achieved on buildings, logos, and scenes from a distance. These instances can be conceived as near-planar and one-sided. For buildings, logos, and scenes from a distance the variation in the viewing angle is limited to a quadrant of 90 degrees at most out of the full 360 circle. For limited variations in viewpoint, matches of local descriptors can be reliably established between the query and a relevant example. In this work, we consider generic instance search, not only flat instance search, where the instance can be an arbitrary object with wider range of viewpoint variability and more sides. We evaluate existing methods for approximately flat instance search on this problem of generic instance instance.

Attributes [10, 11, 23] have received much attention recently. They are used to represent common visual properties of different objects. Attribute representation has been used for image classification [10, 45, 2]. Attributes have been shown to be advantageous when the training examples are insufficiently covering the appearance variations in the low-level feature space [10, 45]. Inspired by this, we propose to use attribute representation to address generic instance search, where there is only 1 example available and there still exists a wide range of appearance variations.

Attributes have been used for image retrieval [38, 21, 46, 45, 32]. In [38, 21, 46], the query is defined by textual attributes instead of images and the goal is to return images exhibiting query attributes. The query attributes need to be semantically meaningful. In this work, we address instance search given one query image, which is a different task as the correct answers have to exhibit the same instance, and we use non-semantic attributes. [45, 32] also consider non-semantic attributes, but for category retrieval instead of instance search.

The use of category-level information to improve instance search has been explored in [48, 8, 13]. [13] uses category labels to learn a projection to map the original feature to a lower-dimensional space such that the lower-dimensional feature incorporates certain category-level information. In this work, instead of learning a feature mapping, we augment the original representation with additional features to capture the category-level information. [8] expands Fisher vector representation with the concept classifier output vector of the 2659 concepts from Large Scale Concept Ontology for Multimedia (LSCOM) [24]. In [48], a 1000-dimensional concept representation [1] is utilized to refine the inverted index on the basis of semantic consistency between images. Both [48] and [8] combine category-level information with low-level representation. In this work, we consider the combination of category-level information with category-specific attributes, not low-level representation.

2.1. Contributions

Our work makes three contributions. We propose to pursue generic instance search from 1 example where the instance can be an arbitrary 3D object recorded from a wide range of imaging angles. We demonstrate that this problem

is harder than the approximately flat and one-sided instance search of buildings [29], logos [20] and remote scenes [16]. We evaluate state-of-the-art approaches on this problem. We observe what works best for buildings loses its generality for shoes and reversely what works worse for buildings may work well for shoes.

Secondly, we propose to use automatically learned category-specific attributes to handle the wide range of appearance variations in generic instance search. Here we assume we know the category of the query instance which provides critical knowledge when there is only one query image. Information of the query category can be given through interactive user interface or automatic image categorization (*e.g.*, shoe, dress, *etc.*). On the problem of searching among instances from the same category as the query, category-specific attributes outperform existing instance search methods by a large margin when large appearance variations exist.

As our third contribution, we extend our methods to search objects without restricting to the known category. We propose to augment the category-specific attributes with category-level information which is carried by the deep learning features learned from large-scale image categorization and the category-level classification scores. We show combining category-level information with category-specific attributes is superior to combining category information with low-level features such as Fisher vector.

3. The difficulty of generic instance search

The first question we raise in this work is how the state-of-the-art methods perform on generic instance search from 1 example where the query instance can be an arbitrary thing. *Can we search for other objects like shoes with the same method for buildings?* To that end, we evaluate several existing instance search algorithms on two datasets, the Oxford buildings dataset [29] and a shoe dataset collected in this work.

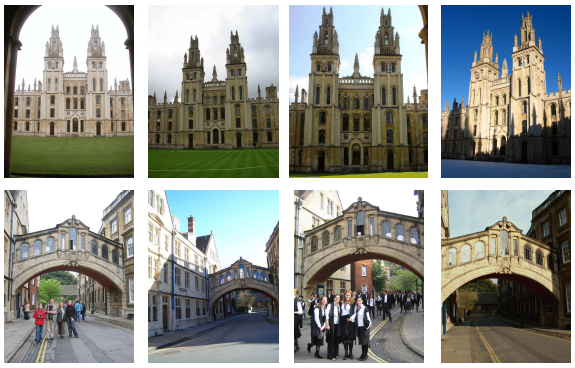
We evaluate four methods. **M1 (ExpVLAD):** A recent work [40] introduces locality at two levels to improve instance search from one example. The method considers locality in the picture by evaluating multiple candidate locations in each of the database images. It also considers locality in the feature space by efficiently employing a large visual vocabulary for VLAD and Fisher vector and by an exponential similarity function to give disproportionally high scores on close local descriptor pairs. The locality in the picture was shown effective when searching for instances covering only a part of the image. And the the locality in the feature space was shown useful on all the datasets considered in the paper. **M2 (Triemb):** [19] proposes triangulation embedding and democratic aggregation. The triangulation embedding encodes a local descriptor with respect to the visual word centers using only directions, not magnitudes. As shown in the paper, the triangulation embedding outperforms Fisher vector [35]. The democratic aggregation assigns a weight to each local descriptor extracted from an image to ensure all descriptors contribute equally to the self-similarity of the image. This aggregation scheme was shown better than the sum aggregation. **M3 (Fisher):** We also consider Fisher vector as it has been widely applied in instance search and object categorization where good performance has been reported [18, 35]. **M4 (Deep):** It has been shown recently that the activations in the top layers of a deep convolutional neural network (CNN) [22] serve as good features for several computer vision tasks [33, 6, 12]. We evaluate the deep learning features on generic instance search.

Datasets. Oxford buildings dataset [29], often referred to as *Oxford5k*, contains 5062 images downloaded from Flickr. 55 queries of Oxford landmarks are defined, each by a query example. *Oxford5k* is one of the most popular available datasets for instance search, which has been used by many works to evaluate their approaches. Figure 1a shows two buildings from the dataset.

As a second dataset, we collect a set of shoe images from Amazon[1]. It consists of 1000 different shoes and in total 6624 images. Each shoe is recorded from multiple viewing angles including views from front, back, top, bottom, side and some others. One image of a shoe is considered as the query and the goal is to retrieve all the other images of the same shoe. Although these images are with clean background as often seen on shopping websites, this is a challenging dataset mainly due to the presence of considerably large viewpoint variations and self-occlusion. We refer to this dataset as *CleanShoes*. Figure 1b shows 3 shoes from *CleanShoes*. There is a shoe dataset available, proposed by [7]. However, this dataset is not suited for instance search as it does not contain multiple images for one shoe. [37] also considers shoe images, but the images are well aligned, whereas the images in *CleanShoes* provide a much wider range of viewpoint variations.

Implementation details. For M1, M2 and M3, we use the Hessian-Affine detector [26] to extract interest points. The SIFT descriptors are turned into RootSIFT [4]. The full 128D descriptors are used for M1 and M2, following [40, 19], while for Fisher vector, the local descriptor is reduced to 64D using PCA, as the PCA reduction has been shown important for Fisher vector [18, 35]. The vocabularies for *Oxford5k* are trained on Paris buildings [30], and the vocabularies for *CleanShoes* are learned on a random subset of the dataset. The vocabulary size is 20k, 64 and 256 for M1, M2 and M3 respectively, following the corresponding references [40, 19, 18]. We additionally run a version of Fisher vector with densely sampled RGB-SIFT descrip-

[1]The properties are with the respective owners. The images are shown here only for scientific purpose.



(a)

(b)

Figure 1: (a) Examples of two buildings from *Oxford5k*, and (b) Examples of three shoes from *CleanShoes*. There exists a much wider range of viewpoint variability in the shoe images.

tors [43], denoted by *Fisher-D*. For M4, we implement the CNN proposed in [22] and use the output of the second fully connected layer as the image representation. The CNN is trained using ImageNet categories. The search performance is measured using mean average precision (mAP).

Results and discussions. Table 1 summarizes the results on *Oxford5k* and *CleanShoes*. *ExpVLAD* adopts a large vocabulary with 20k visual words and the exponential similarity function. As a result, only close descriptor pairs in the feature space matter in measuring the similarity of two examples. This results in better performance than others on *Oxford5k* where close and relevant local descriptor pairs do exist. However, on the shoe images where close and true matches of local descriptors are rarely present due to the large appearance variations, *ExpVLAD* achieves lowest performance. Both *Triemb* and *Fisher* obtain quite good results on buildings but the results on shoes are low. This is again caused by the fact that local descriptor matching is not reliable on the shoe images where large viewing angle differences are present. *Triemb* outperforms *Fisher*, consistent with the observations in [19]. In this work, we do not consider the RN normalization [19] because it requires extra training data to learn the projection matrix and it does not affect the conclusion we make here. *Fisher-D* works better than *Fisher* on *CleanShoes* by using color information and densely sampled points. Color is a useful cue for discriminating different shoes, and dense sampling is better than interest point detector on shoes which do not have rich textural patterns. However, *Fisher-D* does not improve over *Fisher* on *Oxford5k*.

method↓	Oxford5k	CleanShoes
ExpVLAD	**76.54**	16.14
Triemb	61.64	25.06
Fisher	56.72	20.94
Fisher-D	53.62	36.27
Deep	45.50	**36.73**

Table 1: Performance of different methods for instance search: ExpVLAD [40], Triemb [19], Fisher [18] and Deep [22, 6]. For Fisher vector, we consider two versions. *Fisher* denotes the version with interest points and SIFT descriptors, and *Fisher-D* uses densely sampled RGB-SIFT descriptors. The results on *Oxford5k* are based on our own implementation, consistent with those reported in [40, 19, 5, 6]. *ExpVLAD* achieves better performance than others on *Oxford5k*, but gives lowest result on *CleanShoes*. On the other hand, *Deep* obtains best performance on *CleanShoes*, but has lower result than others on *Oxford5k*.

Overall, the performance on shoes is much lower than on the buildings. More interestingly, *ExpVLAD* achieves better performance than others on *Oxford5k*, but gives lowest result on *CleanShoes*. On the other hand, *Deep* obtains best performance on *CleanShoes*, but has lower result than others on *Oxford5k*. We conclude that none of the existing methods work well on both buildings, as an example of 2D near-planar instance search, and shoes, as an example of 3D full-view instance search.

4. Attributes for generic instance search

Attributes have been shown advantageous in categorization when the training examples are insufficiently covering the appearance variations in the low-level feature space [10, 45, 2]. In our problem, there is only 1 example available and there still exists a wide range of appearance variations. As a second question we raise in the paper, *can we employ attributes to address generic instance search?* In this section, we focus on searching among things known to be of the same category using category-specific attributes.

In the literature, two types of attributes have been studied, semantic attributes [23, 2] and non-semantic attributes [45, 36]. Obtaining semantic attributes requires a considerable amount of human efforts and sometimes domain expertise, making it hard to scale up to a large number of attributes. Moreover, the manually picked attributes are not guaranteed to be discriminative for the task under consideration [45]. On the other hand, non-semantic attributes do not need human annotation and have the capacity to be optimized for the recognition task [45, 36]. For some tasks, like zero-shot learning [2] and image retrieval by textual attributes query [38], it is necessary to use human understandable attributes. However, in instance search given 1 image query, having semantic meaning is not really necessary. In this work, we use non-semantic data-driven attributes. Provided with a set of training instances from a certain category, we aim to learn automatically a list of category-specific attributes and use these attribute to perform instance search on new instances from the same category.

We consider three criteria for learning the category-specific attributes. As the first criterion, the attributes need to be able to make distinction among the instances. The second criterion is that the attributes need to be shareable among visually similar training instances. Attributes specific to one training instance are less likely to be descriptive for unknown instances than those shared by several training instances. And sharing needs to be restricted only among visually similar training instances as the latent common patterns among visually dissimilar instances are less likely to be detected on new instances. The third criterion is that the redundancy between the learned attributes needs to be low. Considering the above three standards, we employ an existing approach [45] which fits well with our considerations. Given n training instances from the same category and aiming for k attributes, the method learns an instance-attribute mapping $A \in \mathbb{R}^{n \times k}$ by

$$\underset{A}{\text{maximize}} \quad f_1(A) + \lambda f_2(A) + \gamma f_3(A), \tag{1}$$

where $f_1(A)$, $f_2(A)$ and $f_3(A)$ are defined as follows:

$$\begin{aligned} f_1(A) &= \sum_{i,j}^{n} \|A_{i\cdot} - A_{j\cdot}\|_2^2, \\ f_2(A) &= -\sum_{i,j}^{n} S_{ij} \|A_{i\cdot} - A_{j\cdot}\|_2^2, \\ f_3(A) &= -\|A^T A - I\|_F^2. \end{aligned} \tag{2}$$

$A_{i\cdot}$ is the attribute representation of the i-th instance. $f_1(A)$ ensures instance separability. S_{ij} represents visual similarity between instance i and instance j, measured *a priori* in the low-level feature space, and $f_2(A)$ encourages similar attribute representations between visually similar instances, inducing shareable attributes. $f_3(A)$ penalizes large redundancy between attributes. After getting the instance-attribute mapping A where each column represents one attribute, k attribute classifiers are learned. The learned classifiers are applied on the examples of unknown instances to generate attribute representation, and search is done in the attribute space.

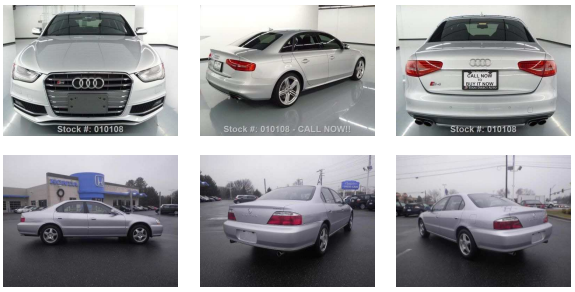


Figure 2: Examples of two cars.

Datasets. We evaluate the learned category-specific attributes on shoes, cars and buildings. For shoes, we consider the *CleanShoes* described in the previous section. For cars, we collect 1110 images of 270 cars from eBay. We denote it by *Cars*. One image of a car is considered as query and the goal is to find other images of the same car. Figure 2 shows some examples of two cars[2]. For buildings, we compose a small dataset by gathering all the 567 images of the 55 Oxford landmarks from *Oxford5k*. We denote it by *OxfordPure*. We reuse the 55 queries defined in *Oxford5k*.

For training shoe-specific attributes, we collect 2100 images of 300 shoes from Amazon, the same source where we collect *CleanShoes*. For learning car-specific attributes, we collect 1520 images of 300 cars from eBay. For learning the building-specific attributes, we use a subset of the large building dataset introduced in [6]. We randomly pick 30 images per class and select automatically the 300 classes that are most relevant to *OxfordPure* according to the visual similarity. We end up with in total 8756 images as some URLs are broken and some classes have less than 30 examples. For all shoes, cars and buildings, the instances in the evaluation sets are not present in the training sets.

Implementation details. We use Fisher vector [35] with densely sampled RGB-SIFT [43] as the underlying representation to learn the attribute classifiers. The same Fisher representation is used to select the relevant training examples for learning building-specific attributes. The visual proximity matrix S in equation 2 is built as a mutual 60-NN adjacent matrix. The proximity between two training instances is computed as the average similarity between the images of the two instances in the Fisher vector space.

[2]The properties are with the respective owners. The images are shown here only for scientific purpose.

attributes↓	*dim*	CleanShoes
Manual	40	18.99
Randomized	40	28.15
Learned	40	37.59
Randomized	1000	55.36
Learned	1000	56.56

Table 2: Performance of different attributes. The result of randomized attributes is the average of 5 runs. Data-driven non-semantic attributes outperform manually defined attributes. The learned attributes [45] achieve best performance, but if the number of attributes is high, the small difference in performance with randomized attributes permits skipping the learning phase.

Results and discussions. In the first experiment, we compare the learned attributes [45] with two alternatives, manually defined attributes and randomized attributes, on *CleanShoes*. For manually defined attributes, we use the list of attributes proposed by [14]. We manually annotate the 2100 training images. In the reference, 42 attributes are defined. However, we merge *super-high* and *high* of "upper" and "heel height" because it is hard to annotate *super-high* and *high* as two different attributes, resulting in 40 attributes. To generate a randomized attribute, we randomly split the training instances into two groups, assuming that instances in one group have a common visual aspect which the other instances do not. Such randomized attributes have also been considered in [10] for image categorization. As shown in Table 2, with the same number of attributes, data-driven non-semantic attributes work significantly better than the manual attributes. Learned attributes are considerably better than the randomized ones when the number of attributes is low. The random splits do not take into account the underlying visual proximity of the training instances and the attributes cannot generalize well on new instances. Such problem is alleviated when a large number of splits are considered. Figure 3 shows three learned attributes. Although the attributes have no explicit semantic meaning, they do capture common patterns between shoes.

In the second experiment, we compare the learned attributes with existing approaches evaluated in the previous section on *CleanShoes*, *Cars* and *OxfordPure*. Table 3 shows the results. The attribute representation works significantly better than the others on the shoe dataset and the car dataset. Attributes are superior in addressing the large appearance variations caused by the large imaging angle difference present in the shoe and car images, even though the attributes are learned from other instances. The attribute representation also works well on the buildings. Besides, compared to others, attribute representation has a much lower dimensionality. We conclude that the proposed method using automatically learned category-specific attributes is more generic than other approaches.

Figure 3: Three learned attributes. Each row is one attribute and the shoes are the ones that have high response for that attribute. Although the automatically learned attributes have no semantic meaning, apparently they capture sharing patterns among similar shoes. The first attribute represents boots. The second attribute describes the high heels and the third one captures the round toe.

method↓	*dim*	CleanShoes	Cars	OxfordPure
ExpVLAD	—	16.14	23.70	**87.01**
Triemb	8064	25.06	18.56	75.33
Fisher	16384	20.94	18.37	70.81
Fisher-D	40960	36.27	20.89	67.41
Deep	4096	36.73	22.36	59.48
Attributes	1000	**56.56**	**51.11**	77.36

Table 3: Performance of learned attributes and existing methods [40, 19, 18, 22]. Attributes achieve much better performance than others on shoes and cars, and are on par with others on buildings.

5. Categories and attributes for generic instance search

In this section, we consider searching for an instance from a dataset which contains instances from various categories. As the category-specific attributes are optimized to make distinctions among instances of the same category, they might not be able to distinguish the instance of interest from the instances of other categories. In order to address the possible confusion of the query instance with instances from other categories, we propose to use the category-level information also.

We consider two ways to capture the category-level information. First, we adopt the 4096-dimensional output of

Figure 4: Examples of two shoes from *StreetShoes*. The left are the query images and the rest are inserted into Pascal VOC 2007 classification dataset [9] which provides distractor images. We only consider the shoe segment in the query example to ensure the target is clear.

method↓	StreetShoes	Oxford5k
Deep(128D)	21.68	62.36
Fisher(128D)	9.38	50.00
Attributes(128D)	3.10	58.37
Deep + Fisher	19.76	61.24
Deep + Attributes	18.43	**72.16**
Deep + Classifier + Fisher	22.70	56.50
Deep + Classifier + Attributes	**30.45**	65.14

Table 4: Performance on *StreetShoes* and *Oxford5k*. The proposed method of combining the category-specific attributes with two types of category-level information outperforms the combination of category-level information with Fisher vector.

the second fully connected layer of a CNN [22] as an additional feature, as it has been shown the activations of the top layers of a CNN capture high-level category-related information [47]. The CNN is trained using ImageNet categories. Second, we build a general category classifier to alleviate the potential problem of the deep learning feature, namely the deep learning feature may bring examples that have common elements with the query instance even if they are irrelevant, such as skins for shoes and skies for buildings. Combining the two types of category-level information with the category-specific attributes, the similarity between a query q and an example d in the search set is computed by

$$S(q,d) = S_{deep}(q,d) + S_{class}(d) + S_{attr}(q,d), \quad (3)$$

where $S_{deep}(q,d)$ is the similarity of q and d in the deep learning feature space, $S_{class}(d)$ is the classification response on d and $S_{attr}(q,d)$ is the similarity in the attribute space.

Datasets. We evaluate on shoes and buildings. A small set of 15 shoes and in total 59 images is collected from two fashion blogs[3]. These images are recorded in streets with cluttered background, different from the 'clean' images in *CleanShoes*. We consider one image of a shoe as the query and aim to find other images of the same shoe. The shoe images are inserted into the test and validation parts of the Pascal VOC 2007 classification dataset [9]. The Pascal dataset provides distractor images. We refer to the dataset containing the shoe images and the distractors as *StreetShoes*. Figure 4 shows two examples. To learn the shoe classifier, we use the 300 'clean' shoes for attributes learning in the previous section as positive examples and consider the training part of the Pascal VOC 2007 classification dataset as negative examples.

For buildings, we use *Oxford5k*. To train the building classifier, we use all the training images for attribute learning in the previous section as positive examples and consider the images from the Pascal VOC 2007 classification dataset as negative examples. The building images in the training set are not clean, still containing elements like skies and trees, and we expect the building classifier will not be as useful as the shoe classifier.

Implementation details. We only consider the object region in the query image to ensure the target is clear. It is worth to mention that although only the object part in the query image is considered, we cannot completely get rid of skins for some shoes and skies for some buildings. We use selective search [42] to generate many candidate locations in each database image and search locally in the images as [40]. We adopt a short representation with 128 dimensions. Specifically, we reduce the dimensionality of the deep learning features and the attribute representations with a PCA reduction. And for Fisher vectors, we adopt the whitening technique proposed in [15], proven better than PCA. We reuse the attribute classifiers from the previous section.

Results and discussions. The results are shown in Table 4. On *StreetShoes*, the proposed method of combining category-specific attributes with two types of category-level information achieves the best performance, 30.45 in mAP. We observe that when considering deep features alone as the category-level information, the system brings many examples of skins. The shoe classifier trained on 'clean' shoe images is effective in eliminating these irrelevant ex-

[3]http://www.pursuitofshoes.com/ and http://www.seaofshoes.com/. The properties are with the respective owners. The images are shown here only for scientific purpose.

Figure 5: Top 5 returned images for two query instances. For the first instance, it has 5 relevant images in the search set, and 4 of them are returned in the top 5 positions. For the second instance, there is only 1 relevant example in the search set and it is returned at the first position. The irrelevant results are very similar visually to the query.

amples. Figure 5 shows the top 5 results of two query instances returned by the proposed method. On *Oxford5k*, the same method achieves the second best performance, 65.14. The best performance is obtained by combining category-specific attributes with deep features only. The building classifier does not help probably due to two reasons. One is that the building classifier is learned on cluttered images with skies and grasses and hence cannot well handle the problem of the deep features mentioned before. Had we had clean building images for training, the classifier would probably help improve the performance. Another reason is that a large portion of the images in *Oxford5k* are buildings and the classifier may push examples of irrelevant buildings to the top of the ranking list. The building classifier would probably be more useful in a larger dataset which contains many non-building examples. The fact that *Oxford5k* contains many building images also explains why category-specific attributes alone can already achieve quite good performance, 58.37. Overall, we conclude that the proposed method of combining the category-specific attributes with two types of category-level information is effective, outperforming the combination of category-level information with Fisher vector.

6. Conclusion

In this paper, we pursue generic instance search from 1 example. Firstly, we evaluate existing instance search approaches on the problem of generic instance search, illustrated on buildings and shoes, two contrasting categories of objects. We observe that what works for buildings does not necessarily work for shoes. For instance, [40] employs large visual vocabularies and the exponential similarity function to emphasize close matches of local descriptors, resulting in large improvement over other methods when searching for buildings. However, the same approach achieves worst performance when searching for shoes. The reason is that for shoes which have much wider range of viewpoint variability and more sides than buildings, matching local descriptors precisely between two images is not reliable.

As a second contribution, we propose to use category-specific attributes to handle the large appearance variations present in generic instance search. We assume the category of the query is known, *e.g.*, from the user input. When searching among instances from the same category as the query, attributes outperform existing approaches by a large margin on shoes and cars. The best performances on *CleanShoes* and *Cars* achieved by existing approaches are 36.73 and 23.70, while the learned category-specific attributes achieve 56.56 and 51.11 at the expense of knowing the category of the instance and learning the attributes. For instance search from only one example, it may be reasonable to use more user input. On the building set, *Oxford-Pure*, the category-specific attributes obtain a comparable performance. We conclude the proposed method using automatically learned attributes is more generic than existing approaches.

Finally, we consider searching for an instance in datasets containing instances from various categories. We propose to use the category-level information to address the possible confusion of the query instance with instances from other categories. We show combining category-level information carried by deep learning features and the categorization scores with the category-specific attributes outperforms combining the category information with Fisher vector.

Acknowledgments This research is supported by the Dutch national program COMMIT/.

References

[1] Large scale visual recognition challenge. http://www.imagenet.org/challenges/LSVRC/2010, 2010.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

[3] R. Arandjelovic and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

[4] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[5] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, 2013.

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

[7] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[8] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[11] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2008.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] A. Gordoa, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *CVPR*, 2012.

[14] J. Huang, S. Liu, J. Xing, T. Mei, and S. Yan. Circle & search: Attribute-aware shoe retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):3, 2014.

[15] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *ECCV*, 2012.

[16] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

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

[18] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.

[19] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *CVPR*, 2014.

[20] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *MM*, 2009.

[21] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012.

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

[23] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[24] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *MultiMedia, IEEE*, 13(3):86–91, 2006.

[25] P. Over, G. Awad, J. Fiscus, G. Sanders, and B. Shaw. Trecvid 2012 - an introduction of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2012.

[26] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[27] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010.

[28] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[29] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[31] 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.

[32] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, 2012.

[33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

[34] J. Revaud, M. Douze, and C. Schmid. Correlation-based burstiness for logo retrieval. In *MM*, 2012.

[35] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: theory and practice. *IJCV*, 105(3):222–245, 2013.

[36] V. Sharmanska, N. Quadrianto, and C. H. Lampert. Augmented attribute representations. In *ECCV*, 2012.

[37] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, 2012.

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

[39] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[40] R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders. Locality in generic instance search from one example. In *CVPR*, 2014.

[41] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013.

[42] J. R. R. Uijlings, K. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[43] K. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596, 2010.

[44] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV*, 2011.

[45] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.

[46] F. X. Yu, R. Ji, M.-H. Tsai, G. Ye, and S.-F. Chang. Weak attributes for large-scale image retrieval. In *CVPR*, 2012.

[47] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[48] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian. Semantic-aware co-indexing for image retrieval. In *ICCV*, 2013.