A practical guide to CNNs and Fisher Vectors for image instance retrieval

Vijay Chandrasekhar[*,1], Jie Lin[*,1], Olivier Morère[*,1,2]

Hanlin Goh[1], Antoine Veillard[2]

[1]Institute for Infocomm Research (A*STAR), 1 Fusionolopis Way, #21-01, 138632, Singapore

[2]Université Pierre et Marie Curie, 4 place Jussieu, 75252, Paris, France

Abstract

The computation of good image descriptors is key to the instance retrieval problem and has been the object of much recent interest from the multimedia research community. With deep learning becoming the dominant approach in computer vision, the use of representations extracted from Convolutional Neural Nets (CNNs) is quickly gaining ground on Fisher Vectors (FVs) as favoured state-of-the-art global image descriptors for image instance retrieval. While the good performance of CNNs for image classification are unambiguously recognised, which of the two has the upper hand in the image retrieval context is not entirely clear yet.

In this work, we propose a comprehensive study that systematically evaluates FVs and CNNs for image retrieval. The first part compares the performances of FVs and CNNs on multiple publicly available data sets. We investigate a number of details specific to each method. For FVs, we compare sparse descriptors based on interest point detectors with dense single-scale and multi-scale variants. For CNNs, we focus on understanding the impact of depth, architecture and training data on retrieval results. Our study shows that no descriptor is systematically better than the other and that performance gains can usually be obtained by using both types together. The second part of the study focuses on the impact of geometrical transformations such as rotations and scale changes. FVs based on interest point detectors are intrinsically resilient to such transformations while CNNs do not have a built-in mechanism to ensure such invariance. We show that performance of CNNs can quickly degrade in presence of rotations while they are far less affected by changes in scale. We then propose a number of ways to incorporate the required invariances in the CNN pipeline.

Overall, our work is intended as a reference guide offering practically useful and simply implementable guidelines to anyone looking for state-of-the-art global descriptors best suited to their specific image instance retrieval problem.

[*] V. Chandrasekhar, J. Lin and O. Morère contributed equally to this work.

Index Terms

convolutional neural networks, Fisher vectors, image instance retrieval.

I. INTRODUCTION

Image instance retrieval is the discovery of images from a database representing the same object or scene as the one depicted in a query image. State-of-the-art image instance retrieval pipelines consist of two major blocks: first, a subset of images similar to the query are retrieved from the database, next, geometric consistency checks are applied to select the relevant images from the subset with high precision. The first step is based on the comparison of *global image descriptors*: high-dimensional vectors with up to tens of thousands of dimensions representing the image contents. Better global descriptors are key to improving retrieval performance and has been the object of much recent interest from the multimedia research community with work on specific applications such as digital documents [1], mobile visual search [1], [2], distributed large scale search [3] and compact descriptors for fast real-world applications [4], [5].

A popular global descriptor which achieves high performance is the Fisher Vector (FV) [6]. The FV is obtained by quantizing the set of local feature descriptors with a small codebook of 64-512 centroids, and aggregating first and second order residual statistics for features quantized to each centroid. The residual statistics from each centroid are concatenated together to obtain the high-dimensional global descriptor representation, typically 8192 to 65536 dimensions. The performance increases as the dimensionality of the global descriptor increases, as shown in [6]. FVs can be aggregated on descriptors extracted densely in the image [7], or around interest points like Difference-of-Gaussian (DoG) interest points [8]. The former is popular for image classification tasks, while the latter is used in image retrieval as the DoG interest points provide invariance to scale and rotation.

As opposed to the carefully hand-crafted FVs, deep learning has achieved remarkable performance for large scale image classification [9], [10]. Deep learning has also achieved state-of-the-art results in many other visual tasks such as face recognition [11], [12], pedestrian detection [13] and pose estimation [14]. In their recent work, Babenko et al. [15] propose using representations extracted from Convolutional Neural Nets (CNN) as a global descriptor for image retrieval, and show promising initial results for the approach. In our work, we show how stacked Restricted Boltzmann Machines (RBM) and supervised fine-tuning can be used for generating extremely compact hashes from global descriptors obtained from CNNs for large scale image-retrieval [16].

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS AND KEY FINDINGS

QUESTIONS	OBSERVATIONS AND RECOMMENDATIONS
Best practices for CNN descriptors	
Best single crop strategy?	The largest possible center crop (discarding parts of the image but preserving aspect ratio) or the entire image (preserving the entire image but ignoring aspect ratio) work comparably, both outperforming padding (preserving both).
Best performing layer?	The first fully connected layer is a good all-round choice on all the tested models.
Do deeper networks help?	Only if the training and test data are similar. Else, extra-depth can hurt performance.
How much does training data matter?	Training data has significant impact on performance. Results also suggest that deeper layers are more domain specific.
Best practices for FV interest points	
Dense or sparse interest points?	It depends on the dataset. If scale and rotation invariance are not required, and the data are highly textured, dense sampling outperforms DoG interest points.
Single-scale or multi-scale interest points?	Multi-scale interest points always improve performance.
CNN versus FV	
How do state-of-the-art CNN and FV results compare on standard benchmarks?	It depends on the characteristics of the data set.
Does combining FV and CNN improve performance?	Yes, combining FV with state-of-the-art CNN descriptors can improve retrieval performance often by a significant margin.
Invariance to rotations	
How invariant are CNN features to rotation?	CNN features exhibit very limited invariance to rotation, performance drops rapidly as query rotation angle is varied.
Are CNNs or FVs more invariant to rotation?	FV based on DoG interest points are robust to rotation changes, as would be expected. CNN descriptors are more robust to rotation changes than FV based on dense sampling.
How do we gain rotation invariance for CNN features?	Max-pooling across rotated versions of database images works well, at the loss of some discriminativeness when query and database images are aligned. However, the same max-pooling approach is not effective on dense FVs.
Are deeper CNN layers more invariant to rotation?	The fully connected layers exhibit similar invariance properties to rotation. Visual features (*pool5*) are slightly more robust to small rotation angles but significantly less robust to larger angles.
Invariance to scale changes	
How scale-invariant are CNN features?	CNN descriptors are robust to scale change and work well even for small query scales.
Are CNNs or FVs more scale-invariant?	CNN descriptors are more robust to scale changes than any FV. All FV variants experience a much sharper drop in performance as query scale is decreased compared to CNN features.
How do we gain scale invariance for CNN features?	Similar to rotation invariance, max-pooling across scaled versions of database images works well for gaining scale invariance, at the cost of some discriminativeness.
Are deeper CNN layers more scale-invariant?	Visual features (*pool5*) are more scale-invariant than the deeper fully connected layers.

While deep learning has unquestionably become the dominant approach for image classification, the case for image retrieval has yet to be clearly settled. The two types of descriptors being radically different in nature, one can expect them to behave very differently based on specific aspects of the problem. On one hand, CNNs seem to obtain good retrieval results with more compact starting representation but many factors related to the network architecture or the training may come into play. On the other hand, FVs may be more robust to training data and more invariant to certain geometrical transformations of the images. In fact, some of the best reported instance retrieval performances are still based on hand-crafted features such as FVs [17].

In this work, we perform a thorough investigation of approaches based on FVs and CNNs on multiple publicly available datasets and analyse the pros and cons of each. The first part of the study determines best practices for FVs and CNNs on details specific to each of the approach. For FVs, we investigate the effects of spare SIFT based on interest point detectors versus dense SIFT (single-scale and multi-scale). For CNN descriptors, we specifically study the impacts of image cropping strategies, layer extracted from the CNN, network depth, and training data. Next, we investigate how each type of descriptors performs compared to the other and if a combination of both types of descriptors can improve results over to the best FVs and CNN descriptors. The final part of our work is dedicated to the impact of geometrical transformations such as rotations and scale changes. Unlike FVs based on interest point detectors, CNNs do not have a built-in mechanism to ensure resilience to such transformations. Hence it is necessary to understand how much CNN descriptors are affected by them. We also propose a number of ways to incorporate transformation invariance in the CNN pipeline.

Our work provides a set of straightforward practical guidelines, some valid in general and some problem dependent, one should follow to get the global image descriptors best suited to their specific image instance retrieval task.

II. Related Work

There has been extensive work on the FV and its variants since it was first proposed for instance retrieval. Several improvements to the baseline FV [6] have been proposed in recent literature, including the Residual Enhanced Visual Vector [18] and the Rate-adaptive Compact Fisher Codes (RCFC) [19]. Recent improvements also include better aggregation schemes [20], and better matching kernels [17]. State-of-the-art results using FVs are based on aggregating statistics around interest points like Difference-of-Gaussian [8] or Hessian-affine interest points [21].

CNNs are now considered to be the mainstream approach for large-scale image classification. ImageNet

2014 submissions are all based on CNNs. After the winning submission of Krizhevsky et al. in the ImageNet 2012 challenge [9], CNN began to be applied to the instance retrieval problem as well. There is comparatively less work on CNN-based descriptors for instance retrieval compared to large-scale image classification. Razavian et al. [22] evaluate the performance of CNN model of [9] on a wide range of tasks including instance retrieval, and show initial promising results. Babenko et al. [15] show that fine-tuning a pre-trained CNN with domain specific data (objects, scenes, etc) can improve retrieval performance on relevant data sets. The authors also show that the CNN representations can be compressed more effectively than their Fisher counterparts for large-scale instance retrieval. In [16], we show how sparse high-dimensional CNN representations can be hashed to very compact representations (64-1024 bits) for large scale image retrieval with little loss in matching performance.

While the papers above show initial results, the CNN architecture and features from [9] are used as a black-box for the retrieval task. There is no systematic study of how the CNN architecture and training data affect retrieval performance. Also, unlike interest points which provide scale and rotation invariance to the FV pipeline, CNN representations used in image-classification are obtained by densely sampling a resized canonical image. CNN features do not provide explicit rotation and scale invariance, which are often key to instance retrieval tasks. Desired levels of scale and rotation invariance for CNN features can nevertheless be indirectly achieved from the max-pooling operations in the pipeline, the diversity of the training data which typically contains objects at varying scales and orientations, and data augmentation during the training phase where data can be preprocessed and input to the CNN at different scales and orientations.
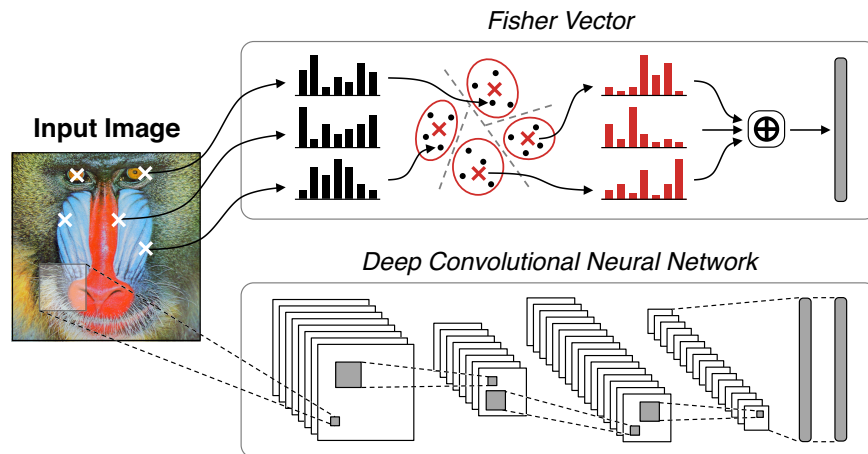

Fig. 1. FV and CNN based pipelines for the computation of global image descriptors.

In this work, we provide a systematic and thorough evaluation of FV and CNN pipelines (see Figure 1) for instance retrieval. We run extensive experiments on 4 popular data sets: *Holidays* [23], *UKBench* [24], *Oxford buildings* [25] and *Stanford Mobile Visual Search* [26] to study how well CNN-based approaches generalize compared with FVs. Our CNN experiments in this work are based on publicly available CNN models in Caffe [27] and can be fully reproduced, unlike CNN models trained by Google, Baidu, Microsoft and Yandex in [28], [29], [15], [30].

III. CONTRIBUTIONS

The main contributions of our work are summarized as follows:

- We provide a comprehensive and systematic evaluation of FVs and CNN descriptors for instance retrieval. Our results based on many standard, publicly available dataset and various pre-trained state-of-the-art models for image classification are fully reproducible.

- We identify the best practices for the use of each type of descriptors through a set of dedicated experiments. For CNNs, we investigate the impacts of the image cropping strategy, the network depth, the layer selected as descriptors, and the training data. For FVs, we study how densely sampled SIFT single-scale and multi-scale descriptors compare with against sparse interest point detectors.

- We compare the best performing FVs and CNN descriptors from our study to various reported state-of-the-art results on the various datasets. We also investigate if a mixture of FVs and CNN descriptors is able to further improve results.

- Unlike FVs based on interest point detectors, CNNs do not have a built-in mechanism to ensure robustness to transformations such as rotations or scale changes. We therefore conduct a set of experiments to compare the performance and robustness of the two types of descriptors when affected by rotations and scale changes. We also propose a number of ways the descriptors could be made more invariant to those transformations.

- The key findings from our study are summarized in Table I intended as a quick reference guide for practical guidelines on the use of FVs and CNNs for image retrieval. The guidelines are sometimes general but often dependent on specific characteristics of the problem which have been properly identified in this study.

IV. EVALUATION FRAMEWORK

A. Data Sets

We evaluate the performances of the descriptors against four popular data sets: *Holidays*, *Oxford buildings (Oxbuild)*, *UKBench* and *Graphics*. The four datasets are chosen for the diversity of data they provide: *UKBench* and *Graphics* are object-centric featuring close-up shots of objects in indoor environments. *Holidays* and *Oxbuild* are scene-centric datasets consisting primarily of outdoor buildings and scenes.

INRIA Holidays. The INRIA Holidays dataset [23] consist of personal holiday pictures. The dataset includes a large variety of outdoor scene types: natural, man-made, water and fire effects. There are 500 queries and 991 database images. Variations in lighting conditions are rare in this data set as the pictures from the same location are taken at the same time.

Oxford Buildings. The Oxford Buildings Dataset [25] consists of 5062 images collected from Flickr representing landmark buildings in Oxford. The collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. Note that the set contains 55 queries only.

UKBench. The University of Kentucky (UKY) data set [24] consists of 2550 groups of common objects. There are 4 images representing each. Only the object of interest present in each image. Thus, there is no foreground or background clutter within this data set. All 10200 images are used as queries.

Graphics. The Graphics data set is part of the Stanford Mobile Visual Search data set [26], which notably was used in the MPEG standard: Compact Descriptors for Visual Search (CDVS) [31]. The data set contains different categories of objects like CDs, DVDs, books, software products, business cards, etc. For product categories (CDs, DVDs and books), at least one of the references is a clean version of the product obtained from the product website. The query images include foreground and background clutter that would be considered typical in real-world scenarii, e.g., a picture of a CD might contain other CDs in the background. This data set distinguishes from the other ones as it contains images of rigid objects captured under widely varying lighting conditions, perspective distortion, foreground and background clutter. Query images are taken with heterogeneous phone cameras. Each query has two relevant images. There are 500 unique objects, 1500 queries, and 1000 database images.

B. Fisher Vectors

FVs are a concatenation of first and second order statistics of a set of feature descriptors quantized with a small codebook. We resize all images (maintaining aspect ratio) so that the larger dimension of

the image is equal to 640 pixels prior to FV extraction. We use the implementation of FVs from the open source library VLFeat [32]. SIFT detectors and descriptors are also chosen from the same library. The three different types of SIFT descriptors used to generate the FVs are Difference of Gaussians (DoG) SIFT, Dense Single-scale SIFT and Dense Multi-scale SIFT.

- **DoG SIFT**. We detect interest points in the DoG scale space, followed by 128-dimensional SIFT descriptors extracted from scaled and oriented patches centered on interest points. Default peak and edge thresholds (0 and 10) are employed to filter out low contrast patches or patches close to the edge of the image. Since the DoG detector extracts scale and rotation invariant interest points, it has been widely applied for the task of instance retrieval. It is important to note that we do not use any feature selection algorithm to select a subset of "good" features - an approach that can result in a significant improvement in performance on the *Graphics* data set [33].

- **Dense Single-scale SIFT**. We extract SIFT descriptors from densely sampled patches (every 4 pixels) with fixed scale and upright orientation. The patch size used for the extraction is $m \times s$ where s is the scale parameter and m is the magnification parameter. We choose the default magnification parameter $m = 6$ across all dense SIFT descriptors. $s = 4$ is chosen for single-scale SIFT. Dense SIFT is faster to compute than DoG SIFT as the expensive interest point detection step is avoided - however, this comes at the cost of lower scale and rotation invariance. Note that dense SIFT is mostly popular for image classification tasks.

- **Dense Multi-Scale SIFT**. We apply dense SIFT extraction at multiple resolutions ($s = \{4, 8, 12, 16\}$). This is aimed at gaining some degree of scale invariance.

Closely following [34][6], we apply dimensionality reduction on SIFT descriptors from 128 to 64 using PCA, and train a Gaussian Mixture Model (GMM) with 256 centroids. Both first order (gradients w.r.t. mean) and second order (gradients w.r.t. variance) statistics are encoded to form the FV, resulting in a $64 \times 256 \times 2 = 32768$-dimensional vector representation for each image. Finally, we apply power law normalization to each component ($\alpha = 0.5$), followed by L_2 normalization to obtain the final normalized FV representation [6]. Each dimension of the FV is stored as a floating point number. No compression is applied. We refer to the three FV as FVDoG (FV computed on DoG points), FVDS (FV computed densely at a single scale) and FVDM (FV computed densely at multiple scales) from here on.

C. Convolutional Neural Net features

In this work, we consider four different pre-trained CNN models for the instance retrieval problem:

- *OxfordNet* [35]: the best performing single network from the Oxford VGG team at ImageNet 2014.

- *AlexNet* [9]: the model referenced as "BVLC reference caffenet" in the Caffe framework [27]. This model was the winning ImageNet submission of 2012. This network closely mimics the original *AlexNet* model of [9].

- *PlacesNet* [36]: a state-of-the-art model for scene image classification providing highest accuracy on the SUN397 dataset [37].

- *HybridNet* [36]: another model for both object and scene images classification, outperforming state-of-the-art methods on the MIT Indoor67 dataset [38].

Details on the architecture, training set and layer sizes of the CNNs are summarized in Table II.

TABLE II
DETAILS ON ARCHITECTURE, TRAINING SET AND LAYER SIZE OF THE CNNS.

	ARCHITECTURE			TRAINING			LAYER SIZE			
	parameters	depth (conv+fc)	input size	training set	classes	data size	pool5	fc6	fc7	fc8
OxfordNet	138M	13+3	$224 \times 224 \times 3$	ImageNet	1000	1.2M	$7 \times 7 \times 512$	4096	4096	1000
AlexNet	60M	5+3	$227 \times 227 \times 3$	ImageNet	1000	1.2M	$6 \times 6 \times 256$	4096	4096	1000
PlacesNet	60M	5+3	$227 \times 227 \times 3$	Places-205	205	2.4M	$6 \times 6 \times 256$	4096	4096	205
HybridNet	60M	5+3	$227 \times 227 \times 3$	Both	1183	3.6M	$6 \times 6 \times 256$	4096	4096	1183

These state-of-the-art models are chosen as they allow us to run interesting control experiments, where the CNN architecture or training data are varied. *PlacesNet* and *HybridNet* share the same architecture as *AlexNet* [36], while being trained on different data. *OxfordNet* and *AlexNet* are trained on the same data, but have different architectures: compared to *AlexNet*, *OxfordNet* is deeper, has twice as many layers, twice the number of parameters, and achieved better image classification performance in the ImageNet 2014 contest [35].

The 4 models are trained differently, using the ImageNet [39] and Places-205 [36] datasets. With categories like "Amphitheater", "Jail cell" or "Roof garden", Places-205 is a scene-centric dataset, while ImageNet, featuring categories such as "Vending machine", "Barn spider" or "Chocolate syrup", is more object-centric. Places-205 is twice as large as ImageNet, but has 5 times fewer classes. *OxfordNet* and *AlexNet* are trained on ImageNet. *HybridNet* is trained on a combination of ImageNet and Places-205 data: the resulting dataset being 3 times larger than ImageNet alone, and having a larger variety of classes.

Given an input image, we first resize it to a canonical resolution, compute the feed-forward neural network activations, and extract the last four layers for each CNN model. We refer by $pool5$, $fc6$, $fc7$ and $fc8$ outputs of the last 4 layers of each network (as denoted in Caffe). $pool5$ is the output of the last convolutional layer after pooling, and $fc6$, $fc7$, $fc8$ are outputs of the fully connected layers. $pool5$ still

contains spatial information from the input image. The size of the last layer $fc8$ is equal to the number of classes. All descriptors are extracted after applying the rectified linear transform, and L_2 normalized: the features are directly output from the Caffe implementation of the CNN models.
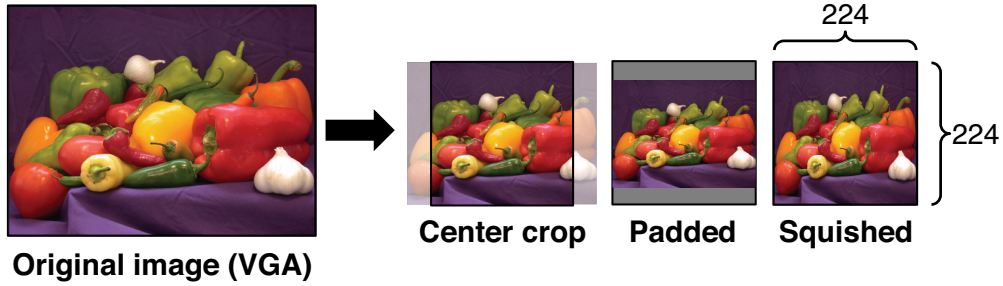
V. EXPERIMENTAL RESULTS

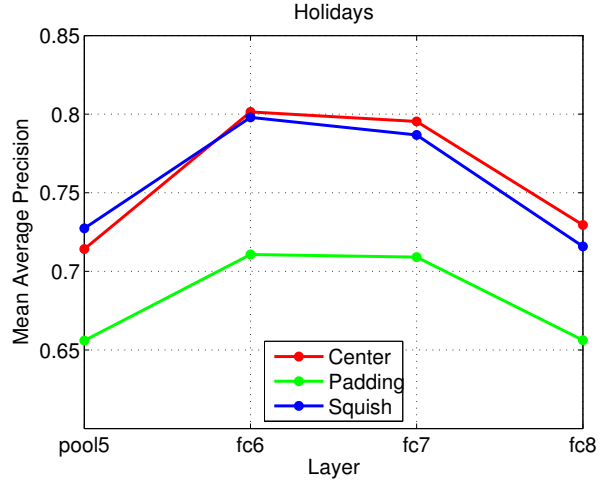Fig. 2. Different single-crop strategies used for input into CNN pipelines.

Fig. 3. MAP for different layers of *OxfordNet*, for different single-crop strategies on the *Holidays* data set. We observe that *Center* crop and *Squish* perform comparably.

A. Best practices for CNN descriptors

What single crop strategy is the best?

CNN pipelines take input images at a fixed resolution (see Table II). We wish to determine which single-crop strategy works best in the context of instance retrieval where images may vary in size and

aspect ratio. We consider the following 3 different cropping strategies illustrated in Figure 2. Numerical values are given to fit *OxfordNet*.

- *Center*: the largest 224 × 224 center crop, after rescaling the image to 224 pixels for the smaller dimension, while maintaining aspect ratio.
- *Padding*: the original image is resized to 224 pixels for the larger dimension, maintaining aspect ratio, and any unfilled pixels are padded with a constant value equal to the training set mean.
- *Squish*: the original image is resized to 224 × 224. The original aspect ratio is ignored potentially resulting in distortions.

In Figure 3, we plot MAP for different layers of *OxfordNet*, for the *Holidays* data set. We note that *Center* and *Squish* perform comparably, outperforming *Padding*. The trend is consistent across the different network layers. We observe similar results for other data sets and CNN models. Most data sets in this study have a center bias for the object of interest, explaining the best performances of the *Center* cropping strategy.

A small improvement in performance for large-scale image classification is obtained by averaging output class probabilities computed over several cropped regions within an image, often extracted at different positions and scales [10]. Such a performance improvement can also be achieved for instance-retrieval by pooling CNN results over several cropped regions, but such a strategy could be applied to other global descriptor pipelines too. For the remaining experiments in this paper, we consider a single *Center* crop for processing all database and query images.

Which CNN layer performs the best?

In Figure 4, we plot MAP for the last 4 layers of *OxfordNet*, *AlexNet*, *PlacesNet* and *HybridNet* for different data sets. We note that for each network, intermediate layers perform best for instance retrieval. Such a sweet spot is intuitive as the final layer represents higher level semantic concepts, while intermediate convolutional and fully connected layers provide rich representations of low level image information. We note that layer $fc6$ performs the best for all CNN, for all data sets except *Graphics*. For *Graphics*, performance drops with increase in depth, as all four CNN models are learnt on natural image statistics, while the *Graphics* data set is biased towards data like CD covers, DVD covers, business cards, and dense text in newspaper articles.

How much improvement can we obtain by deeper CNN architectures?

We compare *OxfordNet* and *AlexNet* results in Figure 4. *OxfordNet* and *AlexNet* are both trained on the same 1.2 million images from the ImageNet data set, but vary in the number of layers: 16 and 8 layers respectively. We note that *OxfordNet* outperforms *AlexNet* on all data sets, except *Graphics*. On
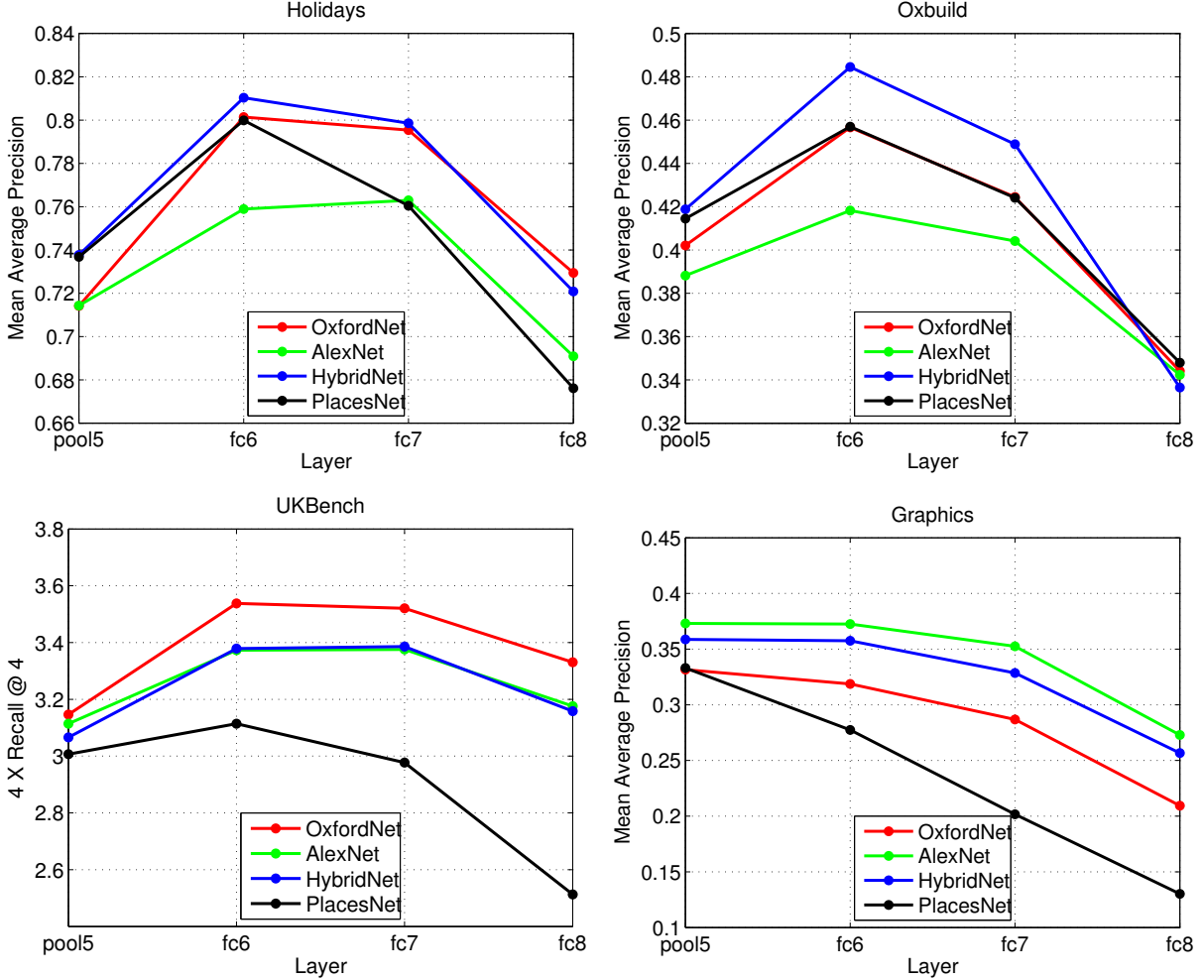
Fig. 4. MAP for the last 4 layers of state-of-the-art publicly available CNN. *OxfordNet* and *AlexNet* are trained on the same data, while *PlacesNet*, *HybridNet* and *AlexNet* have the same network architecture but are trained on different data. We note that performance improves by using deeper networks, and by training on domain specific data, but only if training and testing data have similar characteristics.

Graphics, the performance of *OxfordNet* is worse, strongly suggesting that performance improves with more layers as long as the training data is representative of the test data set in consideration.

How much improvement can we obtain by training CNN models using domain specific data?

For this experiment, we compare *AlexNet* with *PlacesNet* in Figures 4. *AlexNet*, and *PlacesNet* use the same 8-layer CNN architecture, but are trained on different data. We observe that *PlacesNet* outperforms *AlexNet* on *Holidays* and *Oxbuild* data sets. This shows that using training data more representative of the test data can improve performance significantly, as *Holidays* and *Oxbuild* are scene-centric. On the object-centric *UKBench* and *Graphics* data sets, *PlacesNet* performs worse than *AlexNet* due to the mismatch between training and test data.

Further, in Table III, we compare our results to the CNN retrieval results presented in [15]. In [15], Babenko et al. fine-tune a pre-trained *AlexNet* model based on ImageNet training data with domain specific images, e.g., landmarks and objects. As shown in Table III, the authors are able to improve retrieval performance over the *AlexNet* baseline model, on *Holidays* and *UKBench* by fine-tuning with landmark and object data respectively. However, the resulting trade-off is a loss in performance on *Holidays* when fine-tuning with object data and vice-versa.

We compare *OxfordNet* results with those of the fine-tuned models of [15]. We note that the deeper architecture of *OxfordNet* trained on just ImageNet data results in comparable or higher performance than the fine-tuned models on both *Holidays* and *UKBench*, suggesting that there is more gain to be had with deeper networks rather than fine-tuning a shallower network with domain-specific data.

Together with the previous sets of experiments, there is strong combined evidence that deeper layers/models have the potential of achieving higher discriminativeness on domain specific data at the expense of less generalisability on non-specific data.

How much improvement can we obtain by training CNN on larger and more diverse data?

For this experiment, we compare *AlexNet* with *HybridNet* in Figure 4. *AlexNet*, and *HybridNet* use the same 8-layer CNN architecture, but the latter is trained on a combination of ImageNet and Places-205, resulting in a larger training data set with more diverse classes. We note that *HybridNet* performs comparably or better than *AlexNet* on all data sets except *Graphics*, suggesting that increasing the amount and diversity of training data is equally important as increasing depth in the CNN architecture.

TABLE III

STATE-OF-THE-ART CNN AND FV RESULTS FOR INSTANCE RETRIEVAL. 4 × RECALL @ 4 FOR *UKBench*, AND MAP FOR OTHER DATA SETS.

Descriptor	Dim	Holidays	UKBench	Oxbuild	Graphics
OxfordNet	4096	0.80	3.54	0.46	0.33
AlexNet	4096	0.76	3.38	0.42	0.37
HybridNet	4096	0.81	3.39	0.48	0.36
PlacesNet	4096	0.80	3.11	0.46	0.33
CNN (Fine-tuned on Landmarks) [15]	4096	0.793	3.29	0.545	
CNN (Fine-tuned on Objects) [15]	4096	0.754	3.56	0.393	
FVDoG	32768	0.63	2.8	0.42	0.66
FVDS	32768	0.73	2.38	0.51	0.20
FVDM	32768	0.75	2.45	0.55	0.32

B. Best practices for FV interest points

CNN features are obtained by dense sampling over the image. In Table III, we study if such an approach is also effective for FVs. We compare the performance of FVDoG, FVDS and FVDM as described in Section IV.

We note that dense sampling (FVDS and FVDM) improves performance over FVDoG on *Holidays* and *Oxbuild* data sets, while hurting performance on *Graphics* and *UKBench*. Note the large drop in performance of dense sampling on the *Graphics* data set. This is intuitive as queries in the *Graphics* data set contain query objects at different scales and rotations. For *Holidays* and *Oxbuild* data sets, even FVDS improves performance over FVDoG, suggesting that most query and database image pairs occur at roughly the same scale.

Dense sampling is effective for data sets like *Holidays* which consist primarily of outdoor scenes, and are mainly composed of highly textured patches. The improvement in performance of dense sampling approaches can also be attributed to the discriminativeness-invariance tradeoff. Where retrieval does not require scale and rotation invariance, and data are highly textured over the entire image, performance can be improved by dense sampling.

Sampling at multiple scales also seems to consistently improve results over single scale sampling for dense descriptors.

C. Comparisons to state-of-the-art

Does combining FV and CNN improve performance?

In Figure 5, we present retrieval results obtained by combining FVDoG, FVDS, and FVDM individually with *OxfordNet fc6* features. We employ a simple early fusion approach where the FV and CNN features are concatenated after weighting by α and $(1-\alpha)$ respectively. $\alpha = 0$ corresponds to using just FVDoG, FVDS or FVDM features individually, while $\alpha = 1$ corresponds to just the *OxfordNet* feature. This early fusion scheme is also equivalent to weighting the squared L_2 distance measure for matching by α and $1 - \alpha$ for FV and CNN features respectively.

All four data sets show an improvement in peak performance by combining FV and CNN features. The maximum performance is achieved for $\alpha = 0.4$ for the *Holidays*, *UKBench* and *Oxbuild* data sets, and $\alpha = 0.3$ for the *Graphics* data set, using different FV. There is a significant improvement in performance by combining FV and CNN features on all data sets except *Graphics*. The results suggest that a simple hyperparameter can be used to combine FV and CNN across data sets with similar characteristics. Also,

$\alpha = 0.4$ suggests that FV contribute significantly to achieving high retrieval performance (in contrast, an α parameter close to 1 would suggest that most of the contribution is from the CNN feature).

Note that our goal here is to show that performance can be improved significantly by combining FV and CNN features, and not necessarily to achieve highest performance on these retrieval benchmarks. Peak performance presented in Figure 5 can be improved by (a) database-side rotation and scale pooling which helps significantly (see Sections V-D and V-E) (b) better CNN models than *OxfordNet* on individual data sets (see Table III) (c) better FV based on Hessian Affine interest points [21] instead of DoG points used in this paper [32] (d) better FV with more sophisticated aggregation techniques [20], and (e) combining all FV and CNN descriptors together. (f) using more sophisticated fusion and ranking techniques for combining results, like the one proposed in the recent paper [40].

TABLE IV
STATE-OF-THE-ART RESULTS. MAP FOR *Oxbuild* AND *Holidays*, AND $4\times$ RECALL @4 FOR *UKBench*

Descriptor	Dim	Holidays	UKBench	Oxbuild
Bag-of-words 1M [24]	1M		3.19	
VLAD baseline [41]	8192	0.526	3.17	
Fisher baseline [41]	8192	0.495	3.09	
Fisher baseline (ours)	32768	0.63	2.8	0.42
Fisher ADC (320 bytes) [34]	2048	0.634	3.47	
Fisher+color [42]	4096	0.774	3.19	
VLAD++ [43]	32768	0.646		0.555
Sparse-coded features [44]	11024	0.767	**3.76**	
Triangulation Embed [20]	8064	0.77		**0.676**
Triangulation Embed [20]	1920		3.53	
Best CNN results from this paper across all CNN	4096	0.81	3.54	0.48
Fusion of *OxfordNet* and Baseline FV	32768+ 4096	**0.85**	3.71	0.59

Comparisons to state-of-the-art.

In Table IV, we compare state-of-the-art results reported on *Holidays*, *UKBench* and *Oxbuild*. We include a wide range of approaches starting from Bag-of-words [24] to latest FV aggregation methods [20]. We include the best CNN and fusion results reported in this paper. We note that the best CNN results (based on pre-trained models considered in this work) achieve higher performance than state-of-the-art FV approaches [20] on *Holidays* and *UKBench* data sets. There is a gap in performance between CNN results reported in this work and state-of-the-art FV for *Oxbuild*: however, *Oxbuild* is a much smaller data set with only 55 queries. Finally, we note that the simple fusion technique in Figure 5 results in highest or one of the highest performance numbers reported on each data set. Peak performance numbers
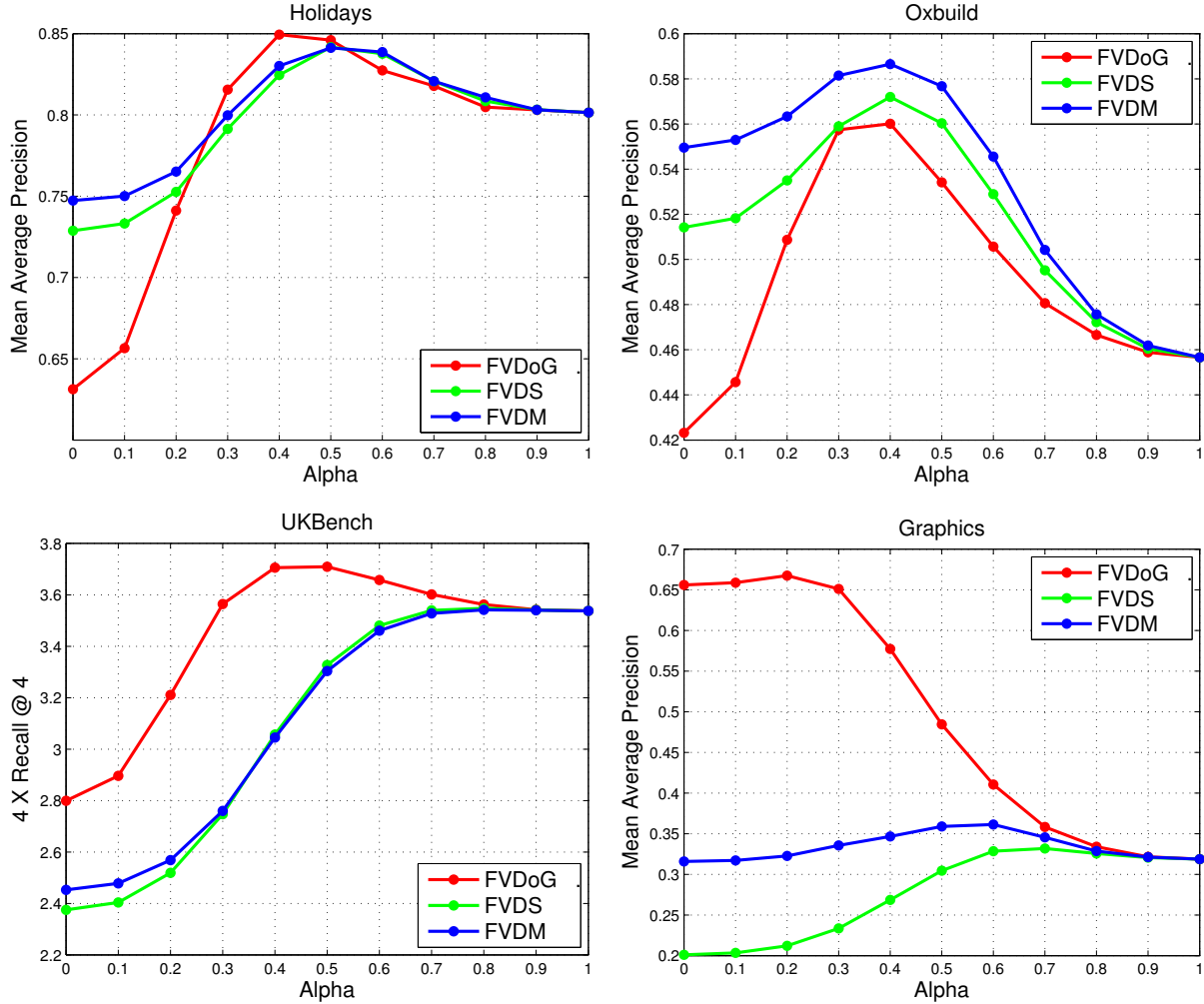
Fig. 5. Combining different FVs with *OxfordNet fc6* with early fusion. FV and *OxfordNet* features are concatenated with weights α and $1 - \alpha$ respectively. $\alpha = 0$ refers to just using FVDoG, FVDS or FVDM above, while $\alpha = 1$ refers to just the *OxfordNet fc6* feature. We observe that retrieval performance improves on all data sets by combining FV and CNN.

for the fusion approach can be improved using approaches (a)-(f) described above.

D. Invariance to Rotation

How invariant are CNN features to rotation?

CNN features, unlike FVDoG, have limited levels of rotation invariance. The invariance arises from the max-pooling steps in the CNN pipeline, and rotated versions of objects present in the training data. In Figure 7, we rotate each query at different angles and measure MAP for the *Holidays* data set for different layers of *OxfordNet*. For these control experiments, query images are cropped circularly in the center (to avoid edge artifacts) and rotated in steps of 10°. The same experimental set up is also
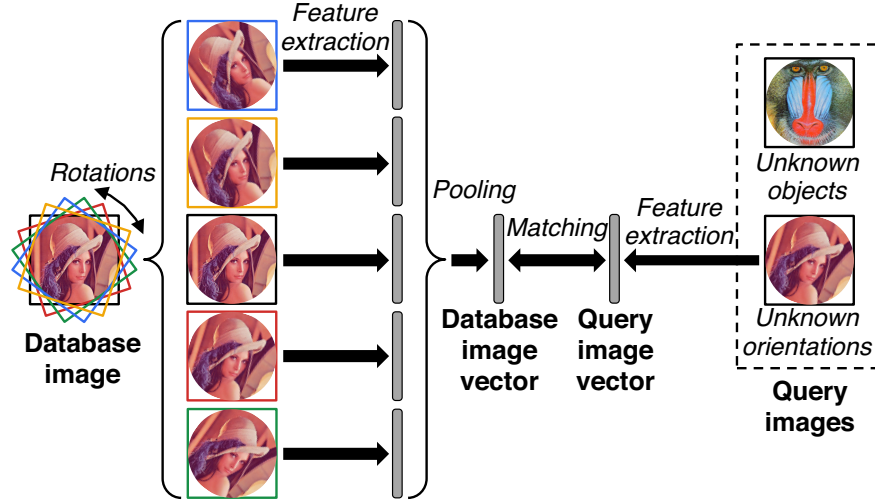
Fig. 6. We extract features from database images at different rotations and pool them to obtain a single representation. We rotate queries and evaluate retrieval performance for different pooling parameters and strategies. All database and query images are cropped circularly at the center to avoid edge artifacts for this experiment, and padded with a default mean RGB value (ImageNet mean).
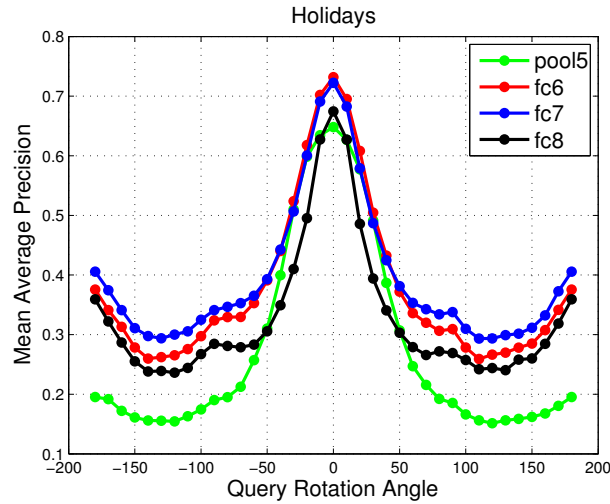

Fig. 7. MAP as query images are rotated for different layers of *OxfordNet*. We note that CNN features have very limited rotation invariance, with performance dropping steeply for all layers of the network beyond $10°$.

employed in evaluation of rotation invariant features in [45]. We note that CNN features have very limited rotation invariance with performance dropping steeply beyond $10°$. Furthermore, the different layers of the network exhibit similar characteristics suggesting that rotation invariance does not increase with depth in the CNN.

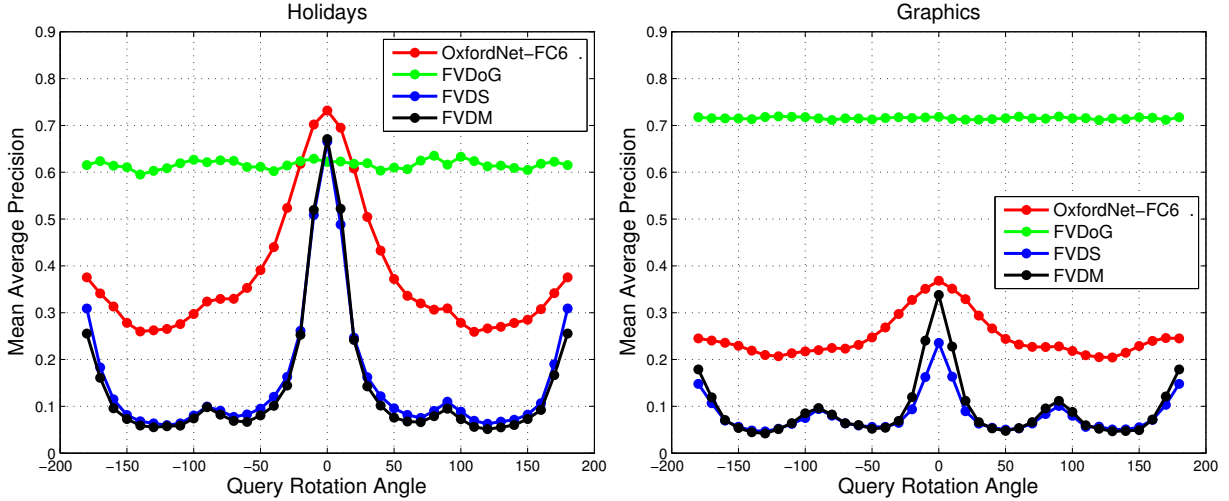**Are FVs or CNNs more rotation invariant?**

Fig. 8. Comparison of *OxfordNet fc6* and FV for rotated queries on the *Holidays* and *Graphics* data sets. The FVDoG is robust to rotation, while *OxfordNet*, FVDS, FVDM suffer a sharp drop in performance.

For the sake of evaluation, we choose one scene-centric and one object-centric data set that are most different: *Holidays* and *Graphics*. In Figure 8, we compare the performance of FV variants and *OxfordNet fc6* as queries are rotated at different angles. Note that FVDoG is robust to rotation - the minor modulation in performance is due to filtering artifacts in the DoG interest point detector. However, the *OxfordNet* features, FVDS and FVDM have a steep drop in performance as queries are rotated. The *OxfordNet* features are more rotation invariant than FVDS and FVDM. Finally, note the large gap in performance between FVDoG and other schemes for the *Graphics* data set. The gap in performance on *Graphics* arises from two contributing factors: (a) the worse performance of the *OxfordNet* features on this data set, and (b) the fact that there are several rotated queries on which the *OxfordNet* features perform worse. The effect of each can be isolated from the next set of experiments we conduct. Next, we discuss how to gain invariance for the CNN, FVDS and FVDM pipelines Ideally, we desire invariance to rotation, while maintaining high discriminability.

How do we gain rotation invariance for CNN features?

We propose a database pooling scheme illustrated in Figure 6 for gaining rotation invariance. Each database image is rotated within a range of $-p°$ to $p°$, in steps of $10°$. The CNN features for each rotated database image are pooled together into one common global descriptor representation. In Figure 9, we present results for max-pooling, where we store the component-wise maximum value across all rotated representations in an angular range. $P = 0$ refers to no pooling, while $P = p$ refers to pooling in the range of $-p°$ to $p°$ in steps of $s = 10°$. The parameter s indicates the quantization step size of angular
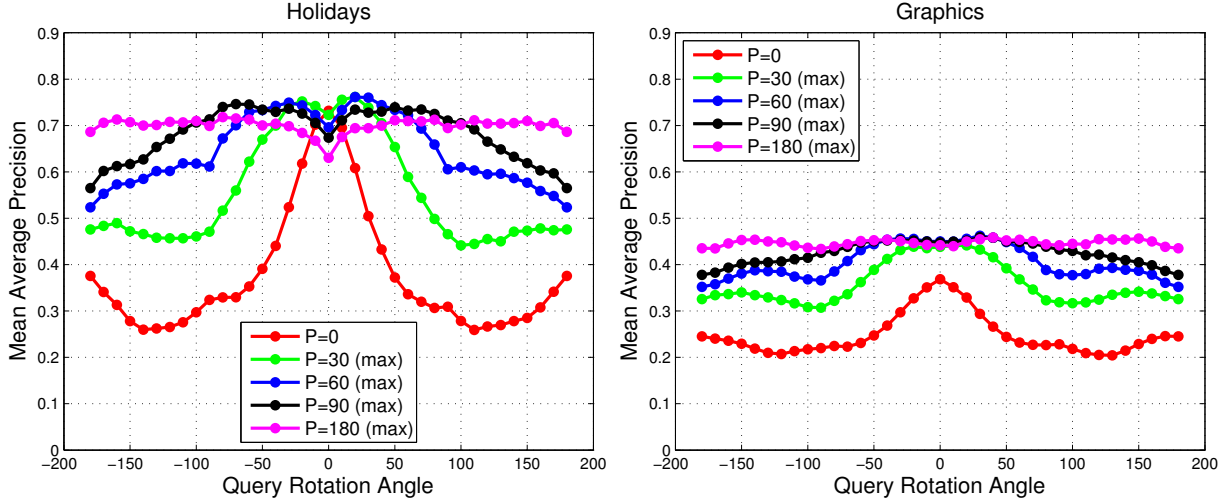
Fig. 9. MAP vs query rotation angle for different pooling parameters. Results are presented on *OxfordNet* $fc6$ layer on the *Holidays* data set. $P = 0$ refers to no pooling. $P = p$ refers to max-pooling over individual feature dimensions, for rotations between $-p^\circ$ and p° in steps of $s = 10^\circ$. Invariance to rotation increases with increasing p, at the expense of lower performance at angle 0°.

rotation of database images.

We plot performance as query rotation angle is varied for varying pooling parameter P, on *OxfordNet* *fc6* layer for the *Holidays* and *Graphics* data sets. The invariance-discriminativeness trade-off is shown in Figure 9. We observe that the max pooling scheme performs surprisingly well for gaining rotation invariance. As P is increased, the performance curve flattens in the range of $-P^\circ$ to P°, at the expense of lower performance for upright queries, i.e, at angle 0. For the *Holidays* data set, most database and query images share similar "upright" orientations. For the *Graphics* data set, note the gap in performance of different schemes at angle 0, between no pooling and different pooling schemes. This gap in performance can be attributed to rotated objects in the query data set. The remaining gap at original angle 0° between FVDoG in Figure 8 and CNN features in Figure 9 can be attributed to the worse performance of the *OxfordNet* features for this data set.

To further understand the effectiveness of database-side pooling, we evaluate different types of pooling methods and database augmentation techniques in Figure 10. We show results for component-wise max pooling and average pooling over rotated database images for different pooling parameters P. We note that max pooling and average pooling perform comparably for small P. For $P = 180^\circ$, we note that average pooling outperforms max pooling.

We compare the two pooling strategies to a simple database augmentation technique labeled *Min-dist*, which stores descriptors for each rotated version of the database image. For *Min-dist*, at query time,
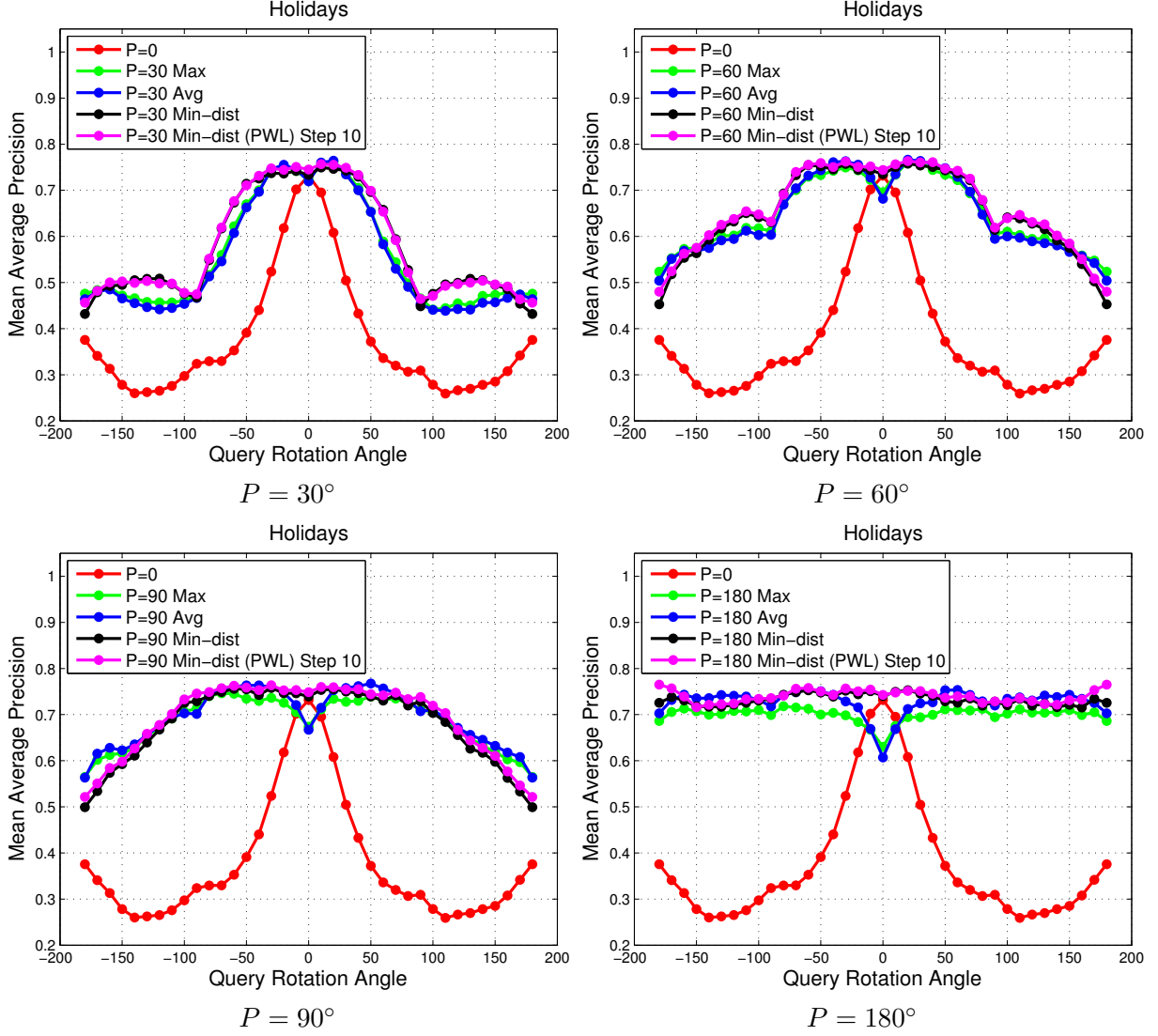
$P = 30°$

$P = 60°$

$P = 90°$

$P = 180°$

Fig. 10. Comparison of different types of database pooling and augmentation techniques, for varying pooling parameter P. *OxfordNet* layer $fc6$ is used on the *Holidays* data set. We notice that max and average pooling come close to the perform of *Min-dist* and *Min-dist(PWL)*, which require storage of multiple feature descriptors ($\frac{2P}{s} + 1$) for each database image, where $s = 10$ is the quantization step size in degrees.

we compute the minimum distance to all the rotated versions for each database image. The *Min-dist* increases the size of the database by $\frac{2P}{s} + 1$, where $s = 10$ is the step size in degrees, and P is the pooling parameter. For small $s \to 0$, the *Min-dist* scheme provides an approximate upper bound on the performance that the max and average pooling schemes can achieve, as a descriptor for each rotated version is explicitly stored in the database. We observe that both max and average pooling are surprisingly effective, as their performance comes close to that of the *Min-dist* scheme while storing only
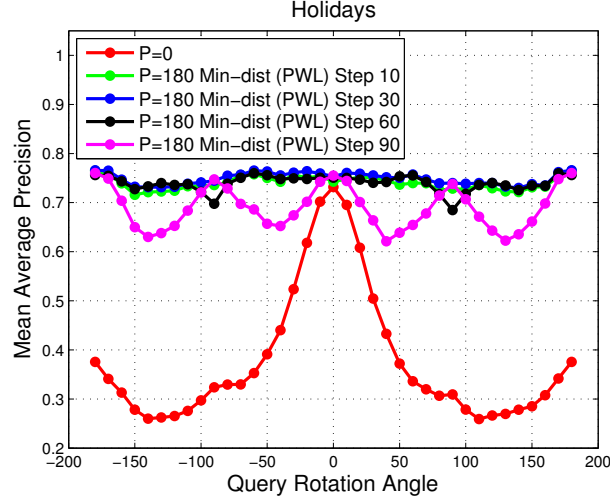
Fig. 11. Results of the *Min-dist (PWL)* scheme as step size parameter s is varied. *OxfordNet* layer $fc6$ is used on the *Holidays* data set. A piece-wise linear approximation of the manifold, on which rotated descriptors of each image lie, is used to trade-off performance and matching complexity. The performance of step size $s = 60°$ is close to that of $s = 10°$, while reducing memory requirements by $6\times$.

one descriptor per database image. Note that the *Min-dist* scheme performs the best, as there is no drop in performance at $0°$, compared to the pooling methods.

Next, we also propose a scheme illustrated in Figure 12 for reducing memory requirements of the *Min-dist* scheme at the expense of increased matching complexity. The scheme labeled *Min-dist (PWL)* assumes a piece-wise linear approximation of the manifold on which the descriptors of each rotated image lie, and computes the closest distance to the manifold. The results for the *Min-dist (PWL)* with step size $10°$ are shown in Figure 10, and it performs comparably to the *Min-dist* scheme. Instead of maintaining database descriptors at finely quantized angular rotations of $s = 10°$, we increase s to $30°, 60°, 90°$ and present results in Figure 11. We note that the performance of step size $s = 60°$ is close to that of $s = 10°$ for *Min-dist (PWL)*, while reducing memory requirements by $6\times$. The drop in performance for the *Min-dist (PWL)* scheme at -135°,-45°,45°,145° for $s = 90°$ shows inherent data set bias at these angles. In conclusion, the proposed simple but elegant *Min-dist (PWL)* scheme helps gain rotation invariance, while requiring storage of fewer descriptors compared to the *Min-dist* approach.

The surprising effectiveness of max and average pooling to gain rotation invariance for CNN features led us to run the same set of experiments on FVDM. We present the results for max pooling, average pooling, *Min-dist*, and *Min-dist (PWL)* with step size $s = 10°$ in Figure 13 for $P = 180°$. Average pooling on FVDM helps gain invariance to rotation while lowering peak performance achieved without
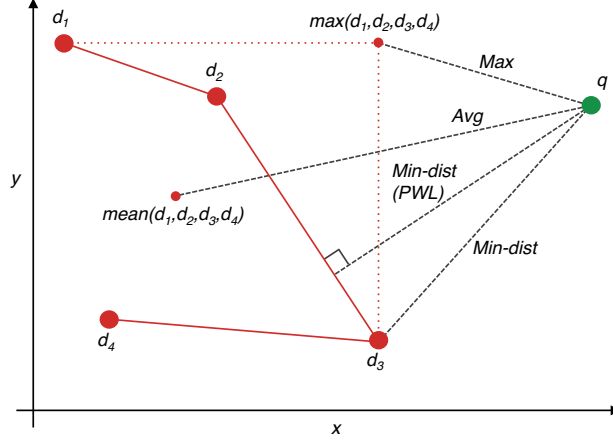
Fig. 12. Illustration of the different pooling schemes. The scheme labeled *Min-dist (PWL)* assumes a piece-wise linear approximation of the manifold on which the descriptors of each rotated image lie, and computes the closest distance to the manifold.

pooling ($P = 0$). However, note the large difference in performance between max and average pooling for FVDM. CNN features are sparse with a small number of dimensions with high values: spikes resulting from the activation of neurons in the network. FVDM data are comparatively more dense. As a result, max pooling on *OxfordNet* features is far more effective than for FVDM. Finally, in Figure 13, *Min-dist* and *Min-dist (PWL)* perform the best, as also observed for *OxfordNet* features.
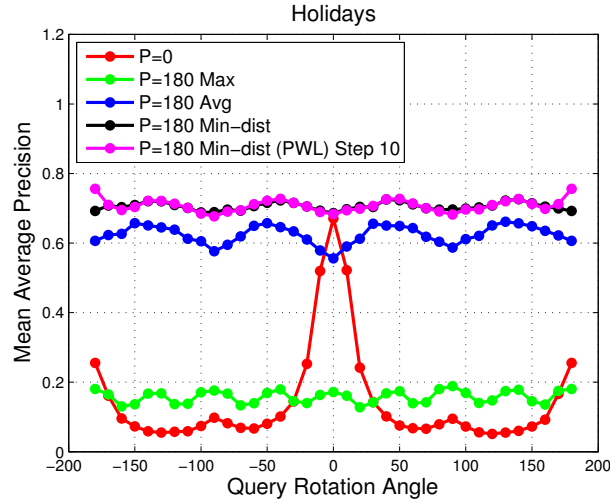

Fig. 13. Comparison of different types of database pooling and augmentation techniques for pooling parameter $P = 180$ for FVDM on the *Holidays* data set. Note the difference in performance of max and average pooling for FVDM, compared to max and average pooling on *OxfordNet* features in Figure 10(d).

E. Invariance to Scale Changes

In this section, we study scale invariance properties of CNNs and FVs. Similar to rotation experiments in Section V-D, we carry out control experiments on the *Holidays* data where we reduce the scale of query images and measure retrieval performance. The starting resolution of all images (database and queries) is set to the larger dimension of 640 pixels (maintaining aspect ratio), as discussed in Section IV.

Both CNN and FV pipelines take in input images at fixed resolution. For FVDoG, FVDS, FVDM pipelines, we resize images to VGA resolution (preserving aspect ratio) before feature extraction, even when the input resolution is smaller. Upsampling images before feature extraction is shown to improve matching performance [8]. The size of center crop input images (after resizing) to the *OxfordNet* pipeline is specified in Table II.

How invariant are CNN features to scale?

We scale query images along both image dimensions by a ratio of 0.75, 0.5, 0.375, 0.25, 0.2 and 0.125 starting from the VGA resolution - the smallest queries are $\left(\frac{1}{8}\right)^{th}$ the size of the VGA resolution image. An anti-aliasing Gaussian filter is applied, followed by bicubic interpolation in the downsampling operation. We present MAP for different layers of *OxfordNet* in Figure 14. We note that the $fc6$ layer experiences only a small drop in performance up to scale 0.25 before steeply dropping off. The *OxfordNet* features are learnt on input images of fairly low resolution, which explains the robustness to large changes in scale. We also note that the three fully connected layers exhibit similar characteristics for scale change: deeper fully connected layers are not more scale invariant. It is interesting to note that $pool5$ is more scale invariant, as seen from the more gradual drop in performance as query scale is decreased: however, $pool5$ is less discriminative with a significant performance gap for smaller scale changes (0.75 to 0.25).

Are CNN or FV more scale invariant?

Similar to the rotation experiment, we compare performance of *OxfordNet* $fc6$ and FVs in Figure 15 for the *Holidays* and *Graphics* data sets. We observe a steeper drop in performance with decreasing scale for FVDM and FVDS compared to *OxfordNet*. Somewhat surprisingly, FVDoG also experiences a sharper drop in performance compared to CNN. The trends are consistent across data sets: the only difference is that the peak performance of FVDoG is higher than CNN on *Graphics*. Trends similar to *Holidays* are observed on the remaining two data sets. The sharp drop in performance of FVDoG can be attributed to the failure of the interest point detector at small scales. CNNs are learnt on smaller images to begin with, and objects shown at different scales at training time, are sufficient for achieving more scale invariance than FVDoG. In comparison to the rotation experiments, it is interesting to note that
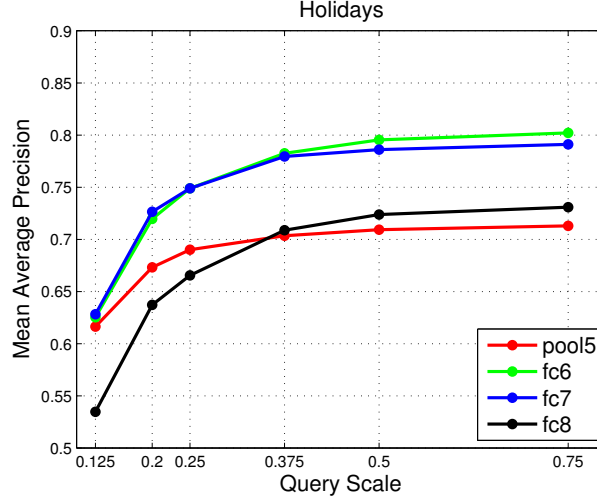
Fig. 14. MAP as query images are scaled to 0.125 of original resolution, for different layers of *OxfordNet* on the *Holidays* data set. We note that *OxfordNet* features are robust to scale change up to 0.25, with performance dropping steeply after.

FVDoG are more robust to rotation changes, while *OxfordNet* features are more robust to scale changes.
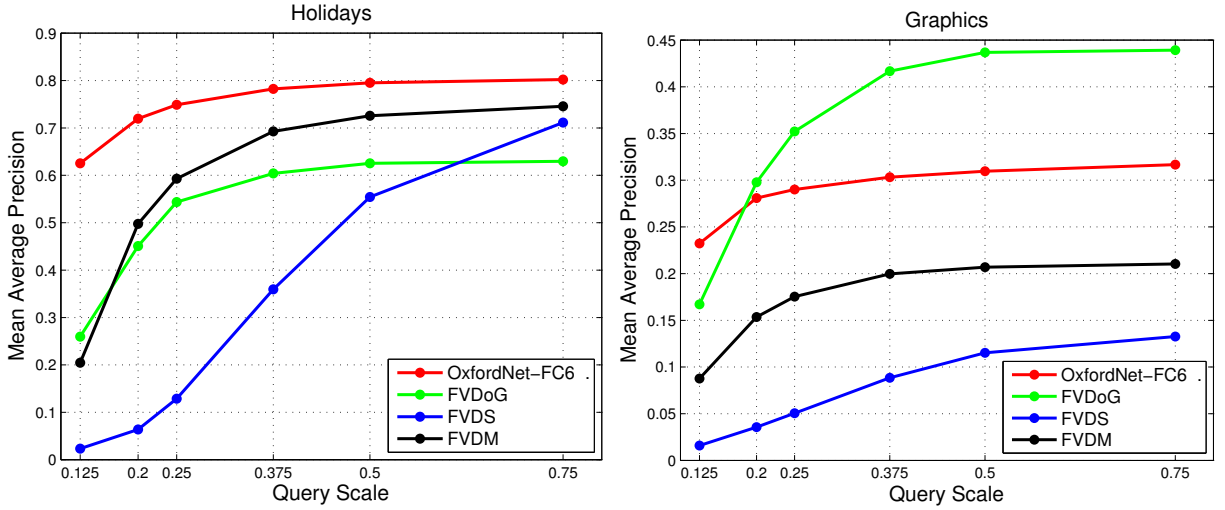

Fig. 15. Comparison of *OxfordNet* $fc6$ and FVs for scaled queries on the *Holidays* and *Graphics* data sets. We observe that *OxfordNet* features are more robust to scale changes compared to FVDoG, FVDS and FVDM, all of which experience a steeper drop in performance as query scale is decreased.

Does database-side pooling improve scale invariance for CNN features ?

Next, we discuss how performance at small scales can be improved by pooling descriptors on the database side. As illustrated in Figure 16, the component-wise pooling operation across scales is similar to the database-pooling performed on rotated images. The parameter SP refers to the number of scales
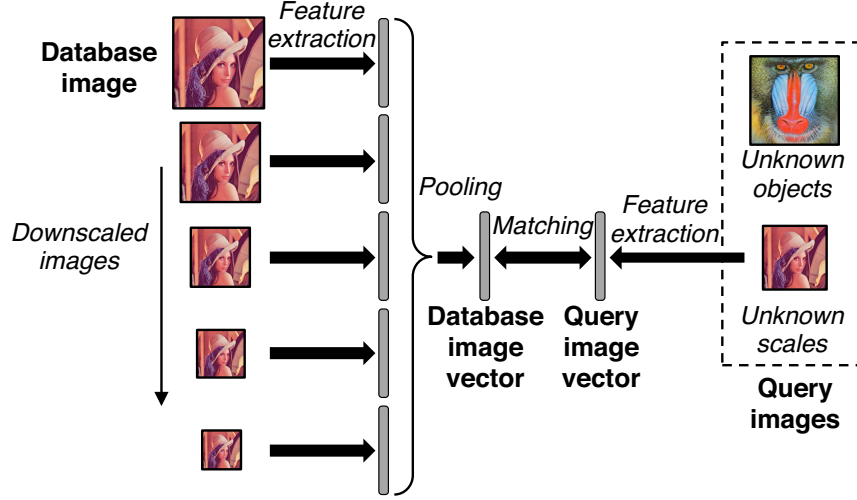
Fig. 16. We extract features from database images at different scales and pool them to obtain a single representation. We scale queries to different sizes, and evaluate retrieval performance for different pooling parameters and strategies.

over which *OxfordNet* features are pooled. $SP = n$ refer to pooling across the first $n + 1$ scales of the set of six scale-ratios (seven including one) $(1, 0.75, 0.5, 0.375, 0.25, 0.2, 0.125))$. $SP = 1$, hence, refers to no database pooling.

In Figure 17(a), we first study MAP vs query scale for different types of pooling on the *Holidays* data set for $SP = 6$ (pooling over all scales in consideration). *OxfordNet fc6* features are used in this experiment. We note that max-pooling outperforms average pooling by a small margin, and comes close to the performance of the *Min-dist* scheme, which stores the descriptors of all the scaled versions of the database image and computes the minimum distance. Similar to the rotation experiment, the *Min-dist (PWL)* scheme, which computes the minimum distance to a piece-wise linear manifold of the CNN descriptors for the six scaled images, is also effective for the scale experiment. *Min-dist (PWL)* outperforms *Min-dist* by a small margin, as it is more robust to matching query data which lie at intermediate quantized scales. For $SP = 6$, there is a significant improvement in performance at small scales for the pooling schemes, with only a marginal drop in performance for points close to the original scale (seen from the right most points on the curve in Figure 17(a)).

In Figure 17(b), we study varying pooling parameter SP for max-pooling. Performance at small scales increases as SP is increased, with only a marginal drop at query scale 0.75. A significant gain in performance of 10% is achieved for the smallest query scale 0.25, showing the effectiveness of the max-pooling approach.
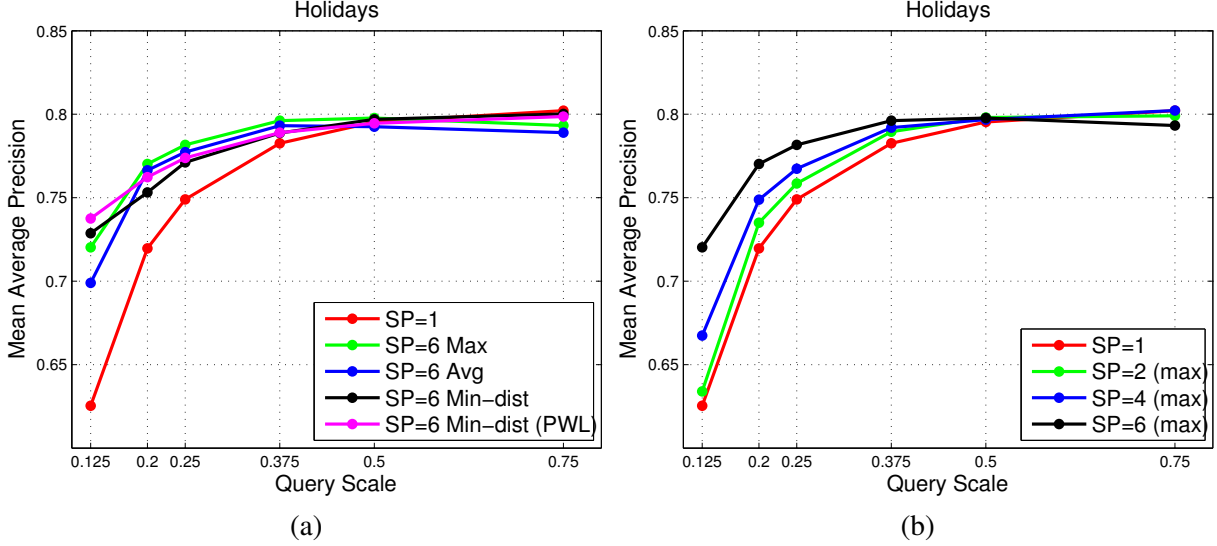
Fig. 17. Performance of different database-side pooling schemes, as query scale is changed. Results reported on *OxfordNet* $fc6$ on the *Holidays* data set. $SP = 1$ refers to no database-pooling. $SP = n$ refer to pooling across the first $n + 1$ scales of the set $(1, 0.75, 0.5, 0.375, 0.25, 0.2, 0.125))$. In Figure (a), we notice that max pooling comes close to the performance of *Min-dist* which requires storing descriptors at all scales. In Figure (b), we observe that performance improves at small scales with database side pooling as parameter SP is increased.

VI. OPEN QUESTIONS

The systematic study in this paper opens up several interesting avenues for future work. We highlight the most important open questions here.

- Pre-trained CNN models trained for large-scale image classification tasks, with larger amounts of data have the potential of improving performance further for instance retrieval. For instance, CNN models trained on the full ImageNet data set with 14 million images and 10000 classes could lead to more discriminative features for the instance retrieval task.

- While supervised CNN models have far outperformed their unsupervised CNN counterparts for large-scale image classification, the latter approach deserves careful attention in the context of instance retrieval. For the instance retrieval task, we desire rich representations of low level image information, which can be learnt directly from the large amounts of unlabelled image data available on the internet. As image classification is not the end goal, unsupervised CNN models trained with large amounts of data might achieve comparable or better performance for instance retrieval tasks. Availability of large amounts of training data (e.g., the Yahoo 100 million image data set [46]) and recent advances in open-source software for large-scale distributed deep learning (e.g. Torch [47]) will enable training of large-scale unsupervised CNN models. If unsupervised CNN models work well for instance retrieval,

they will enable easier training and adaptation to different types of image databases.

- Rotation and scale invariance are key to instance retrieval tasks. While the database pooling schemes proposed in this work are highly effective, they are more of an after-thought to solving the invariance problem in the CNN context. Learning CNN representations which are inherently scale and rotation invariant is an exciting direction to pursue.

- Interest point detectors provide an efficient and effective way of achieving desired levels of invariance (ranging from scale and rotation invariance to affine invariance). The carefully hand crafted SIFT descriptor has been remarkably effective for the instance retrieval task: however patch level descriptors can now be learnt with large amounts of data, using data sets like the Winder and Brown patch data sets [48], and the Stanford Mobile Visual Search patch data set [49]. A hybrid approach of interest point detectors with learnt CNN descriptor representations could lead to a significant improvement in retrieval performance.

- Hybrid interest point detection schemes like the dense interest point detector proposed originally in [50] need to be revisited, in light of the effectiveness of CNN features which are extracted by dense sampling in the image. A recent survey of dense interest point detectors [51] is a good starting point.

- Finally, our study has demonstrated that unlike for large scale image classification, combining CNNs with "less effective" types of descriptors such as FVs is a valid way to improve retrieval performance. We point to readers the recent work from Gong et al. [52] and Xu et al. [53] in this field who have proposed effective FV/VLAD style encoding schemes for CNN descriptors.

VII. CONCLUSIONS

In this work, we proposed a systematic and in-depth evaluation of FV and CNN pipelines for image retrieval. Our study has lead to a comprehensive set of practical guidelines we believe can be useful to anyone seeking to implement state-of-the-art descriptors for image retrieval. Some of the recommendations are general good practices while others are more problem specific.

We also showed that unlike image classification, the supremacy of CNNs over FVs does not always verify in the case of image retrieval and strategies mixing both approaches are most likely optimal. In particular, the lack of transformation invariance of the descriptors appears to be one of the main drawbacks of CNNs. We managed to propose a number of simple and effective approaches which can be followed to patch these deficiencies. Nevertheless, we believe that better integrating invariance is key to the improvement of performance.

REFERENCES

[1] L.-Y. Duan, R. Ji, Z. Chen, T. Huang, and W. Gao, "Towards Mobile Document Image Retrieval for Digital Library," *Transactions on MultiMedia (TMM)*, vol. 16, no. 2, pp. 346–359, 2014.

[2] D. Chen and B. Girod, "A Hybrid Mobile Visual Search System with Compact Global Signatures," *Transactions on MultiMedia (TMM)*, vol. 17, no. 7, 2015.

[3] R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, and W. Gao, "Learning to Distribute Vocabulary Indexing for Scalable Visual Search," *Transactions on MultiMedia (TMM)*, vol. 15, no. 1, pp. 153–166, 2013.

[4] L.-Y. Duan, J. Lin, Z. Wang, T. Huang, and W. Gao, "Weighted Component Hashing of Binary Aggregated Descriptors for Fast Visual Search," *Transactions on MultiMedia (TMM)*, vol. 17, no. 6, pp. 828–842, 2015.

[5] J. W. Y.-G. Jiang, X. Xue, and S.-F. Chang, "Query-Adaptive Image Search with Hash Codes," *Transactions on MultiMedia (TMM)*, vol. 15, no. 2, pp. 442–453, 2013.

[6] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale Image Retrieval with Compressed Fisher Vectors," in *Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 3384–3391.

[7] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice," *International Journal of Computer Vision (IJCV)*, vol. 105, no. 3, pp. 222–245, Dec. 2013.

[8] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, November 2004.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1–9.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1701–1708.

[12] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1891–1898.

[13] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 2056–2063.

[14] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1653–1660.

[15] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural Codes for Image Retrieval," in *European Conference on Computer Vision (ECCV)*, 2014.

[16] V. Chandrasekhar, L. Jie, O. Morère, A. Veillard, and H. Goh, "Compact Global Descriptors for Visual Search," in *Data Compression Conference (DCC)*, Snowbird, Utah, March 2015.

[17] G. Tolias, Y. Avrithis, and H. Jegou, "To aggregate or not to aggregate: Selective match kernels for image search," in *International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 1401–1408.

[18] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Processing (SP)*, vol. 93, no. 8, pp. 2316–2327, 2013.

[19] J. Lin, L.-Y. Duan, T. Huang, and W. Gao, "Robust Fisher Codes for Large Scale Image Retrieval," in *International Conference on Acoustics and Signal Processing (ICASSP)*, 2013.

[20] H. Jégou and A. Zisserman, "Triangulation embedding and democratic aggregation for image search," in *Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, April 2014.

[21] K. Mikolajczyk, *Software for Computing Hessian-affine Interest Points and SIFT Descriptor*, 2010.

[22] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014, pp. 512–519.

[23] H. Jégou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," in *European Conference on Computer Vision (ECCV)*, Berlin, Heidelberg, October 2008, pp. 304–317.

[24] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," in *Computer Vision and Pattern Recognition (CVPR)*, New York, USA, June 2006, pp. 2161–2168.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, June 2007, pp. 1–8.

[26] V. Chandrasekhar, D.M.Chen, S.S.Tsai, N.M.Cheung, H.Chen, G.Takacs, Y.Reznik, R.Vedantham, R.Grzeszczuk, J.Back, and B.Girod, "Stanford Mobile Visual Search Data Set," in *MultiMedia Systems Conference (MMSys), San Jose, California, February 2011*, 2011.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *International Conference on Multimedia (ICM)*, 2014, pp. 675—-678.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[29] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *arXiv preprint arXiv:1501.02876*, 2015.

[30] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project Adam: Building an Efficient and Scalable Deep Learning Training System," in *Operating Systems Design and Implementation (OSDI)*. Broomfield, CO: USENIX Association, Oct. 2014, pp. 571–582.

[31] *MPEG CDVS (Compact Descriptors for Visual Search) Benchmark. Stanford Digital Repository*.

[32] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *International Conference on Multimedia (ACM)*, 2010, pp. 1469–1472.

[33] G. Francini, S. Lepsøy, and M. Balestri, "Selection of local features for visual search," *Signal Processing: Image Communication (SPIC)*, vol. 28, no. 4, pp. 311–322, 2013.

[34] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 9, pp. 1704–1716, 2012.

[35] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556*, 2014, pp. 1–10.

[36] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[37] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3485–3492.

[38] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 413–420.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database," in *Computer Vision and Pattern Recognition (CVPR)*. Ieee, Jun. 2009, pp. 248–255.

[40] F. Yang, Z. Jiang, and L. S. Davis, "Submodular Reranking with Multiple Feature Modalities for Image Retrieval," in *Asian Conference on Computer Vision (ACCV)*, 2014.

[41] H. Jégou, M. Douze, C. Schmid, and P. Perez, "Aggregating Local Descriptors into a Compact Image Representation," in *Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 3304–3311.

[42] A. Gordoa, J. Rodriguez-Serrano, F. Perronnin, and E. Valveny, "Leveraging category-level labels for instance-level image retrieval," in *Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 3045–3052.

[43] R. Arandjelović and A. Zisserman, "All about VLAD," in *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[44] T. Ge, Q. Ke, and J. Sun, "Sparse-coded features for image retrieval." British Machine Vision Conference (BMVC), September 2013.

[45] G. Takacs, V. Chandrasekhar, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod, "Unified Real-time Tracking and Recognition with Rotation Invariant Fast Features," in *Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, June 2010, pp. 934–941.

[46] Yahoo, *Yahoo! 100 million image data set*.

[47] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning software library," IDIAP, Tech. Rep., 2002.

[48] S. Winder, G. Hua, and M. Brown, "Picking the Best DAISY," in *Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, June 2009, pp. 178–185.

[49] *CDVS Patches*, 2013.

[50] T. Tuytelaars, "Dense interest points," in *Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2281–2288.

[51] A. Iscen, G. Tolias, P. Gosselin, and H. Jegou, "A comparison of dense region detectors for image search and fine-grained classification." *Transactions on Image Processing (TIP)*, 2015.

[52] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 392–407.

[53] A. H. Z. Xu, Y. Yang, "A Discriminative CNN Video Representation for Event Detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.