1-1-2016

Image Descriptors from ConvNets: Comparing Global Pooling Methods for Image Retrieval

Ian Comor
University of Wollongong

Yan Zhao
University of Wollongong, yz298@uowmail.edu.au

Zhimin Gao
University of Wollongong, zg126@uowmail.edu.au

Luping Zhou
University of Wollongong, lupingz@uow.edu.au

Lei Wang
University of Wollongong, leiw@uow.edu.au

Image Descriptors from ConvNets: Comparing Global Pooling Methods for Image Retrieval

Abstract

2016 IEEE.A major component of a generic image retrieval pipeline is producing concise and effective descriptors for each image. Previous works have shown impressive results in image retrieval when using descriptors from the black-box output of the fully-connected stage of pretrained Convolutional Neural Networks (ConvNets). However, previous work on descriptors pooled from the deep feature maps from late convolutional layers can produce more discriminative descriptors for generic image retrieval, while being relatively concise. When planning to globally pool such feature maps from a ConvNet, some options to consider are (1) the depth of the network, (2) choice of layer to pool, and (3) the level of dimension reduction. The previous work on global pooling methods uses differing techniques without a clear consensus on which method is best. This motivates us to establish a baseline pipeline from which to compare these options and their effect on retrieval results. Our contribution is a systematic and comprehensive experimental study of different pooling strategies of deep features for image retrieval, and the various options. Our results show that the nature of the dataset (object- heavy or scene-heavy) warrants a different pooling strategy. Significantly, we visualise the level of image discrimination brought by the different pooling methods on the datasets, and show that pooling need not have a priori spatial weights to effectively find objects within the image. The results underline the need to consider the context of the image dataset when developing image retrieval pipelines using ConvNets.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Comor, I., Zhao, Y., Gao, Z., Zhou, L. & Wang, L. (2016). Image Descriptors from ConvNets: Comparing Global Pooling Methods for Image Retrieval. 2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016 (pp. 1-8). IEEE Xplore: IEEE.

Image Descriptors from ConvNets: Comparing Global Pooling Methods for Image Retrieval

Ian Comor, Yan Zhao, Zhimin Gao, Luping Zhou, Lei Wang
School of Computing and Information Technology
University of Wollongong
Wollongong, NSW, Australia
Email: {ic286, yz298, zg126}@uowmail.edu.au, {lupingz, leiw}@uow.edu.au

***Abstract*—A major component of a generic image retrieval pipeline is producing concise and effective descriptors for each image. Previous works have shown impressive results in image retrieval when using descriptors from the black-box output of the *fully-connected stage* of pre-trained Convolutional Neural Networks (ConvNets). However, previous work on descriptors pooled from the deep *feature maps* from late convolutional layers can produce more discriminative descriptors for generic image retrieval, while being relatively concise. When planning to globally pool such feature maps from a ConvNet, some options to consider are (1) the depth of the network, (2) choice of layer to pool, and (3) the level of dimension reduction. The previous work on global pooling methods uses differing techniques without a clear consensus on which method is best. This motivates us to establish a baseline pipeline from which to compare these options and their effect on retrieval results. Our contribution is a systematic and comprehensive experimental study of different pooling strategies of deep features for image retrieval, and the various options. Our results show that the nature of the dataset (object-heavy or scene-heavy) warrants a different pooling strategy. Significantly, we visualise the level of image discrimination brought by the different pooling methods on the datasets, and show that pooling need not have *a priori* spatial weights to effectively find objects within the image. The results underline the need to consider the context of the image dataset when developing image retrieval pipelines using ConvNets.**

I. Introduction

Image retrieval is an intuitive tool whereby the user presents an image as a query (instead of a text-based search) to a system with an image dataset, and gets back a set of dataset images in order of similarity to the query image. Each image in the dataset needs to be described by a concise descriptor to allow for rapid (and accurate) similarity comparisons to the query image's descriptor. Image retrieval has previously been conducted using techniques such as SIFT features [1], and the colour histogram [2]. The newest frontier in powerful image retrieval is based on deep learning, using Convolutional Neural Networks (ConvNets). Like the other methods, pre-trained ConvNets analyse an image and produce a useful descriptor (Fig. 1).

Their power, utility, and ease of use for object and scene classification have led to the production of a number of freely available pre-trained networks accessible to the community [3] [4] [5] [6]. But ConvNets have historically been trained to classify images as one of a limited number of categories, so what is the idea in using them for image retrieval? As it turns out, feeding two visually similar images separately through a pre-trained ConvNet produces two concise descriptors that are geometrically closer than the descriptors of two dissimilar images. The input images do not even have to be of the same context as the database in which the ConvNet was trained [7] [8] [9].

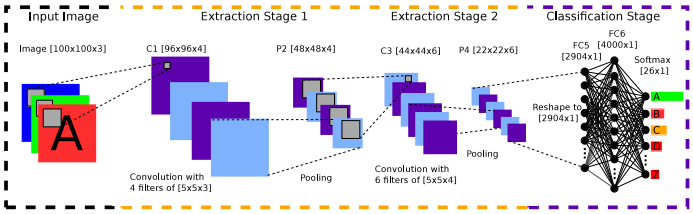

Fig. 1. A basic ConvNet trained to recognise images of letters, with two feature extraction stages and a classification stage. Note the input image is a volume and intermediate stages are volumes of feature map 'slices'. Learned feature maps are convolved in a sliding window manner over the image to produce feature maps. Each pooling stage subsamples the feature maps, reducing the resolution by half. The resulting feature volume P4 enters the classification stage, which abstracts the spatial information. In this example, there are 26 possible classification outputs, A to Z.

The white-box nature of the *convolutional stages* of the ConvNet (Fig. 1) means information can be extracted from the output of any intermediate layer and pooled into a concise vector. However, simply using an off-the-shelf ConvNet to carry out generic image retrieval may not produce the best results. What happens if a different ConvNet is used entirely, or the descriptor is extracted and pooled differently, or the resulting descriptor made more compact? The positions of objects in the images may be consistent, or may be at random locations. While these issues have fuelled some examination in the literature about better utilising ConvNets in an image retrieval pipeline [10] [11] [12] [13], there is a critical need to understand how these issues affect results, and whether there is a 'one size fits all' approach.

We will use a set of baseline options, including a sum-pooling strategy and a particular convolutional layer, and show how the results are affected by choosing different options against the baseline. Since visually similar images produce 'close' vectors, it is anticipated that changes to irrelevant noise (unimportant parts of the image) affect the descriptor less significantly, while changes to salient objects should change the descriptor more significantly. We experiment on deep-learning-based methods for image retrieval that use ConvNets

to produce image descriptors. Using a framework to compare current global pooling methods from the literature and options relating to extraction, we provide a comprehensive set of results that verify that the options to choose rely on the context of the dataset in question.

For the rest of this paper, we overview the related work to this study (section II), explain important pooling methods (section III) and our results (section IV). Finally, we provide our discussion of these results (section V) and conclusion (section VI).

II. Related Work

The power of ConvNets in the computer vision community was initially their ability to classify images of objects with unprecedented accuracy [3] [4]. More recently, the outputs of the classification stage of existing trained ConvNets have been found to have discriminative power useful for image retrieval, while being relatively concise [11] [14] [15].

ConvNets (Fig. 1) are neural networks of learnable weights designed to accept image data, and are structured as a series of feature extraction stages [16], generally followed by a fully-connected (FC) classification stage [4]. Each feature extraction stage typically has a convolutional layer, a rectifying layer (ReLU), and a pooling/subsampling layer. The convolutional layer convolves a set of learnable image kernels over the input volume, and produces a set of output feature maps. The ReLU layer sets all negative values to zero, and has been shown to learn faster than traditional non-linear functions [17], despite causing information loss [7]. The pooling layer subsamples the feature maps, typically with the $\max(\cdot)$ function, to introduce minor spatial invariances [18]. The FC stage is reminiscent of the traditional neural network, and abstracts the spatial information of the convolution stages. The final layer is generally a softmax function used for classification [4].

Much work has been done utilising this final output for vision tasks, including image retrieval [11] and scene recognition [19]. However, outputs from the fully-connected stage is essentially the output of a black box. Stepping back a few layers in the ConvNet reveals other intermediate outputs of the ConvNet's convolutional, ReLU, and pooling layers [12] [10] [13] [20]. Suddenly the black box nature of the image descriptor is shattered, and one can see highly semantic information that highly represents the structure of the input image [12] [10] [21] [13] [22] [23]. But can any layer be selected for extraction? [24] and [4] show that feature maps of earlier layers look for simple features, and one needs to focus on the later layers to find high-level semantics.

Then there is the question of what to *do* with the feature maps to produce a descriptor: how does one reconcile a strategy in extracting usable descriptors considering the nature of the dataset? One might make some assumptions about how to increase their utility: focus on particular spatial regions [12] where we 'expect' objects to be, or perhaps assume we'll find them everywhere [20]. But if the dataset is a set of scenes, maybe there will be very few objects at all!

Once we have a descriptor for each image, is that the end of it? Perhaps not; the descriptor can be reduced in dimensionality [14]. Perhaps a smaller descriptor may contain less information, but it could just as easily remove redundant information. [10] shows dimension reduction impacting negatively on precision.

Looking at all these questions, regarding layer choice, pooling strategy, and dimension reduction, there is a need to find which options improve image retrieval precision. There are some existing clues: semantic information peaks at later convolutional layers [7] [13] [10] or shows improvement over earlier layers [22], so the later layers have a more powerful ability to recognise high level concepts. We therefore aim our attention at the very final convolutional and pooling layers before the FC stage.

The output of a non-FC layer is a volume of information that can be pooled to produce a vector. This, like the output of the FC stage, can act like a global descriptor. What kinds of pooling strategies can take place?

Previously, [11] took the raw descriptors of the final convolutional layer and the FC layers to perform image retrieval. While their results showed FC layers providing better performance than the deep features, they later experimented on a method to sum-pool the deep features on a map-per-map basis in [12]. This 'SPoC' descriptor used a gaussian weight scheme over the feature maps to give more attention to the object in the image's spatial center. This was followed by a post-processing step of whitening and dimension reduction. This scheme gives prior bias to centred objects, and could unduly hinder the ability to discriminate useful objects near the image borders.

[10]'s cross-dimensional weighting scheme utilises not only the spatial information in each feature map, but also the information across the feature maps. Their 'CroW' descriptor performs a basic pooling operation before calculating weighting factors for not only the spatial locations, but across the feature maps.

Why has focus shifted from the classification stage to the feature extraction stages for useful descriptors? An interesting observation of [10] is the substantial retention of spatial information in the feature maps of the final convolutional and pooling layers. Their visualisations of the feature maps reveal a kind of low-resolution greyscale representation of the original image. This suggests that despite going through a series of complex convolutions, rectifications, and poolings, the spatial positions of features in the original image are strongly retained in the final feature maps, but are spliced across the feature maps. This means that the feature maps could be split into regions that correspond to the spatial locations of the input image. [25] demonstrated that spatial max-pooling of a 2×2 grid over the convolutional feature maps led to improved results over global max pooling. This essentially creates four vectors of pooled values, and theoretically takes advantage of the spatial positions of features within the feature maps. Despite not being strictly a global technique, the 2×2 grid works on convolutional feature maps of any arbitrary size. Similarly,

[22] utilised a pyramid max-pooling strategy by taking regions of multiple scales and regions, and concatenating them into a vector. [21] also used multiple scales and regions, but split the *original* image instead, and fed each r regions separately into the ConvNet, and performed r^2 comparisons for each reference image. This is out of scope of global descriptors and is not covered here, and we look at 2×2 max pooling only.

Although much progress has been achieved for image retrieval with various pooling strategies applied on deep features, a number of options in image retrieval pipelines have not yet arrived at consistent opinions. This work aims to highlight the advantages and disadvantages of these strategies on several datasets and provide practical guidance for deep-feature-based retrieval.

III. Important Pooling Methods

The related work presented some pooling ideas that have been implemented in the literature. We compare five feature pooling strategies in our image retrieval experiments to produce image descriptors, each forming a vector $\mathbf{v} \in \mathbb{R}^C$.

Sum Pooling (SumPool): This is the simplest case, and takes the sum of each spatial position in each of C feature maps, to produce the vector of C length. Feature maps are of height H and width W. Each feature map F_i is sum-pooled to form a scalar:

$$s_i = \sum_{y=1}^{H} \sum_{x=1}^{W} F_{i,(x,y)} \tag{1}$$

where $F_{i,(x,y)}$ is the position (x, y) of the i^{th} feature map, to form $\mathbf{v} = [s_1, s_2, ..., s_C]^T$.

MAC: This technique uses the $\max(\cdot)$ function over each feature map instead of the summation in **SumPool**. The technique [25] [20] follows a similar idea to the pooling regions of the ConvNet pooling layers, but this uses a 1×1 (global) grid over each feature map. This is carried out over each feature map in order to collapse each to a scalar, to make the final vector $\mathbf{v}$:

$$\mathbf{v} = [\max(F_1), \max(F_2), ..., \max(F_C)] \tag{2}$$

2×2 Max Pooling (2×2MAC): This divides the feature volume into four equal-sized regional volumes of size $[\frac{H}{2} \times \frac{W}{2} \times C]$. A MAC vector $\mathbf{v}_i$ is produced from each region i, $i \in 1..4$, and we get $\mathbf{v}_1..\mathbf{v}_4$. The four vectors are concatenated:

$$\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{v}_4^T]^T \tag{3}$$

This results in a vector of length $4C$.

SPoC: This descriptor by [12] performs the sum-pooling operation with a gaussian weighting operation on the feature maps, designed to give more weight towards features at the center of the image. The weighting operation gives the weight at (x, y) as:

$$w_{(x,y)} = \exp\left(- \frac{(y - \frac{H}{2})^2 + (x - \frac{W}{2})^2}{2\sigma^2} \right) \tag{4}$$

Each resulting element in the i^{th} feature map is weighted as $F_{i,(x,y)} \times w_{(x,y)}$. The final vector $\mathbf{v}$ is produced by the sum-pooling of the new weighted volume.

CroW: This descriptor by [10] exploits sparsity information across the feature volume, and performs weighting over different spatial locations and different feature maps.

The C feature maps firstly undergo a pooling step, then are aggregated to form a 'supermap' of the same spatial size, where each spatial position is the sum of the position in all C feature maps:

$$F_{super} = \sum_{i=1}^{C} F_i \tag{5}$$

Each spatial position of F_{super}, namely $F_{(x,y)}$, is powerscaled to produce the spatial weights:

$$\alpha_{(x,y)} = \left(\frac{F_{(x,y)}}{(\sum_{x=1}^{W} \sum_{y=1}^{H} F_{super}^{\gamma})^{1/\delta}} \right) \tag{6}$$

To produce the feature map weights, [10] finds the non-zero elements of a feature map i, namely Q_i, calculated by the proportion of non-zero elements in the feature map:

$$Q_i = \frac{\sum_{x=1}^{W} \sum_{y=1}^{H} \mathbb{1}[F_{i,(x,y)} > 0]}{W \times H} \tag{7}$$

The feature map weight β for feature map i is then produced by:

$$\beta_i = \log \left(\frac{C\epsilon + \sum_{h=1}^{C} Q_h}{\epsilon + Q_i} \right) \tag{8}$$

Each feature map is then sum-pooled with the weights:

$$s_i = \sum_{x=1}^{W} \sum_{y=1}^{H} F_{i,(x,y)} \alpha_{(x,y)} \beta_i \tag{9}$$

to form the vector $\mathbf{v} = [s_1, s_2, ..., s_C]^T$.

IV. Experimental Results

Our experiments compare different options against the baseline to establish whether the option improves the retrieval results in the pipeline. We show the results of changing the ConvNet depth, the layer used in pooling, and the dimension reduction, across the different pooling strategies and datasets. We choose four datasets from which to produce experimental results.

A. Datasets

INRIA Holidays: The INRIA Holidays dataset [26] contains 1491 holiday-themed colour photographs in 500 small groups, where each group contains photos of the same scene or object from a different viewpoint. The first image of each group is the query, and the other images in that group are the groundtruths. Following [12], all non-upright images are manually rotated to an upright position.

Paris6k: The Paris6k dataset [27] contains 6392 images (after the 20 corrupted images are removed) of landmark buildings and objects in Paris. There are 55 queries and a

given set of groundtruths. While the queries are of landmarks, the dataset includes a lot of 'distracting' images.

Oxford5k: The Oxford5k Buildings dataset [28] is similar to Paris6k, and contains 5063 images, with 55 queries and a given set of groundtruths.

NAA29k: We also use the dataset 'NAA29k', which is a subset of the immense digitised PhotoSearch gallery (of 350,000 images) from the National Archives of Australia. This dataset contains 28,912 scene-heavy images from different eras of Australian history, and are of diverse sources and context. The images are mostly black and white, and range from personal portraits and workplaces to landscapes and natural scenery. Each image was resized to 256×256. 600 groundtruth queries were manually produced, and formatted in the same way as Paris6k and Oxford5k.

Notably, the Paris6k and Oxford5k are semantically similar and contain lots of objects, while the INRIA Holidays and NAA29k datasets have more scene-based imagery.

B. Implementation Details

In preliminary experimentation, we found that the deep networks VGG16 and VGG19 trained on the ImageNet database of objects [5] had overall poorer performance than the place_vgg16 and place_vgg19 trained on scene data [6]. We therefore focus our experimentation on the latter two networks. The preliminary experiments also looked at PCA whitening [29] versus no whitening, and found whitening to be better in all cases. Thus all the experiments shown here perform whitening, even when no dimension reduction occurs. In [10] the whitening parameters from Oxford5k are used in Paris6k and vice versa. However, in our experiments, we use the whitening parameters only from the dataset being queried.

For the SPoC method, [12] follows the three-sigma rule. We set σ to be $\frac{H}{3}$.

For the CroW method, [10] set $\gamma = \delta = 0.5$. We set $\gamma = 2$ to ensure F_{sum} is positive (in case the layer used in extraction does not follow a ReLU layer), but keep $\delta = 0.5$. We also set $\epsilon = 0.01$. The initial pooling step is only implemented in the pool5 layers in our experiments. Also note that [10] use the 16-layer network trained on the ImageNet dataset [5] in their experiments.

For all experiments, we 'eliminate' the query from the results list, and the query is never in the groundtruth of the retrieval result.

We also ignore query bounding boxes in the Oxford5k and Paris6k datasets, and use the entire image as the query, in order to be methodologically equal to the INRIA Holidays and NAA29k datasets.

The post-processing step after extracting the pooled vector always includes l_2-normalisation:

$$\mathbf{v}' = \frac{\mathbf{v}}{||\mathbf{v}||_2} \tag{10}$$

then PCA dimension reduction and whitening [29], followed again by l_2-normalisation.

The pooling method ultimately controls how the feature maps are used to form a concise vector. With our aim to provide a convincing overview of the advantages and disadvantages of these options, we use a baseline comparison which worked generally well across all four datasets. This baseline uses the place_vgg19 network, the conv5_3 layer, 512 dimensions, and the SumPool strategy. The extraction pipeline focuses on pooling, PCA post-processing and dimension reduction. Querying always uses the l_2 distance to determine the closeness of descriptors (and therefore similarity of images). The mean average precision (mAP) calculation from [26] is used, which is a single-value measure (averaged over all queries) of the distribution of positive results down the ranked list.

We use the Caffe toolbox [30] to extract features. Caffe takes 224×224 crops of the images as desired by the two networks. While this image resolution is not as high as in [11] and [10], we keep this crop the same for all images in all the four datasets used. For preprocessing, we subtract the mean pixel from each image before feeding it through the ConvNet. Since later layers contain more semantic information [22], we experiment on the last two convolutional layers (*conv5_4* and *conv5_3* for place_vgg19 [6], and *conv5_3* and *conv5_2* for place_vgg16 [6]), and the final max-pooling layer *pool5*.

We used the baseline options and varied one option each time, and we show the relative results of each experiment.

Firstly, we alter the network on the baseline but keep all other options the same. We should expect a deeper network to provide more discriminative power over a shallower network [5]. Some pooling strategies should also favour the type of dataset being queried. Fig. 2 shows that the deeper 19-layer network provides better results in *all* cases over the 16-layer network trained on the same data. However, the 2×2MAC showed the least improvement in results from the 16-layer to 19-layer network. Furthermore, there is a clear advantage of the SumPool and CroW methods over the other methods in all datasets. This implies that the SumPool and CroW are naturally more powerful in their image discriminative abilities.

In the second experiment, we use the baseline with the 19-layer network, but change the convolutional layer from which to extract the pooled features. We may expect to see poorer results on the pool5 layer due to loss of information caused by pooling. Again we compare the retrieval results on the four datasets using all five pooling strategies. The results are shown in Fig. 3, and show the conv5_3 layer to be superior in most cases. Interestingly, the 2×2MAC strategy was less responsive to the change in layer, while the SumPool and CroW strategies were most responsive.

The above experiments were carried out on the baseline using 512 dimensions. The CroW, SPoC, SumPool, and MAC strategies are therefore not yet reduced, while the 2×2MAC strategy has been reduced from its original 2048 dimensions. To see how further dimension reduction affects results, we repeat the baseline experiment on all four datasets and all five strategies by reducing their dimensionality down to 16. The results are shown in Fig. 4. Reducing the dimensionality for the Oxford5k and INRIA Holidays datasets had little impact on the results until dimensions were under 100, after

which the results worsened (Fig. 4). The standout is the Paris6k dataset, which actually *increased* performance when decreasing from 512 to 32 dimensions under our experimental conditions. We attribute this to the nature of the dataset, whereby information of simple buildings was more accurately represented by reduced descriptors than non-reduced ones, and the smaller crop sizes. Under reduced dimensions, the discriminatory nature of the descriptors were not distracted by other objects in the image. To visualise this in Fig. 6, we resize the image of the Eiffel Tower and discover how the resulting descriptors change according to differences made in the image. By greying out parts of an image and feeding it through a ConvNet, we should expect to see the final descriptor change somewhat - but removing important parts should change the descriptor more than removing unimportant parts. We resized the image to 256×256 and sequentially greyed out blocks of 32×32 pixels, and fed them through the places_vgg19 network and extracted a SumPool descriptor from the conv5_3 layer. This strategy follows a similar technique in [31]. Reducing the dimensions shows the discrimination of the descriptor favour the salient parts (the tower) rather then the surrounds. This implies that the reduced dimensions did in fact reduce the 'noise', but eventually became too concise to work effectively.

Noticeably in all these experiments, both the Oxford5k and Paris6k datasets are slightly favoured by the CroW method, while the SumPool method was good all round. This shows that the nature and context of the dataset is important in choosing the options for descriptor extraction. Visually, Oxford5k and Paris6k are similar in that there are highly discriminative buildings as queries, with lots of distracting images. The most discriminative differences came from the change in pooling method, whereby taking specific locations was overall detrimental to finding the salient features within an image.

To visualise more broadly how the different pooling method discriminate features within images, we perform a similar visualisation technique as in Fig. 6, but using several images that are either object-focussed or scene-focussed, using at least one image from each of the four databases. Pronounced changes in the descriptors (compared to the original image's descriptor) should signify more discriminative feature discovery by the pooling strategy. The results in Fig. 7 show that pooling the deep features does find discriminative features. However, the SPoC strategy favours centred features, even if they are not as salient, while the other strategies favour discriminative features regardless of spatial position, such as the aeroplane engine, columns, and spire.

V. Discussion

Our results are consistent with [12], whereby the gaussian operation in SPoC decreases performance of the pooling operation on Holidays, and favours centred objects. This is, intuitively, due to the largely scene-like images of the Holidays dataset. SPoC also performed the worst on the NAA29k dataset. For scene-like images, there is no guarantee that highly-discriminative features will appear at the center. Since

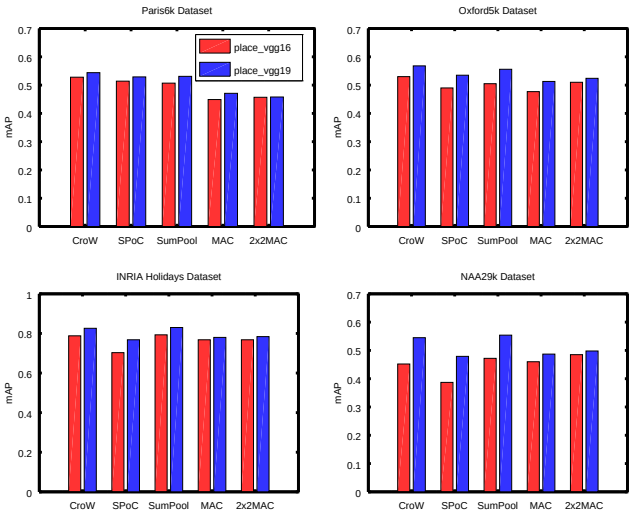

Fig. 2. Comparison of the two networks places_vgg16 and places_vgg19. For each pooling method on each dataset, places_vgg19 always produced a higher mean average precision. In particular, the NAA29k shows more pronounced improvement than the smaller datasets.

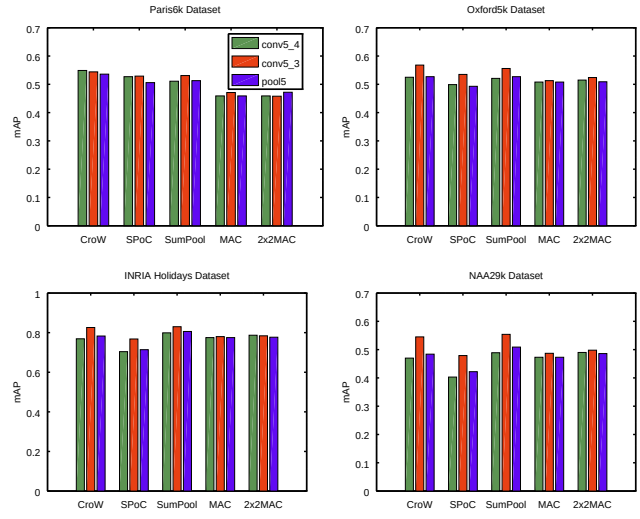

Fig. 3. The mAP values when using a different convolutional or pooling layer on the baseline. The conv5_3 layer is best in 85% of the cases presented. CroW on the Paris6k dataset showed the best result on features extracted from the conv5_4 layer, while all other datasets showed conv5_3 to be superior.

the other strategies could find centred objects as well (Fig. 7), there appears no need to perform a specific spatial bias in the pooling method.

The NAA29k dataset responded best to the simpler SumPool descriptor, and we suspect this is due to the more diverse collection of image types and contexts. The advantage of the SumPool descriptor was also noticeable on the Holidays dataset, which has mostly scene-like images.

A. What Properties to Choose?

These experiments on pooling deep convolutional features of ConvNets for image retrieval highlight the issue of choosing the right options for a particular dataset. We would suggest a

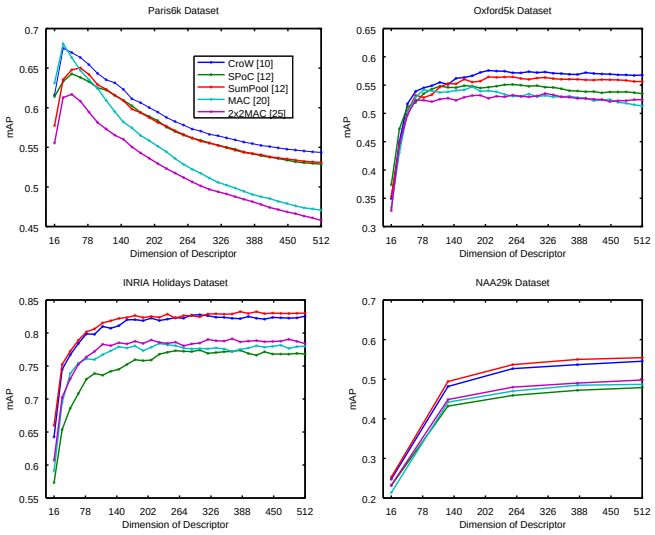

Fig. 4. Mean average precision of each method using different amounts of dimension reduction. Note that CroW favours Paris6k and Oxford5k while SumPool favours Holidays and NAA29k. Also note the very early peak in Paris6k - manual inspection of the top image results suggests the simple building structures in the Paris6k dataset require less dimensions to retain basic structural information, while higher dimensionality introduces distracting 'noise' from irrelevant surrounding objects.

visual analysis of the dataset in question: is it more object-oriented or scene-oriented? Do objects or features occur at different spatial locations?

Our empirical results show that some options can be left the same: if using the 19-layer scene-trained network [6], sticking to the conv5_3 may be best no matter the dataset being used.

However, it is also clear that dimension reduction should be investigated for its effects, as it can be better for some datasets while not others (Fig. 4), depending on image size and other experimental properties. We attribute the unusual decrease in performance at higher dimensions in Paris6k (compared to [10]) to the small image size and nature of the dataset. The query images and their positive results contain large buildings that are highly recognisable when the image is resized, while other distracting features are eliminated.

B. What is the Query Looking For?

While the results presented here convincingly present options that lead to better precision, the groundtruth and dataset are also barriers to improvement. In a dataset such as NAA29k, which may contain rare images, a key goal would be to query (for example) an image of a building and find *all* instances of that building in the dataset, even if occluded, small, or distant. However, the Paris6k and Oxford5k datasets place such images in the 'junk' category. When visually observing some results of queries, these junk results would sometimes appear. From the framework used, these are a detriment to performance, despite the query stage successfully retrieving the image containing the distant or occluded building. In this case, the power of the system was its own detriment, so for some image retrieval pipelines, it can be useful to consider junk results (Fig. 5).

VI. Conclusion

Experiments on deep-learning-based image retrieval using pooled ConvNet descriptors showed that object-heavy datasets are favoured by pooling methods that find specific spatial features. Comparing the Sum-Pooling method, the SPoC method, the CroW method, a MAC and a 2×2 MAC method, confirmed that the choice of pooling method has a strong effect on the query results. Importantly, object-heavy datasets were favoured by the CroW method, while the scene-based datasets were favoured by the SumPool method. Using a visualisation technique to examine the discriminative ability of the pipeline revealed the bias of SPoC's weighting method, but also showed such methods could be detrimental on scene-heavy datasets. This means that the SumPool descriptor is a strong choice for finding salient features in any spatial location, but can be distracted by outlying features. It is hoped that this experimental study can provide more insight on the importance of selecting the right options in the image retrieval pipeline, including a suitable pooling method for the convolutional information. We conclude that more accurate image retrieval can occur with carefully-selected dimension reduction and pooling strategy after interpreting the context and the nature of the image dataset being used.

Fig. 5. Three query images from Paris6k (top row), and underneath, two images of each query considered 'junk' despite containing the query building.

Acknowledgment

The authors thank National Archives of Australia for allowing this project to access their archival photo collections. This work was supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au). Also, this research was undertaken with the assistance of resources from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 1–9.

Fig. 6. An image from Paris6k (left), followed by the discriminatory strength of the baseline method with (from left to right) 512 dimensions, 32 dimensions, 10 dimensions, and 2 dimensions. Brighter areas of red correspond to higher levels of change to the final descriptor compared to the unchanged image. While reducing the dimensions focusses the discrimination on the desired feature, the retrieval performance actually peaks at 32 dimensions as the descriptor becomes too concise.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1106–1114.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[6] L. Wang, S. Guo, W. Huang, and Y. Qiao, "Places205-vggnet models for scene recognition," *CoRR*, vol. abs/1508.01667, 2015.

[7] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014.

[8] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description, and segmentation," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 65–94, 2016.

[9] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 1717–1724.

[10] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," *CoRR*, vol. abs/1512.04065, 2015.

[11] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky, "Neural codes for image retrieval," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8689. Springer, 2014, pp. 584–599.

[12] A. Babenko and V. S. Lempitsky, "Aggregating local deep features for image retrieval," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 1269–1277.

[13] L. Liu, C. Shen, and A. van den Hengel, "The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 4749–4757.

[14] M. Carvalho, M. Cord, S. E. F. de Avila, N. Thome, and E. Valle, "Deep neural networks under stress," *CoRR*, vol. abs/1605.03498, 2016.

[15] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8695. Springer, 2014, pp. 392–407.

[16] D. S. Touretzky, Ed., *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*. Morgan Kaufmann, 1990.

[17] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computing*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 647–655.

[20] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *CoRR*, vol. abs/1511.05879, 2015.

[21] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual instance retrieval with deep convolutional networks," *CoRR*, vol. abs/1412.6574, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014.

[23] A. Mousavian and J. Kosecka, "Deep convolutional features for image based retrieval and scene categorization," *CoRR*, vol. abs/1509.06033, 2015.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013.

[25] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 36–45.

[26] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I*, ser. Lecture Notes in Computer Science, D. A. Forsyth, P. H. S. Torr, and A. Zisserman, Eds., vol. 5302. Springer, 2008, pp. 304–317.

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[28] J. Philbin O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[29] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7573. Springer, 2012, pp. 774–787.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, K. A. Hua, Y. Rui, R. Steinmetz, A. Hanjalic, A. Natsev, and W. Zhu, Eds. ACM, 2014, pp. 675–678.

[31] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," *CoRR*, vol. abs/1511.07247, 2015.

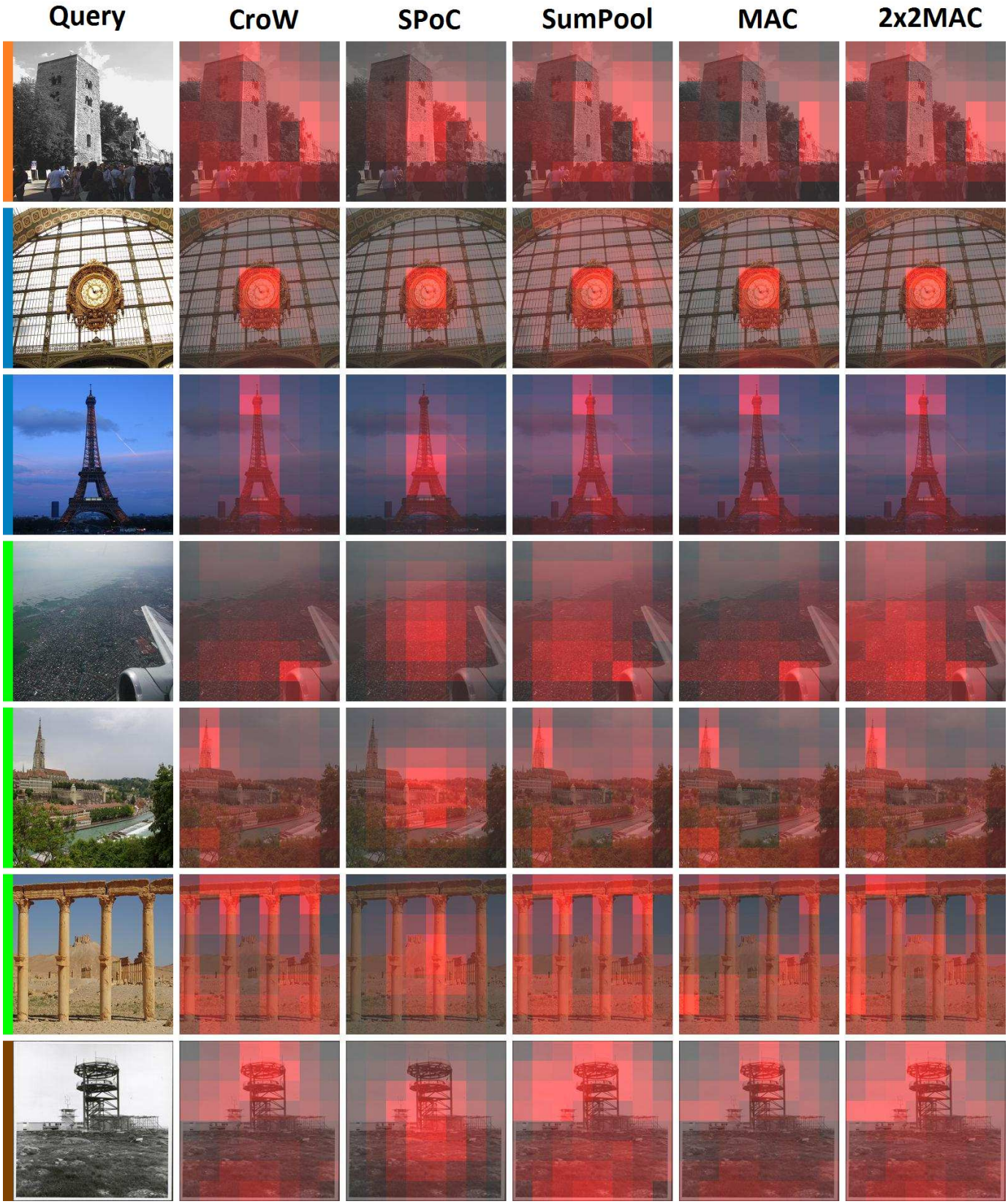

Fig. 7. A selection of images and their discriminative features highlighted. The row beginning with an orange vertical bar is from Oxford5k, blue from Paris6k, green from Holidays, and brown from NAA29k. The rescaled image is shown on the left, followed by its discriminative features in CroW, SPoC, SumPool, MAC, and 2×2MAC on the baseline. Significantly, the SPoC method is biased towards center objects even if there are none, such as in the 'scaffold' (bottom row) and the spire (5^{th} row). However, the other methods are able to discriminate centred objects without *a priori* weighting (clock, 2^{nd} row).