PYRAMID POOLING OF CONVOLUTIONAL FEATURE MAPS FOR IMAGE RETRIEVAL

Abin Jose, Ricard Durall López, Iris Heisterklaus and Mathias Wien

Institut für Nachrichtentechnik, RWTH Aachen University, Aachen, Germany
jose@ient.rwth-aachen.de, ricard.durall@rwth-aachen.de,
heisterklaus@ient.rwth-aachen.de, wien@ient.rwth-aachen.de

ABSTRACT

We propose a novel method for content based image retrieval based on the features extracted from the convolutional layers of the deep neural network architecture. Some of the popular approaches form the feature vectors from the fully connected layers of the convolutional neural networks or directly concatenate the features from the convolutional layers. However, the main problem with the use of feature vectors from fully connected layers is that the spatial information about the objects are lost. This motivated us to use the features from the convolutional layer. We incorporate a pyramid pooling based approach to form more compact and location invariant feature vectors. We have measured the Mean Average Precision (MAP) on benchmark databases such as the Holidays and Oxford5K datasets using features extracted from the AlexNet model. The proposed method gives better retrieval results compared to other state-of-the-art approaches which use feature vectors from fully connected layers and convolutional layers without spatial pooling.

Index Terms— Spatial pyramid pooling, image retrieval, feature extraction, convolutional layer, deep learning.

1. INTRODUCTION

In image retrieval, for forming meaningful feature vectors, it is crucial to have good local image descriptors [1]. The local feature vectors are effectively combined using global representation models such as Bag-of-Words (BOW) [2]. Further improvements in this direction were made by VLAD [3] and Fisher kernel [4] approaches which incorporate higher order statistics. Recently, deep learning has gained popularity for numerous computer vision problems. This is mainly attributed to the availability of fast computational capabilities and enormous amounts of training data.

Even though neural networks are widely used for classification problems, many researchers have utilized the feature extraction capability for image retrieval. An initial attempt in this direction was made by Krizhevsky et al. [5]. Convolutional neural networks (CNNs) have also been used to produce feature vectors suitable for retrieval using the siamese neural network architecture [6].

There have been quite a few approaches using the convolutional layers for forming feature vectors for retrieval. For instance, max-pooling was used in [7] and average-pooling was used in [8]. The hybrid pooling approach [9] combines the average and maximum response from last convolutional layer. Sum-pooling [10] sums the feature activations of the convolutional layer along the height and the width. [11] extracts CNN activation for local patches at different scale levels and then performs orderless VLAD [3]. A major breakthrough in image retreival tasks using CNNs was by Babenko et al. [12]. Using the AlexNet [5] architecture, they formed the feature vectors from convolutional (layer 5) and fully connected layers (layers 6 and 7).

Our approach is mainly motivated by the vizualization attempts by [13] and [14]. [9] have illustrated the semantic information captured by each filter of the final convolutional layer. The final feature maps are having a very small resolution compared to the actual image size. Nevertheless, the feature maps carry spatial signatures in the final convolutional layer. Taking motivation from this, we have designed a feature pooling method using different pyramid levels. This incorporates the spatial information into the feature vectors. Spatial pyramid pooling was used for training a neural network by [15]. However, this method trains a neural network in an end-to-end manner for image classification. Our approach focuses more on effectively pooling the feature maps from the final convolutional layer.

We conduct our experiments using different pyramid combinations composed of windows with different strides and sizes. The MAP values on publically available Holidays [16] and Oxford5K [17] datasets are evaluated and compared with the other state-of-the-art approaches such as neural codes [12] and hybrid pooling [9]. The retrieval performance shows that the proposed method outperforms the retrieval results using feature vectors extracted from layer 5 and from fully connected layers 6 and 7. The reduction in dimensionality is also discussed. The paper is organized as follows. Section 2, discusses the network architecture and the problems associated with the current neural codes. In Section 3 the Spatial pyramid pooling approach is elaborated. Experimental results are summarized in Section 4. Concluding remarks and future research directions are discussed in Section 5.

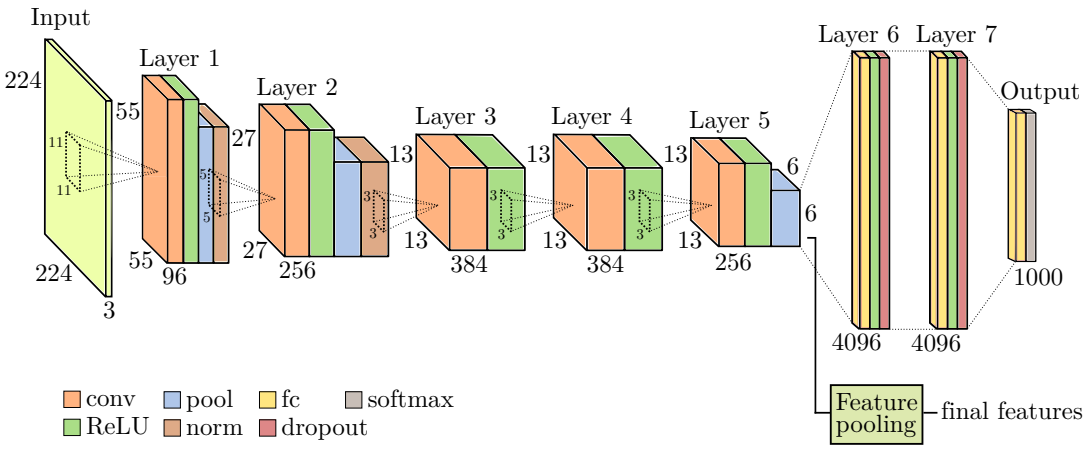

Fig. 1: Overview of the feature pooling using the AlexNet architecture.

2. NETWORK ARCHITECTURE AND FILTER ACTIVATIONS

In content based image retrieval systems, it is vital to have good feature vectors which describe the semantic content of the images. Feature vectors can be extracted from the fully connected layers (layers 6 and 7) or from the convolutional layers (layer 5). The fully connected layers do not carry any spatial information about the scene. So, in order to generate feature vectors with spatial signature, we extract the feature vectors from the final convolutional layer. We have used the AlexNet model (Fig. 1) for extracting the feature vectors. The feature vectors are pooled from the final convolutional layer which is the layer 5.

2.1. Convolutional layer and Feature pooling

The convolutional layer is the feature extracting unit of a neural network. The convolutional layer consists of a set of learnable filters. As a result of the convolution, a two-dimensional activation map (also called feature map) is obtained. Each filter activation carries the spatial signature of the image as well. Fig. 2 shows the filter responses for an example image. Each filter response carries information which highlights different spatial regions in the image. In the AlexNet model, there are 256 filter responses in final convolutional layer, each with a spatial resolution of 6×6 making the total dimension of feature vectors to be $6 \times 6 \times 256 = 9216$. We address mainly two questions in this paper:

1 How to compress the high dimensional feature vectors without loosing the discriminating capability?

2 How to incorporate the spatial information into the feature vectors ?

Question 1 is solved by max-pooling or average-pooling of the feature activations from each of the 256 filters. This also introduces invariance to translation. Max-pooling is robust to scale changes as maximum response of a feature activation will not change with the scale. However, there might be other local maxima which are vital for identifying a scene. Also, max-pooling is not robust to scenes with clutter which produces maximum response at clutter locations rather than the object locations. In such scenarios, average-pooling is more relevant. Average-pooling still averages out the feature activations and does not incorporate the spatial cues.

Question 2 is addressed by using the spatial pyramid pooling method explained in the following section.

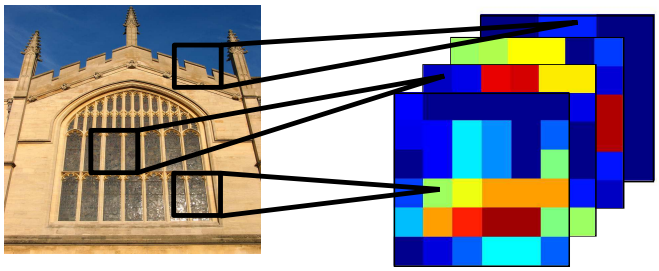

Fig. 2: Input image (left), different activation maps from the convolutional layers (right).

3. SPATIAL PYRAMID POOLING

In order to incorporate the spatial information into the feature vectors, we use the pyramid pooling approach. The feature map is first divided into different spatial bins by using a sliding window with different strides. The maximum activation from each of the spatial bin is then extracted. Different window arrangements are combined together to form a pyramidal representation. The pooled features from all the windows are concatenated together to form the final feature representation. The details of the spatial bins and pyramid pooling are elaborated below.

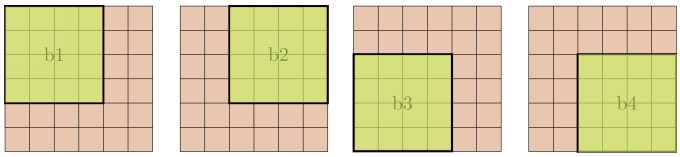

Fig. 3: The 4 different bins b1, b2, b3, and b4 formed for window size of 4×4 with a stride of 2 for a 6×6 activation map.

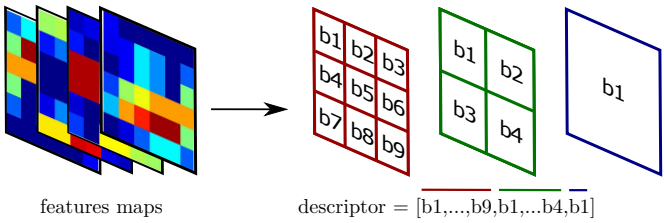

Fig. 4: Illustration of pyramid pooling and formation of final feature vector.

3.1. Spatial bins

First, the 6×6 dimensional feature map is divided into different sub regions called bins. Table 1 summarizes the different window sizes used for forming different bins. The windows are moved in a sliding window manner and the maximum response of the feature map is pooled from each bin. This helps in capturing both the strength of the feature maps and also their spatial positions. For instance, sliding window from "Window2" for a 6×6 feature map is shown in Fig. 3. The window is moved with a stride of 2 resulting in 4 bins b1 to b4. This will generate a feature descriptor which has a dimension of $4 * 256 = 1024$. "Window3" is made of 2 sliding windows calculated independently since the maximum dimension of the feature map is 6.

Table 1: Sliding window sizes and corresponding strides.

	HxW size	Stride
Window1	2x2	2
Window2	4x4	2
Window3	3x6 and 6x3	1

3.2. Pyramid pooling

A typical example of forming a feature vector with pyramid pooling is show in Fig. 4. Since our feature maps have a spatial resolution of only 6×6, we have used small window combinations to form different pyramid structures as given in Table 2. For instance, Pyramid 1 is formed by combining "Window1" and "Window3". "Window1" with a stride of 2 will pool 9 maximum values. Concatenating it for the 256 filters will generate 2304 dimensional vector. Similarly "Window3" will generate 4 maximum values and on concatenating it for 256 filters will generate 1024 dimensional vector. Thus the final feature vector using Pyramid 1 will contain 3328 dimensions. The actual dimension of the feature vector obtained from layer 5 without spatial pooling is 9216. However, with hybrid pooling, the dimension is much lower (512) since the final feature vector is obtained by concatenating the maximum and average values from the 256 activation maps.

The main motivation of using a pyramid structure is that in simple max-pooling operation in layer 5, even though the dimensionality is significantly reduced to 256, the information about the immediate maxima in the adjacent bins are lost. For instance, in Fig. 5 the filter activation which closely represents the tower is shown. Taking a single maximum activation from the filter will not form a meaningful descriptor. However, applying the sliding window based pooling will capture the maximum response in different spatial bins. The activation map carries the spatial signature about the towers. The remaining 255 filters will have different responses which correspond to different regions such as water, sky, trees and ship.

Table 2: Pyramid models. MAX indicates max-pooling applied to the full 6×6 spatial map.

	Layers
Pyramid 1	Window1 + Window3
Pyramid 2	Window2 + Window3
Pyramid 3	MAX + Window3
Pyramid 4	MAX + Window1 + Window2
Pyramid 5	MAX + Window2 + Window3

Fig. 5: Image of a tower and a feature map which gives maximum response at tower locations.

4. EXPERIMENTAL RESULTS

In our experiments, we have used the neural network architecture shown in Fig.1 which is the AlexNet model. The

pretrained model was used and the pooling of feature vectors was done from layer 5 which is the final convolutional layer. We have extracted the feature vectors using 2 different pre-trained models. The first network was trained using the ImageNet [18] database and the second network was trained using Places dataset [19].

4.1. Datasets and experimental setting

Oxford buildings dataset (Oxford5K) [17]: The Oxford Buildings Dataset contains 5062 images from Flickr. There are 11 different landmark images each represented by 5 possible query images making a total of 55 different query images.

INRIA Holidays Dataset (Holidays) [16]: This dataset contains 1491 vacation photographs corresponding to 500 groups based on same scene or object. The images are taken at the same time but with different translation, rotation, and moderate viewpoint changes. First image from each group serves as a query image.

For both the datasets, the performance is measured by the MAP obtained by averaging the Average Precision over all the queries. The query images are always not counted as true positives in the evaluation.

4.2. Retrieval results for network trained on the ImageNet dataset

Table 3: MAP using spatial pyramid pooling methods compared with other state-of-the-art methods for AlexNet model trained on ImageNet dataset.

Descriptor	Dimensions	Holidays	Oxford5K
Neural codes layer 5	9216	0.6828	0.3837
Neural codes layer 6	4096	0.7170	0.4004
Neural codes layer 7	4096	0.7162	0.3650
Hybrid pooling	512	0.7634	-
Pyramid 1	3328	**0.7732**	0.4477
Pyramid 2	2048	0.7693	**0.4889**
Pyramid 3	1280	0.7718	0.4471
Pyramid 4	3584	0.7693	0.4422
Pyramid 5	2304	0.7705	0.4461

The retrieval results for network trained with ImageNet is given in Table 3. For the Holidays database, the pyramid pooling approach gives a better MAP. The MAP for retrieval using feature vectors from layer 5 is only 0.6828. With the proposed pyramid pooling approach, it has increased in the range of 0.7693 to 0.7732. The dimensions of feature vectors using the pyramid pooling approach is also lower compared to the dimensions of neural codes from layer 5 and from fully connected layers 6 and 7. However, the dimension of feature vectors from layer 5 is still lower for the hybrid pooling approach [9] with slightly lower MAP. For the Oxford5K dataset, the MAP values are again higher with pyramid pooling approach. The neural codes from layer 5 gives a MAP of 0.3837 but pyramid pooling improves the result to an average value of 0.4544.

4.3. Retrieval results for network trained on the Places dataset

Table 4 summarize the retrieval performance for pyramid pooling approach using a network trained on the Places dataset. The pyramid pooling approach has slightly lower MAP (with an average value $= 0.75266$) compared to the values obtained using hybrid pooling approach which is around 0.7924. However, the MAP is still better compared to the neural codes from layer 5. For the Oxford5K dataset, the MAP values are higher than the values obtained using simple pooling layers. It is important to note that the retrieval performance here is lower than the values obtained for the network trained with the ImageNet dataset. The main reason for this is that the Oxford5K dataset is a more object-centric dataset. So the ImageNet pre-trained model will give better feature representation.

Table 4: MAP using spatial pyramid pooling methods compared with other state-of-the-art methods for AlexNet model trained on Places dataset.

Descriptor	Dimensions	Holidays	Oxford5K
Neural codes layer 5	9216	0.6771	0.3717
Neural codes layer 6	4096	0.6914	0.3634
Neural codes layer 7	4096	0.6709	0.3482
Hybrid pooling	512	**0.7924**	-
Pyramid 1	3328	0.7543	0.4228
Pyramid 2	2048	0.7523	**0.4289**
Pyramid 3	1280	0.7514	0.4241
Pyramid 4	3584	0.7539	0.4209
Pyramid 5	2304	0.7514	0.4261

5. CONCLUSIONS AND FUTURE WORK

We have proposed a novel method for generating the feature vectors of images from the final convolutional layer by pooling the feature activations from windows of different sizes and strides. This spatial pyramid pooling of feature activations from the convolutional layer helps in capturing the spatial information in the scene. This pooling approach also reduces the dimension of the feature vectors. Our experimental results have shown that this method outperforms other state-of-the-art image retrieval methods which use the features directly by concatenating the feature activations from the convolutional layer. However, when looking at trade off between the descriptor size and performance, hybrid pooling still seems advantageous. Extension of this pooling methods to other deep learning models, and combination of feature vectors from different convolutional layers are some of the future research directions.

6. REFERENCES

[1] David G Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157.

[2] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the International Workshop on Multimedia Information Retrieval*. ACM, 2007, pp. 197–206.

[3] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3304–3311.

[4] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] Abin Jose, Shen Yan, and Iris Heisterklaus, "Binary hashing using siamese neural networks," in *IEEE International Conference on Image Processing ICIP*, 2017.

[7] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016.

[8] Tiancheng Zhi, Ling-Yu Duan, Yitong Wang, and Tiejun Huang, "Two-stage pooling of deep convolutional features for image retrieval," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2465–2469.

[9] Arsalan Mousavian and Jana Kosecka, "Deep convolutional features for image based retrieval and scene categorization," *arXiv preprint arXiv:1509.06033*, 2015.

[10] Artem Babenko and Victor Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1269–1277.

[11] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European Conference on Computer Vision*. Springer, 2014, pp. 392–407.

[12] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural codes for image retrieval," in *European Conference on Computer Vision*. Springer, 2014, pp. 584–599.

[13] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[14] Matthew D. Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[16] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*. Springer, 2008, pp. 304–317.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[19] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva, "Places: An image database for deep scene understanding," *arXiv preprint arXiv:1610.02055*, 2016.