# MULTI-IMAGE AGGREGATION FOR BETTER VISUAL OBJECT RETRIEVAL

*Cai-Zhi Zhu*[†] *Yu-Hui Huang*[⋆] *Shin'ichi Satoh*[†]

[†]National Institute of Informatics, Japan [⋆] RWTH Aachen University, Germany

## ABSTRACT

We study how aggregating multiple images, on query or database side, impacts the performance of visual object retrieval in a Bag-of-Words framework. To this end, we first compare five different multi-image aggregation methods, and suggest selecting the average pooling method in most cases for its superior advantages in accuracy, speed, and memory footprint. Then we prove with experiments that more images generally yield better retrieval performance. What is more, we illustrate that simply aggregating query images without selection is far from optimal. Comprehensive experiments were conducted on three large-scale object retrieval datasets, and the new state-of the-art was achieved. This research can be leveraged in some real applications such as mobile search, where the retrieval performance will be improved once users snap multiple query images.

***Index Terms***— Visual object retrieval, multi-image aggregation, ranking aggregation

## 1. INTRODUCTION

We address how aggregating multiple images, on query or database side, impacts the performance of object retrieval in a Bag-of-Words (BoW) based framework. A related work of how to efficiently aggregate multiple images was recently proposed by Zhu and Satoh [1], in which average pooling was utilized to aggregate BoW vectors of all contained images in each query topic and database video. With this method the best performance was achieved on the TrecVid Instance Search 2011 (abbr. INS2011) challenge [2]. In this literature, we further extend this work by comparing the average pooling method with other four aggregation methods on three large-scale datasets. Similar work was recently done by Arandjelović and Zisserman [3]. They studied how issuing multiple queries significantly improves recall and enables to find quite challenging occurrences of the queried object. Our work differs from theirs in following aspects. First, we study the relationship between the retrieval performance and the number of images contained in query topics or database videos, and reveal that simply aggregating images without selection is far from optimal, which have never been studied before. Second, we propose and test new aggregation methods different from theirs, *e.g.*, the maximum pooling method designed for retrieval. Third, our conclusion of which aggregation method performs the best is different from theirs. We prove our conclusions with comprehensive experiments on three large-scale datasets: the Oxford105K, the INS2011 [2] and the INS2012 [4] datasets. Fourth, we not only focus on how to aggregate multiple query images, but also extend the study of multi-image aggregation to database videos.

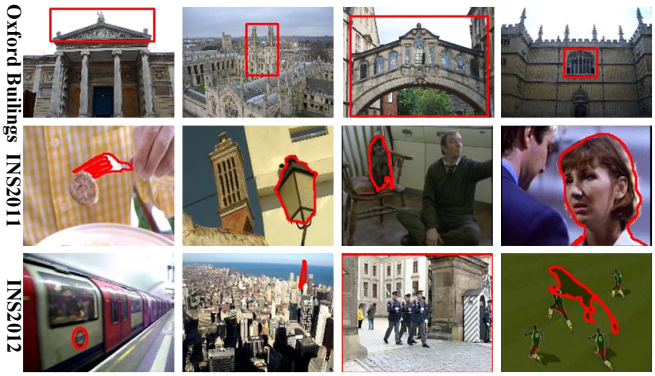



**Fig. 1**. Example query images and ROIs.

The rest of the paper is organized as follows. Section 2 introduces three benchmark datasets and the baseline system. Section 3 describes five methods for aggregating multiple images. In Section 4 and 5 we analyze the problem of aggregating multiple images contained in queries and database videos, respectively, and put forwards our views with experiments. Section 5 concludes this paper.

## 2. DATASETS AND BASELINE

### 2.1. Benchmark Datasets

Three large-scale datasets: the Oxford105K, the INS2011 and the INS2012, are used as benchmarks, with query samples shown in Figure 1, where red shapes overlaid delimit the ROI.

#### *2.1.1. Oxford105K dataset*

The Oxford buildings dataset consists of 5062 high resolution images crawled from Flickr. To test large scale retrieval, another 100k Flickr images are appended as distractor data to form an Oxford105K dataset. For the Oxford dataset, typ-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

ically systems to be evaluated issue a single query at a time, and then the mean Average Precision (mAP) over all 55 query images will be computed [5, 6, 7]. In this research we choose this dataset to test multi-query aggregation. The Oxford dataset provides 5 query images for each 11 landmarks in the Oxford area, and these 5 images share the same ground-truth. As a result, we can group these images together and regard it as a unique query topic corresponding to a specific landmark. Therefore, we are able to test the aggregation of up to 5 query images on these 11 query topics.

#### *2.1.2. TrecVid instances search datasets*

The description of the TrecVid instance search challenge [2, 4] is as follows. Given a set of database videos and a collection of query topics that delimit a person, object, or place, for each query topic, up to the 1000 video clips most likely to contain a recognizable instance of the query topic should be returned. Each query topic consists of a set of query images and associated labeled shapes delimiting the ROI. There are 20,982/76,751 videos and 25/21 query topics in the INS2011/INS2012 datasets, respectively.

### 2.2. Baseline System

A standard BoW retrieval framework [5, 7] is taken as the baseline. We first detect affine-Hessian interest points [8] and extract Root SIFT [7] from each image. Then a large visual vocabulary made up of 1 million visual words will be trained with an efficient approximate k-means algorithm (AKM) [5]. The hard assignment will then be applied and each image will be encoded into a 1 million dimensional term frequency-inverse document frequency (*tf-idf*) vector. Finally the similarity scores of normalized *tf-idf* vectors of items on the query and database sides will be computed for ranking.

Note we choose hard assignment instead of soft assignment [6], as we find that the latter one actually reduces the performance when multiple images are aggregated. To focus on the analysis of the impact of multi-image aggregation and avoid any possible uncertainties, we use a BoW baseline system without reranking, such as spatial reranking [5, 7] and query expansion [9]. Those reranking methods are certainly complementary to our aggregation methods, as they aim at improving the initial ranked accuracy.

## 3. AGGREGATION METHODS

We introduce five methods for aggregating multiple images on the query and database sides.

(1) **Average pooling (Avg-Pooling).** It was first proposed by Zhu and Satoh [1]. They mixed together all SIFT features extracted from multiple images in a query and database item, therefore, each item will be represented by single BoW vector for later scoring. Their method can be regarded as average pooling of BoW vectors of multiple images. It is efficient because the similarity score between each query and database pair will be computed only once, no matter how many images are contained in the query or database item. The effectiveness is also verified on the TrecVid INS2011 dataset with superior performance, while no further experiment was done to compare it with other aggregation methods.

(2) **Maximum pooling (Max-Pooling).** Similar to (1) while take the maximum pooling of BoW vectors of all contained image. This method is inspired by the experience learned from image classification [10], where the maximum pooling generally outperforms the average pooling.

(3) **Average of similarity score (Sim-Avg).** Each contained image will be involved in computing similarity score independently, and the final ranked lists are aggregated by sorting with the average of the scores obtained from each image.

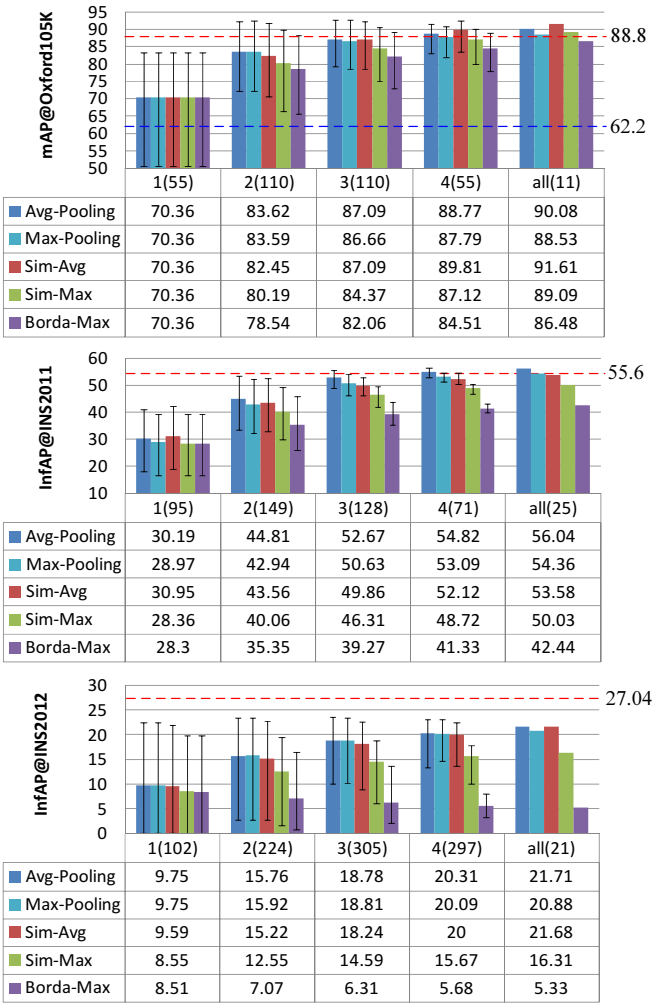
(4) **Maximum of similarity score (Sim-Max).** Similar to (3) while the ranked lists are aggregated by sorting with the maximum of the scores obtained from each image.

(5) **Maximum of Borda count (Borda-Max).** Similar to (4) while the ranked lists are aggregated by sorting with the maximum of the Borda count [11] obtained from each image. This method is the same as sorting with the minimum of the ranks obtained from each image.

Note our work differs from Arandjelović and Zisserman's [3]. We aggregate not only query images but also database images, and also methods themselves are different.

## 4. AGGREGATE QUERY IMAGES

Our experiments on aggregating multiple query images are collected in Figure 2, which is explained as follows. Each time a certain number (denoted as $Q_n$) of images were sampled from each query topic to form a subset, and the retrieval results of all images in each subset will be aggregated. To avoid any possible bias while sampling query images, we enumerated all possible subsets in each query topic. In Figure 2, the numbers out of the parentheses shown in the *x*-axis labels are $Q_n$, and the numbers in the parentheses are the total numbers of subsets given $Q_n$. The *y*-axis indicates the official score: mAP on the Oxford105K and mean inferred average precision (infAP) on the TrecVid datasets, where the mean is taken over all queries, all in percent accuracy. We tested five different sampling cases when $Q_n \in [1, 2, 3, 4, all]$, see the *x*-axis in Figure 2, in which $Q_n = 1$ (the leftmost bar clusters) means querying with single image at a time. It corresponds to the standard query process on the Oxford105K dataset, and $Q_n = all$ (the rightmost bar clusters) means to aggregate all images in each query topic for search, which actually meets the requirement of the TrecVid instance search challenge. Given $Q_n$ and the aggregation method, the computation process for each bar in Figure 2 is as follows. First, the average, the best and the worst accuracy are computed over subsets of each query topic. Then we take the average over




| | 1(55) | 2(110) | 3(110) | 4(55) | all(11) |
|---|---|---|---|---|---|
| Avg-Pooling | 70.36 | 83.62 | 87.09 | 88.77 | 90.08 |
| Max-Pooling | 70.36 | 83.59 | 86.66 | 87.79 | 88.53 |
| Sim-Avg | 70.36 | 82.45 | 87.09 | 89.81 | 91.61 |
| Sim-Max | 70.36 | 80.19 | 84.37 | 87.12 | 89.09 |
| Borda-Max | 70.36 | 78.54 | 82.06 | 84.51 | 86.48 |

| | 1(95) | 2(149) | 3(128) | 4(71) | all(25) |
|---|---|---|---|---|---|
| Avg-Pooling | 30.19 | 44.81 | 52.67 | 54.82 | 56.04 |
| Max-Pooling | 28.97 | 42.94 | 50.63 | 53.09 | 54.36 |
| Sim-Avg | 30.95 | 43.56 | 49.86 | 52.12 | 53.58 |
| Sim-Max | 28.36 | 40.06 | 46.31 | 48.72 | 50.03 |
| Borda-Max | 28.3 | 35.35 | 39.27 | 41.33 | 42.44 |

| | 1(102) | 2(224) | 3(305) | 4(297) | all(21) |
|---|---|---|---|---|---|
| Avg-Pooling | 9.75 | 15.76 | 18.78 | 20.31 | 21.71 |
| Max-Pooling | 9.75 | 15.92 | 18.81 | 20.09 | 20.88 |
| Sim-Avg | 9.59 | 15.22 | 18.24 | 20 | 21.68 |
| Sim-Max | 8.55 | 12.55 | 14.59 | 15.67 | 16.31 |
| Borda-Max | 8.51 | 7.07 | 6.31 | 5.68 | 5.33 |

**Fig. 2**. Experiments on aggregating query images.

the average accuracy of each topic, and we get the bar and corresponding score in the table. Likewise, averaging over the best/worst accuracy we will obtain the error bars. For comparison, the state-of-the-art performance when $Q_n = 1$ and $Q_n = all$ are shown in blue and red dashed lines in Figure 2, respectively. As we can see, the new state-of-the-art performance was achieved on both Oxford105K and INS2011 datasets.

On the TrecVid datasets, two things are worth mentioning: (1) when $Q_n = 1$, we can see that the infAP scores acquired by different aggregation methods are different. That is because aggregation methods also take effect on the test video (see the next section); (2) when the total number of images in a query topic is smaller than $Q_n$, we simply use all query images.

### 4.1. Which aggregation method wins?

From Figure 2, we can see that Avg-Pooling $\approx$ Sim-Avg $>$ Max-Pooling $>$ Sim-Max $>$ Borda-Max on all three datasets

(a) Query 9034 on INS2011, Avg-Pooling 58.57 : Sim-Avg 18.17

(b) Query Balliol on Oxford buildings, Avg-Pooling 93.09 : Sim-Avg 88.23

(c) Query Hertford on Oxford buildings, Avg-Pooling 83.68 : Sim-Avg 95.64

(d) Query 9060 on INS2012 , Avg-Pooling 56.29: Sim-Avg 60.98

**Fig. 3**. Examples when the Avg-Pooling and the Sim-Avg performs differently.

in terms of performance. The Avg-Pooling outperforms the Sim-Avg on the TrecVid datasets, while the Sim-Avg shows its slight advantage on the Oxford105K dataset. The Max-Pooling is slightly inferior to the Avg-Pooling method. These three methods are consistently and remarkably better than the Sim-Max and the Borda-Max. The latter one is the worst in all cases, which means fusing similarity score is more preferable to fusing Borda count directly.

Figure 3 shows some typical examples when the Avg-Pooling and the Sim-Avg perform differently. The Avg-Pooling is usually better than the Sim-Avg when query images are visually similar to each other, for instance, in the case when images captured from different angle of views or in different distances, see Figure 3(a-b). On the contrary, the Sim-Avg outperforms the Avg-Pooling when images are more diverse, see Figure 3(c-d). In Figure 3(c), queries are in different colors, e.g., the second one is black and white, and the fourth one is purple. Figure 3(d) depicts a query of person in different dresses and in different occasions. Therefore we have following observations: the Avg-Pooling is able to aggregate limited diversity of images and enriches the BoW representation of query topics, while if the diversity is too much, the aggregated representation will be distracted.

It is worth mentioning that our conclusions are inconsistent with Arandjelović and Zisserman's [3]. In their work they compared aggregation methods after reranking. We argue that reranking probably introduces additional uncertainties. They claimed that the MQ-Max, which corresponds to the Sim-Max in our case, is the best according to their subjective analysis, while in terms of the retrieval performance they acquired on the Oxford105K dataset, the Joint-Avg (*i.e.*, Avg-Pooling) and MQ-Avg (*i.e.*, Sim-Avg) are remarkably and consistently better than the MQ-Max, which is actually consistent with our experiments.

### 4.2. Performance vs. number of images

In general, the retrieval performance keeps increasing with $Q_n$ for all the aggregation methods. The only exception is the Borda-Max method on the INS2012 (see Figure 2) . We simply ignore it since Borda-Max always performs the worst and degrades dramatically while the dataset is getting harder. Figure 2 illustrates how simply increasing the number of query images will improve performance remarkably. Let's take the Avg-Pooling as example. Comparing $Q_n = 1$ and $Q_n = all$, we can see that the performance is improved by 28%, 86% and 123% on the Oxford105K, INS2011 and INS2012 dataset, respectively. The improvement becomes even more significant while the database is getting harder, *e.g.*, on the INS2012, the infAP score is more than doubled. It is quite impressive since the improvement is acquired without any change on the algorithm. This specially inspires us to circumvent the technical bottleneck of current retrieval algorithms with the following feasible way: simply letting users input multiple query images so as to improve the retrieval performance. This idea is believed to have bright prospects in some applications, for instance, in the case of mobile search, snapping multiple images for search might be accepted by users.

### 4.3. Selective aggregation—future direction

Although simply increasing the quantity of query images will generally improves the retrieval performance, it does not imply that all query images are equally important for retrieval. We observe a big gap between the best and worst cases reflected by the error bars in Figure 2. What's more, the best performance acquired by optimally aggregating fewer query images is even better than simply aggregating all images without selection. The gap is rather notable on the INS2012 dataset. This illustrates that our current aggregation methods without selection are actually not optimal. Therefore, to investigate a selective aggregation method could be worthy of further research.

## 5. AGGREGATE DATABASE IMAGES

Multiple images for database items could be available in some cases. A typical example is to search videos as in the TrecVid instance search challenge. The standard way of video retrieval is first to sample frames from videos, *e.g.*, to detect key frames, or to sample frames at certain rate as what we did in our experiments, and then follow the standard image search framework. In Figure 4, we compared different aggregation methods for searching videos at different sampling rates on the two TrecVid datasets, where all query images in each topic were aggregated. From Figure 4, the following conclusions can be drawn:

(1) Similar to the query side, we get the conclusion that Avg-Pooling $\approx$ Sim-Avg $>$ Max-Pooling $>$ Sim-Max $>$ Borda-Max. In particular, compared with other three methods, the pooling methods regard each video, rather than each frame, as a searching unit, therefore, it has considerable advantage in efficiency. For instance, given $N$ database items with each containing $M$ images, both the time complexity and the memory footprint of the pooling methods are only linear to $N$, while the other three methods are linear to $M*N$. Provided that both query and database items contain multiple images, the pooling methods will be more efficient than the others by several orders of magnitude in both speed and memory. For instance, on the TrecVid datasets, we sampled 3 frames/second and got 100 frames per video on average, in such case the pooling methods are around 100 times faster than others given a query, and the memory footprint is nearly one percent. Therefore, we recommend that in most cases the Avg-Pooling should be used for aggregating multiple images, unless query images look very diverse, as shown in Figure 3(c-d), in such case the Sim-Avg should be used instead. (2) In general, higher frame sampling rate yields better results. If we take the Avg-Pooling as an example, comparing with lowest sampling rate, aggregating more frames at the highest sampling rate will improve the performance by 28% and 26% on the INS2011 and INS2012 datasets, respectively.

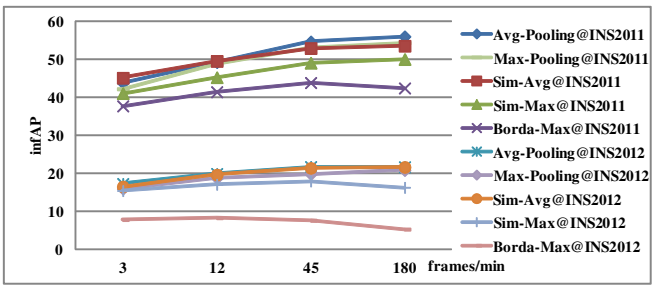



**Fig. 4**. Performance against aggregation methods at different frame sampling rates.

## 6. CONCLUSIONS AND FUTURE WORK

We have investigated how multi-image aggregation would greatly improve the retrieval performance. To this end, we compared five aggregation methods and suggested selecting the Avg-Pooling method in most cases for its good performance and superior advantages in both time complexity and memory footprint. Our further study revealed that a selective aggregation method could be worthy of further research. The new state-of-the-art performance was acquired in our experiments. This research also sheds light on how to effortlessly improve the retrieval performance by asking users input multiple queries, which is believed to be feasible in some real applications such as mobile search.

## 7. REFERENCES


[1] C.-Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," in *ICMR*, 2012.

[2] O. Paul, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, and G. Quéenot, "Trecvid 2011 - an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *TRECVID*, 2011.

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

[4] O. Paul, G. Awad, J. Fiscus, G. Sanders, and B. Shaw, "Trecvid 2012 - an introduction of the goals, tasks, data, evaluation mechanisms and metrics," in *TRECVID*, 2012.

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

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

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

[8] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *IJCV*, vol. 1, pp. 63–86, 2004.

[9] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *ICCV*, 2007.

[10] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[11] J. Aslam and M. Montague, "Models for metasearch," in *SIGIR*, 2001.