

Waseda at TRECVID 2016: Ad-hoc Video Search

Kazuya Ueki, Kotaro Kikuchi, Susumu Saito, and Tetsunori Kobayashi

Faculty of Science and Engineering, Waseda University,
Room 40-701, Waseda-machi 27, Shinjuku-ku, Tokyo, 162-0042 Japan
ueki@pcl.cs.waseda.ac.jp

Abstract. Waseda participated in the TRECVID 2016 Ad-hoc Video Search (AVS) task [1]. For the AVS task, we submitted four manually assisted runs. Our approach used the following processing steps: manually creating several search keywords based on the given query phrase, calculating a score for each concept using visual features, and combining the semantic concepts to obtain the final scores. Our best run achieved a mean Average Precision (mAP) of 17.7%. It was ranked the highest among all the runs submitted.

1 System Description

Our method consists of three steps:

1. Manually select several search keywords based on the given query phrase (Subsection 1.1).
2. Calculate a score for each concept using visual features (Subsection 1.2).
3. Combine the semantic concepts to get the final scores (Subsection 1.3).

1.1 Manual search keyword selection

Given a query phrase, we manually picked out some important keywords. For example, given the query phrase “any type of fountains outdoors”, we extracted the keywords “fountain” and “outdoor”. Here, we explicitly distinguished *and* from *or*; that is, given the query phrase “one or more people walking or bicycling on a bridge during daytime”, we created the new search query “*people*” *and* (“*walking*” *or* “*bicycling*”) *and* “*bridge*” *and* “*daytime*”. In this case, there is no need for a video to include both “walking” and “bicycling”; it is sufficient if one of these is included in the video.

1.2 Score calculation using visual features

In our submission, we extracted visual features from pre-trained convolutional neural networks (CNNs). First, we selected at most 10 frames from each shot at regular intervals, and the corresponding images were input to the CNN to obtain the respective feature vectors from hidden or output layers. These (at most 10) feature vectors were then bound to one feature vector by element-wise max-pooling. We used a total of nine kinds of pre-trained models to calculate scores of concepts as shown in Table 1.

1. TRECVID346

We extracted 1,024-dimensional vectors from pool5 layers of the pre-trained GoogLeNet model [6], which was trained with the ImageNet database. Then we trained support vector machines (SVMs) for each concept using the annotation provided by collaborative annotation [2]. The shot score for each concept was calculated as the distance to the hyperplane in the SVM model.

Table 1. Pre-trained models used in our runs.

Model name	Database	Number of concepts	Concept type(s)
TRECVID346	TRECVID (ImageNet)	346	Object, Scene, Action
PLACES205	Places	205	Scene
PLACES365	Places	365	Scene
HYBRID1183	Places, ImageNet	1,183	Object, Scene
IMAGENET1000	ImageNet	1,000	Object
IMAGENET4437	ImageNet	4,437	Object
IMAGENET8201	ImageNet	8,201	Object
IMAGENET12988	ImageNet	12,988	Object
IMAGENET4000	ImageNet	4,000	Object

2. PLACES205, PLACES365, and HYBRID1183

Because in many queries, concepts that we need to detect comprise not only objects but also scenes, we selected three types of so-called Places-CNNs [7]: the Places205-AlexNet and the Places365-AlexNet models, which were trained on 205 and 365 scene categories with 2.5 and 1.8 million images, respectively; and the Hybrid-AlexNet model, which was trained on 1,183 categories (205 scene categories and 978 object categories) with 3.6 million images. Shot scores were obtained directly from the output layer (before softmax is applied) of the CNNs.

3. IMAGENET1000, IMAGENET4437, IMAGENET8201, IMAGENET12988, and IMAGENET4000

In order to increase the number of object categories, we also used pre-trained ImageNet models. The IMAGENET1000 model was provided by the Caffe development team [3]. This network (AlexNet) was trained with the ImageNet dataset containing 1.2 million images and 1,000 categories and was used in the ILSVRC2012 benchmark. The other networks (GoogLeNet) were provided by the University of Amsterdam [5]. Here too, shot scores were calculated directly from the output layer of the CNNs.

The score for each semantic concept was normalized over all the test shots such that the maximum and the minimum scores were 1.0 (most probable) and 0.0 (least probable), respectively.

If there was no concept name matching a given search keyword, a semantically similar concept was chosen based on word similarity using the word2vec algorithm. If a given search keyword did not have a semantically similar concept, this keyword was not used. If there was more than one model for a given search keyword, we took the average of the scores from multiple models.

1.3 Score fusion

Given the keywords manually selected from a query phrase, we calculated the final scores by score-level fusion. We needed to consider both *and* and *or* operators when we combined several concepts. For the *or* operator, we simply took the maximum score of multiple concepts. For the *and* operator, we employed two methods: summing and multiplying scores. The details are discussed in the next section.

2 Submissions

2.1 Submitted runs

We submitted the following four runs to the TRECVID 2016 Ad-hoc Video Search (AVS) task. The differences between these four runs are the fusion methods used and whether or not fusion weights were applied.

– **Waseda1**

For a query phrase and a test video, the total score was simply calculated by multiplying the scores of the selected concepts:

$$\prod_{i=1}^N s_i, \quad (1)$$

where N is the number of selected concepts and s_i is the normalized concept score. This criterion focuses more on shots that include all the selected concepts. Therefore, shots having all the selected concepts will tend to appear in the higher ranks. On the other hand, if one of the concepts is not correctly detected or the performance of the concept detection model is low, it could have a harmful effect on the final performance.

– **Waseda2**

This run is almost the same as Waseda1 except for the incorporation of a fusion weight. We suppose that a rare keyword is of higher importance than an ordinary keyword. For example, for the query phrase “a man indoors looking at camera where a bookcase is behind him”, it is rare for a “bookcase” to be seen in an image compared to a “man”. For this case, we would like to assign a higher weight to “bookcase”. Therefore, the total score was calculated by

$$\prod_{i=1}^N s_i^{w_i}, \quad (2)$$

where w_i is the fusion weight. We used the IDF values calculated from the Microsoft COCO database [4] as fusion weights.

– **Waseda3**

The total score was calculated by summing the scores of the selected concepts:

$$\sum_{i=1}^N s_i. \quad (3)$$

The final score is calculated under somewhat looser conditions than in the Waseda1 and Waseda2 runs. In this criterion, all the selected concepts are not necessarily included in a shot, but we expect that shots including as many concepts as possible will be found in the higher ranks.

– **Waseda4**

This is similar to Waseda3 except that the fusion weight is used (the same one as in the Waseda2 run):

$$\sum_{i=1}^N w_i \cdot s_i. \quad (4)$$

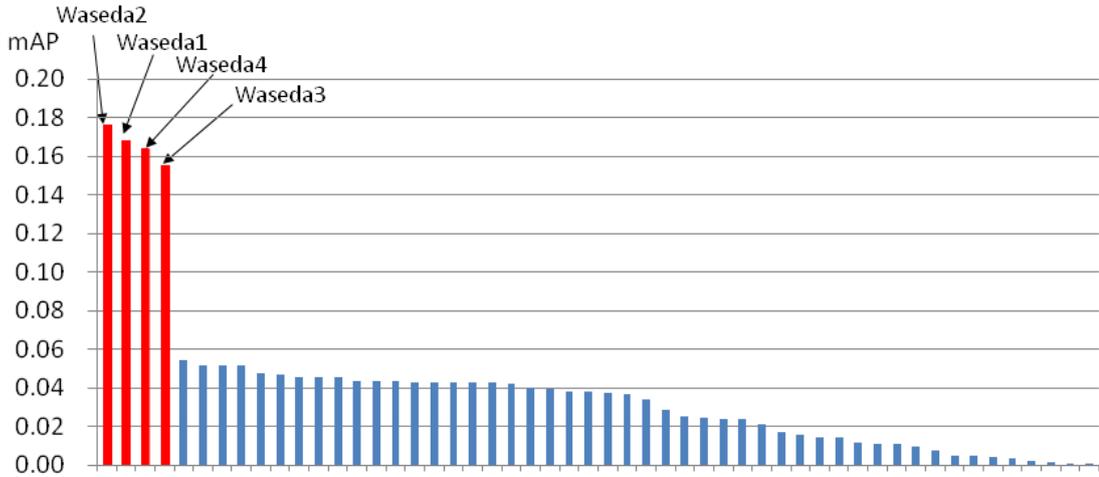


Fig. 1. Comparison of Waseda runs with the runs of other teams on IACC_3.

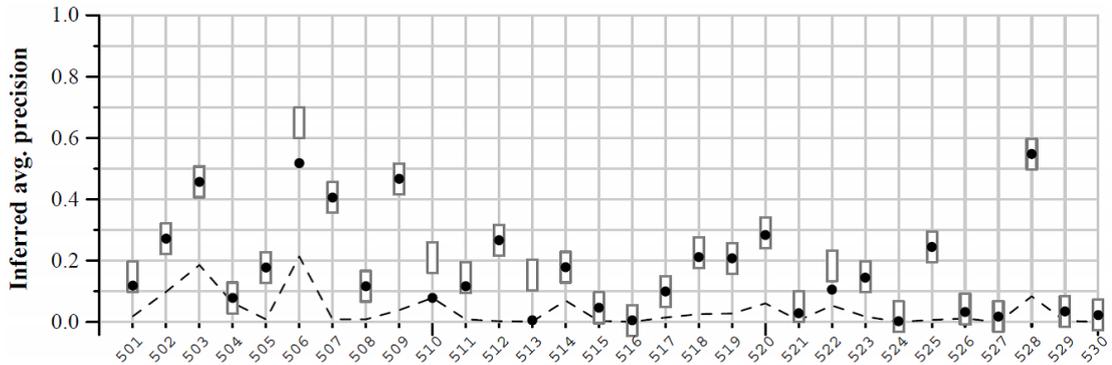


Fig. 2. Average precision of our best run (Waseda2) for each query. Run score (dot), median (dashes), and best (box) by query.

2.2 Results

Fig. 1 shows the results of all runs. The mAPs of our submitted runs are 16.9%, 17.7%, 15.6%, and 16.4%, which ranked in positions 1 through 4 among the 52 runs.

Fig. 2 shows the average precision for each semantic concept. For some concepts, our runs achieved the best average precision. One of the reasons we could achieve the higher performance is that we used a relatively large number of semantic concepts from pre-trained CNNs. On the other hand, although we could cover most objects such as “cup”, “knife”, and “candle”, we could not cover action-related concepts such as “drinking from a cup”, “holding a knife”, or “lighting a candle”. As for the fusion method, a result of the comparison of Waseda1 (or Waseda2) with Waseda3 (or Waseda4) shows that the stricter condition in which all the concepts in a query phrase must be included has the better performance. Additionally, comparing Waseda1 (or Waseda3) with Waseda2 (or Waseda4), we can say that the rarely seen concepts are much more important for the video retrieval task.

3 Conclusion

For this year’s runs, we solved the problem of ad-hoc video search by a combination of many semantic concepts. We achieved the best performance among all the submissions; however, the performance of ad-hoc video search was still relatively low. Our future work will be focused on increasing the number of semantic concepts, especially those related to action. Since we need to obtain good concepts from the query phrase, selecting visually informative keywords and resolving word-sense ambiguities are other important tasks.

In this year’s submission, we focused only on the manually assisted video search. In the future, we also would like to build a fully automatic video retrieval system.

4 Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 15K00249 and Waseda University Grant for Special Research Projects 2016A-026.

References

1. G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F. Smeaton, G. Quénot, M. Eskevich, R. Aly, and R. Ordeman. TRECVID 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings of TRECVID 2016*. NIST, USA, 2016.
2. S. Ayache and G. Quénot. Video corpus annotation using active learning. In *30th European Conference on Information Retrieval (ECIR’08)*, pages 187–198, 2008.
3. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
4. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, Zürich, September 2014.
5. P. Mettes, D. C. Koelma, and C. G. Snoek. The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR ’16*, pages 175–182, New York, NY, USA, 2016. ACM.
6. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.
7. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.