UQMG @ TRECVid 2015: Instance Search

Jiewei Cao, Zi Huang, and Heng Tao Shen The University of Queensland, Australia j.cao3@uq.edu.au, huang@itee.uq.edu.au, shenht@itee.uq.edu.au

Abstract

The UQMG group submits three runs for instance search at TRECVid 2015 [13]: all of them are automatic runs. Instead of adopting the traditional retrieval approach, e.g., Bag-of-Visual-Word (BoVW), our approach consists of three major steps: video decomposition, feature extraction and indexing. During decomposition, video segmentation is applied and various objects are extracted. Here a visual object is a minimal unit, and a video might consists of thousands of objects. Then we extract the visual feature of the object by using a convolutional neural network (ConvNet), which is a high-dimensional vector outputted by a fully connected layer of the network. Finally, the instance search problem is treated as finding the approximate nearest neighbors (ANN) of a given query in a large set of data points in high-dimensional space. Our best mAP is 0.114.

1 Introduction

Video INstance Search (INS) aims to find videos that contain a specific person, or object, or place, by using only a few visual examples of the target. Currently, most approaches tackle the problem in a similar way to the content-based image retrieval (CBIR), which are based on Bag-of-Visual-Word (BoVW) [16][15][12] by extracting visual features from video frames. In this work, we address the problem in a different pipeline: we first decompose the videos into different spatio-temporally coherent regions which correspond to various objects. These decomposed video units will then be processed and indexed for retrieval.

Video segmentation tackles the problem of discovering spatio-temporally consistent segments from videos [7, 6, 5, 10, 9]. After segmentation, each region will be represented by its visual feature. Recent studies show that the generic descriptors of images extracted from the convolutional neural networks (ConvNets) are powerful representations for a wide range of computer vision tasks, such as image classification and retrieval [3, 2, 14, 11]. Therefore, we use the feed-forward neuron activations in the fully-connected layers of a ConvNet as image features, which are high-dimensional vectors. Lastly, we apply approximate nearest neighbors (ANN) techniques, such as locality sensitive hashing (LSH) [1] and k-nearest neighbor graph (K-NNG) [4], for searching when given a query.

2 Proposed Methods

Here are the steps applied in our task submission:

- Firstly, an original video is divided into a number of short video clips (e.g., 30 seconds) before video segmentation.
- 2. We apply hierarchical graph-based video segmentation [7] to discover the spatio-temporal segments from the video clips.
- 3. Since step 2 will generate a large amount of video segments for a single clip, it is impractical to process all the segments. Therefore we filter out segments whose area are smaller than a certain threshold θ_{size} (e.g., 0.01% of the whole frame size).
- 4. An extracted segment is consisted of several image corps across certain frames. They are resized and fed into CaffeNet [8], and we use the outputs of the first fully-connected layer as image features. All the features are then average pooled as the final feature for the segment.
- 5. ANN techniques are adopted to index all the segment features. We have tested both LSH (annoy¹) and K-NNG (KGraph²).
- 6. For online instance searching, given a query, we extract the same ConvNet feature and perform ANN searching. The visually similar segments are returned and ranked based on cosine similarity.

3 Experiments

We submitted three different runs:

- Run 1: default settings for [7], $\theta_{size} = 0.05$, using LSH for ANN.
- Run 2: default settings for [7], $\theta_{size} = 0.05$, using K-NNG for ANN.
- Run 3: default settings for [7], $\theta_{size} = 0.01$, using LSH for ANN.

Both run 1 and 2 have approximately 13 million segments extracted, while there are 34 million in run 3. The mAPs for run 1, 2 and 3 are 0.108, 0.108 and 0.114 respectively.

Since all runs show similar performance, here only the detailed evaluations of run 1 are showed, including the mAP (Fig. 1), precision & recall (Fig. 2) and searching time (Fig. 3). From Fig. 1 we see that the rigid query objects have higher precision scores than non-rigid queries (such as human face, animal). The reason might be ConvNet features are not suitable for those object types. Fig. 3 shows that our system can perform real-time searching.

¹https://github.com/spotify/annoy

²http://www.kgraph.org/



Figure 1: Average precision for different topic queries.



Figure 2: Precision and recall.



Figure 3: Searching time for different topic queries.

4 Conclusion

There are still many rooms for improvement. Firstly, the quality of video segmentation is crucial in our approach. There are still lots of objects can not be segmented correctly. Secondly, better visual features could be applied. In the future, we will investigate conventional features (e.g., SIFT) as well.

Acknowledgements

Programme material ©BBC.

References

- A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun. ACM, 51(1):117–122, 2008.
- [2] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *CoRR*, abs/1406.5774, 2014.
- [3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In ECCV, pages 584–599, 2014.
- [4] W. Dong, C. Moses, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In WWW, pages 577–586, 2011.
- [5] F. Galasso, M. Keuper, T. Brox, and B. Schiele. Spectral graph reduction for efficient image and streaming video segmentation. In *CVPR*, pages 49–56, 2014.
- [6] F. Galasso, N. S. Nagaraja, T. J. Cárdenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, pages 3527–3534, 2013.
- [7] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In CVPR, pages 2141–2148, 2010.
- [8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.
- [9] A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Classifier based graph construction for video segmentation. In CVPR, pages 951–960, 2015.
- [10] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang. Sold: Sub-optimal low-rank decomposition for efficient video segmentation. In *CVPR*, pages 5519–5527, 2015.
- [11] J. Lin, O. Morere, V. Chandrasekhar, A. Veillard, and H. Goh. Deephash: Getting regularization, depth and fine-tuning right. CoRR, abs/1501.04711, 2015.

- [12] M. Murata, H. Nagano, R. Mukai, K. Kashino, and S. Satoh. Bm25 with exponential idf for instance search. *TMM*, 16(6):1690–1699, 2014.
- [13] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Queenot, and R. Ordelman. TRECVID 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2015.
- [14] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574, 2015.
- [15] C.-Z. Zhu, H. Jegou, and S. Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV*, pages 1705–1712, 2013.
- [16] C.-Z. Zhu and S. Satoh. Large vocabulary quantization for searching instances from videos. In *ICMR*, page 52, 2012.