ITEC-UNIKLU
Ad-hoc Video Search Submission 2016

Manfred Jürgen Primus, Bernd Münzer,
Stefan Petscharnig, Klaus Schoeffmann
ITEC - Information Technology, Klagenfurt University
Klagenfurt, Austria
{juergen.primus,bernd,spetsch,ks}@itec.aau.at

October 28, 2016

Abstract

In this report we describe our approach to the fully automatic Ad-hoc video search task for TRECVID 2016. We describe how we obtain training data from the web, create according CNN models for the provided queries and use them to classify keyframes from a custom sub-shot detection method. The resulting classifications are fed into a Lucene index in order to obtain the shots that match the query. We also discuss our results and point out potentials for further improvements.

1 Introduction

The Ad-hoc video search (AVS) task is a new challenge that has been issued for the first time in TRECVID 2016 [1]. It ensues the previous Semantic Indexing task (2010-2015) and models a use-case where an end user searches for specific segments of a video collection that contains arbitrary combinations of persons, objects, activities, locations or similar. The task does not allow any manual intervention but demands a fully automatic approach.

The test data set (IACC.3) contains 4593 Internet Archive videos with durations between 6.5 and 9.5 minutes. The total duration is about 600 hours, the total file size is 144GB.

In the following, we present our approach to the AVS task and discuss the obtained results.

2 Ad-hoc Video Search (AVS) Approach

2.1 Architecture

The overall architecture of our approach is illustrated in Figure 1. The individual steps are described in detail below.

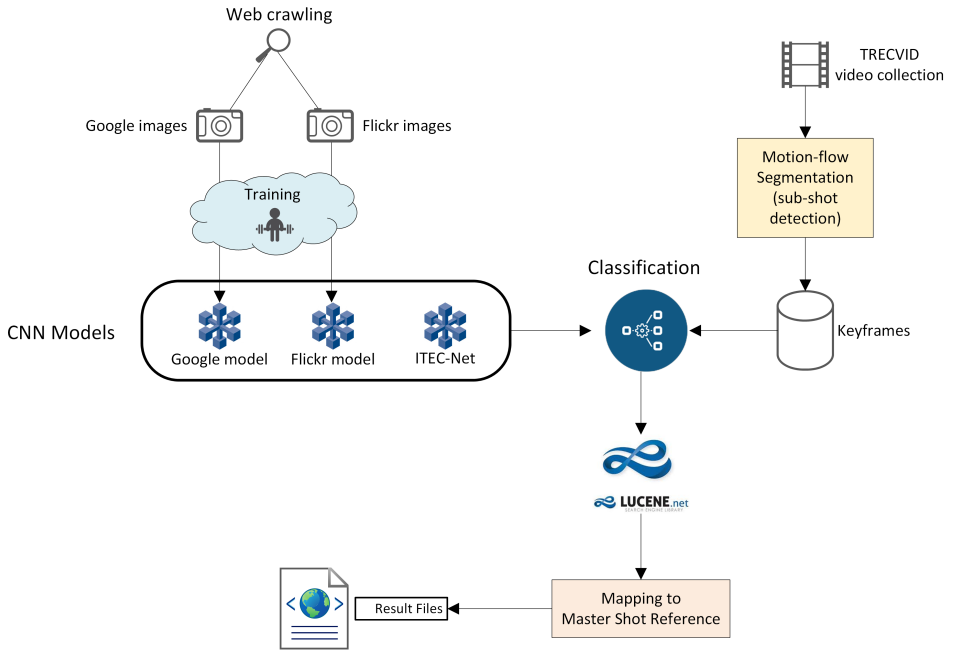


Figure 1: Architecture of our approach

2.2 Training Data and Model Generation

The AVS task does not allow any manual intervention into the whole pipeline of gathering the query results. That also implies that data, which is used for training, should not be adapted in any form except fully automatic ones. This restriction is inspired by the real world scenario, when a user searches for video segments showing certain persons, objects, events, locations, and so forth. That also implies that you will not be able to be prepared for all options. The aim of our approach is to get the requested scenes without the need of any manual intervention.

Collection of training data The core of our search pipeline is a convolutional neural network (CNN). The caffe AlexNet model is used as basis for our classification pipeline. The set of 1000 ImageNet categories provided with the AlexNet model does not fit the given queries sufficiently. For instance, poster, outdoor, George W. Bush, or outdoor are examples for missing categories. There are categories for buildings (church, nursery, prison, etc.) but no special category for destroyed buildings. For this reason, the model has to be retrained. Typically, this is done with a new set of training data and an exchange of the output-layer.

The adaptation of the CNN-model needs sample images that represents categories, which are necessary for the given queries. Our solution is to use the queries to retrieve sample images from different search engines. These images are used to train a convolutional-neural-network that can categorize the scenes respectively keyframes of scenes regarding to the given queries. The images for the training of the CNN-model are gathered using the APIs from Google [1] and Flickr [2]. For each query a set of Google results and a set of Flickr results is collected.

Each of the provided 30 queries is analyzed using the Stanford Part-Of-Speech Tagger (POS-Tagger) [3]. The POS-Tagger opens the possibility to reduce a sentence to a base form consisting of nouns and verbs only. Determiners, prepositions, adverbs, adjectives, etc. are removed. The reduction of the queries increases the amount of the returned images of the search engines. For the training of the CNN-models a maximum of 1000 images per query-class is used. This limit is reached for most of the results provided by Flickr. The results obtained by Google custom search are more scattered. Especially for the Google search the reduction of the query-sentences is important to enhance the count of the results. Figure 2 illustrates the number of images that were found per query.

Flickr returns more than 1000 images in 22 cases, more than 900 images in 3 cases, and it returns less than 800 images in 5 cases. Query 513 "Military personnel interacting with protesters" shows only 146 representative images in Flickr and provides the smallest result set on both search engines.

Google did not reach 1000 images in any result set. Twenty-eight sets of sample images have more than 700 entries. Only query 512 "Palm trees" and query 524 "A man with beard and wearing white robe speaking and gesturing to camera" have about 400 samples each. The mean value of the

[1] https://cse.google.com/cse/
[2] https://www.flickr.com/services/api/
[3] http://nlp.stanford.edu/software/tagger.shtml

Flickr dataset is 907 entries with a standard deviation of 226 samples. The Google dataset has a mean value of 795 entries with a standard deviation of 130 samples.

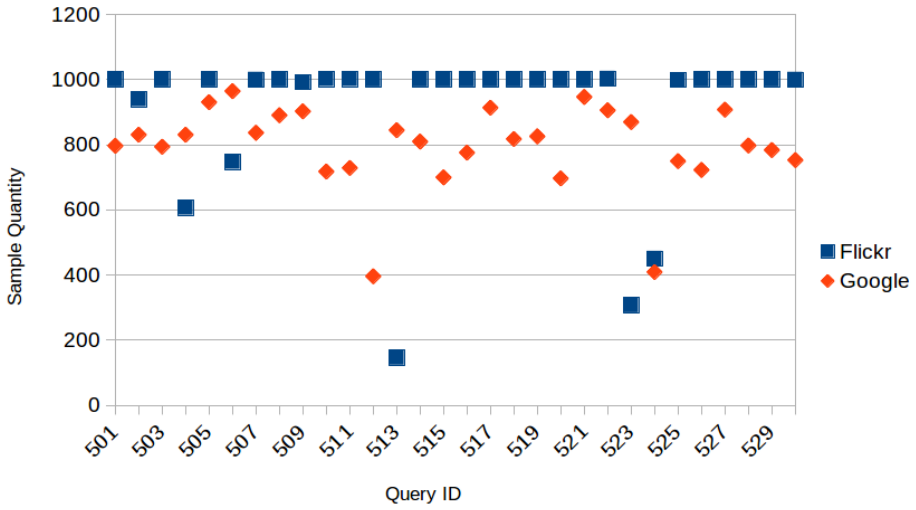


Figure 2: Amount of images per query for Google and Flickr results.

CNN-Model generation In order to carry out the tasks, three different CNN-models are used. Two CNN-models – Flickr model and Google model – are specially designed for the queries of the AVS-task. A third model – ITEC-Net – is an extension of the AlexNet model [4]. AlexNet provides 1000 different classes but lacks some general concepts like flower, people, human, sky, and so forth. The next paragraph describe the generation of these models.

For the training of the Flickr- and the Google-model the images, which has been crawled from the Flickr and Google search engines, are used. Each sample image is labeled with the corresponding query. For the training of both models we used the caffe framework [3] and the AlexNet architecture. For the training the images are re-sized that the smaller side has a size of 256 pixels. Afterwards, the images are cropped with respect to the center resulting in a 256×256 sized image. For the training a random 227×227 patch is extracted. As simple data augmentation method randomly selected input images are mirrored along the horizontal axis. The training is run in batches with 500 training examples in 180 epochs.

For the training of the ITEC-Net model 419,640 images have been manually selected from ImageNet. These images are categorized into 77 additional classes that are used to extend AlexNet. The additional classes describe more general concepts like animal, plant, sport, bird, human, people, sky, and so forth. Some of the new concepts cover also entities searched by the queries like guitar, beach, car, glasses, and so on.

2.3 Motion-flow based video segmentation

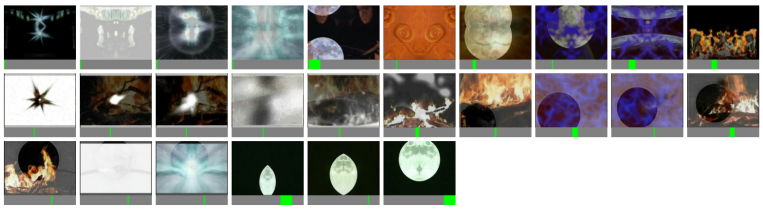
We use a custom video segmentation in addition to the provided master shot reference. The classification operates only on the keyframes of these shots. For the final results, they have to be mapped to the master shot reference in order to produce a valid result.

The main reason for using a different video segmentation method is that the master shot reference has some weaknesses. In many cases, it contains very long shots which in fact show content that is visually quite diverse. In our opinion, it makes sense to use a finer breakdown for analysis. This is even more the case for interactive search. Although we are aware that interactive search is not the focus of the automatic runs in the context of TRECVID, we also use our approach for interactive challenges like the Video Browser Showdown [5].

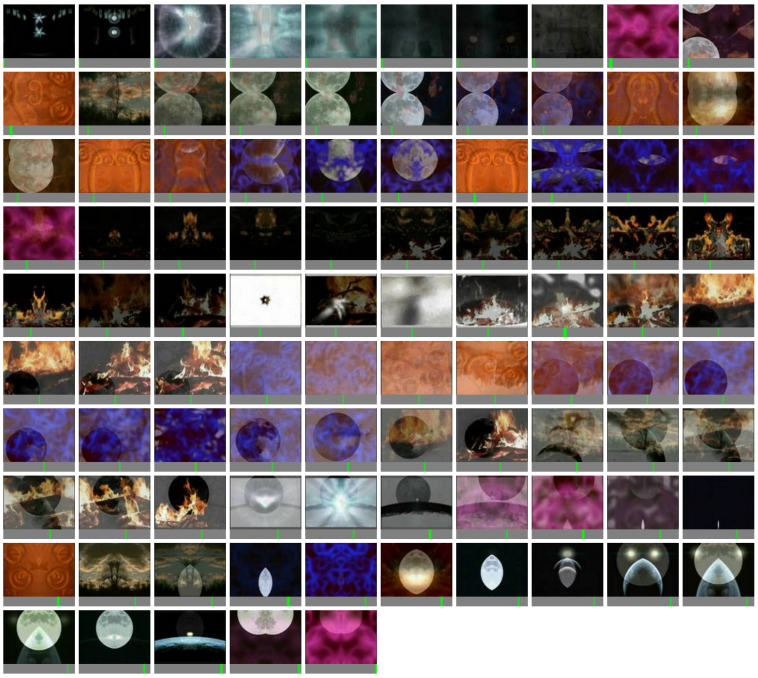
Our algorithm is not only intended to detect segments in the classical sense of shot transition detection. An additional aim of our algorithm is the detection of visual changes. These changes could also happen, if a huge object leaves or enters a scene or due to a pan shot, where the content changes a lot within a classical shot segment. For the detection of these changes our algorithm is based on optical flow tracking. For that purpose we start with an initial set of densely sampled points in the frame and track them with the Kanade-Lucas-Tomasi (KLT) algorithm [2] from one frame to another. As soon as the number of trackable points falls below a specific threshold t_C we detect a shot change and restart the tracking with a fresh set of densely sampled points. For each detected shot the middle frame is selected as keyframe.

Figure 3 illustrates the difference between the master shot reference and our video segmentation for one example video from the TRECVID video collection. The gray bar below each thumbnail represents a timeline where the green section indicates the position and length of the corresponding shot with regard to the entire video. We can see that the master shot reference contains several shots that are quite long, e.g., the last one (74 seconds). In our motion-flow segmentation, this shot is divided into 9 subshots, each of

which with a quite distinct visual appearance. A further example is depicted in Figure 4. While the master shot reference consists of only one shot for the entire video, our segmentation identifies 11 segments, which yields a much more representative model of the video.



(a) Master shot reference

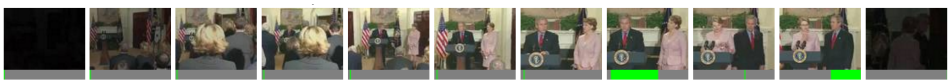


(b) Motion-flow based segmentation

Figure 3: Comparison of master shot reference and our motion-flow based segmentation for video 37617

However, our segmentation method does not always produce a higher number of shots than the master shot reference. We also observed cases where it reduces redundancy without reducing the representativity of the segmentation result. This phenomenon is illustrated in Figure 5. The shots

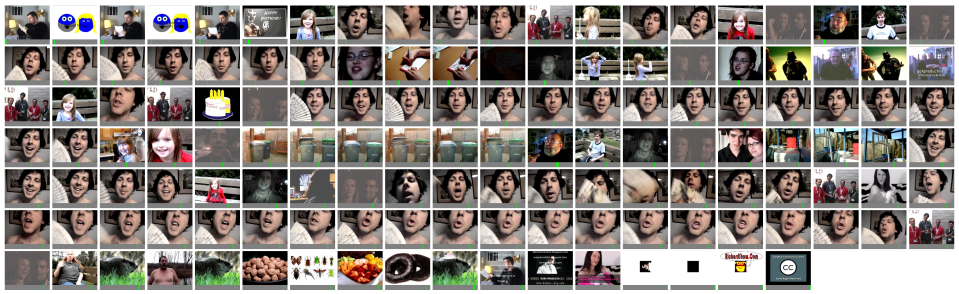
(a) Master shot reference



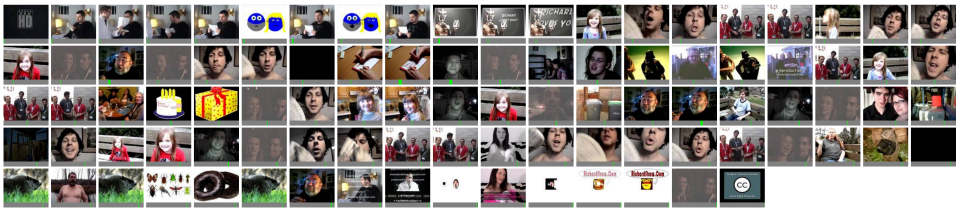
(b) Motion-flow based segmentation

Figure 4: Comparison of master shot reference and our motion-flow based segmentation for video 36086

in the sixth row of the master shot reference have an extremely short duration of 2-4 frames and therefore are represented by very similar keyframes. In our motion-flow based segmentation, the same video section is represented by 3 shots, instead of 20.



(a) Master shot reference



(b) Motion-flow based segmentation

Figure 5: Comparison of master shot reference and our motion-flow based segmentation for video 38573

2.4 Classification and Indexing

Each keyframe of a subshot is classified using the three different CNN-models (ITEC-Net, Flickr-model, Google-model). The models return a set of concepts and the corresponding confidence interval for every keyframe. The concepts are stored as inverted list. Therefore, Lucene [4] is used, which allows a fast and efficient retrieval of the required results.

3 Ad-hoc Video Search Results

We submitted three independent runs with different system configurations. An overview of the runs is given in Table 1 along with the corresponding number of correctly found shots. More details about the performance of the individual runs are illustrated in Figure 6. Configuration 1 obtained the by far best result. As can be seen in the diagram, it especially performed particularly well for query 528 ("Find shots of a person wearing a helmet"). However, the overall results are rather modest. Due to an error in the implementation several shortcomings happened. If a master shot consists of multiple motion-flow segments, it happens that the master shot was referenced multiple times in the result list. One example is shown in Figure 4. Furthermore, the mapping from motion-flow segments to master shot ids is not unambiguous if the shots are not aligned.

Table 1: Overview of submitted runs

Run	**Configuration**	**True shots**
1	ITEC-Net, Flickr-, Google-model	604
2	Flickr model	307
3	Google model	165
Ground Truth		*13257*

A second major problem can be attributed to the semantic quality of the sample images gathered from the Flickr- and Google-platforms. In a qualitative comparison the images in the Flickr-dataset reflect the queries slightly better than the Google samples. Nevertheless, both datasets are very noisy. As solution, each of the dataset has to be processed manually to

[4] http://lucene.apache.org/core/

exclude mismatched examples, which is not allowed due to the regulations of the AVS-task.

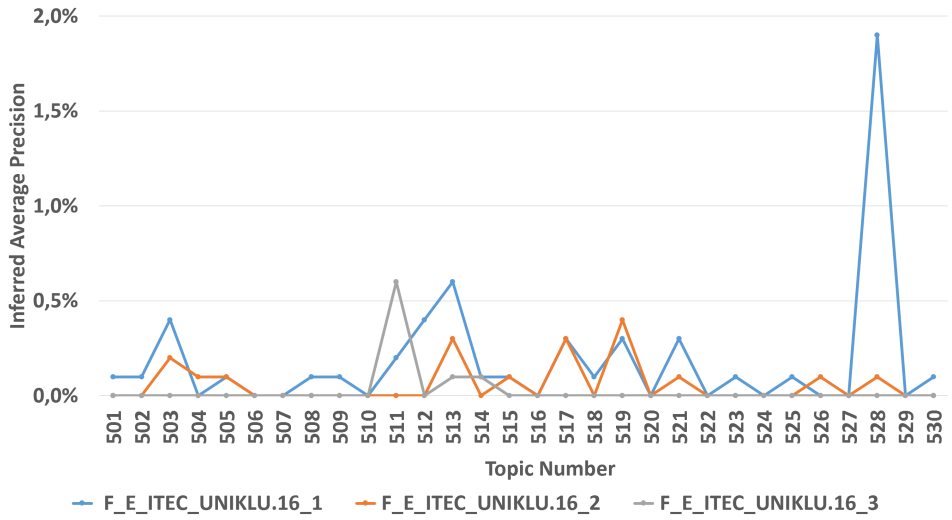


Figure 6: Results of the test runs

4 Conclusion

In this paper we describe our approach to the very challenging Ad-hoc video search task for TRECVID 2016. We experimented with different web sources (Google, Flickr) that were used to train specialized CNN models, but only achieved modest results. However, we observed a considerable improvement by combining these models with a custom CNN model that has been derived from AlexNet. A well-prepared CNN model trained with image samples of high semantic quality that covers a sufficient amount of concepts is difficult to realize. The improvement of the fully automatic pipeline to generate classification models can be a practical way.

References

[1] George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F. Smeaton, Georges Quéenot, Maria Eskevich, Robin Aly, and Roeland Ordelman. Trecvid 2016: Evaluating video search,

video event detection, localization, and hyperlinking. In *Proceedings of TRECVID 2016*. NIST, USA, 2016.

[2] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker - Description of the algorithm. http://robots.stanford.edu/cs223b04/algo_affine_tracking.pdf, 2001.

[3] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] Klaus Schoeffmann, David Ahlström, Werner Bailer, Claudiu Cobârzan, Frank Hopfgartner, Kevin McGuinness, Cathal Gurrin, Christian Frisson, Duy-Dinh Le, Manfred Del Fabro, Hongliang Bai, and Wolfgang Weiss. The video browser showdown: a live evaluation of interactive video search tools. *International Journal of Multimedia Information Retrieval*, 3(2):113–127, 2014.