Technische Universität Chemnitz at TRECVID Instance Search 2015

Marc Ritter[1], Markus Rickert[2], Lokesh Juturu Chenchu[1], Stefan Kahl[2], Robert Herms[2], Hussein Hussein[1], Manuel Heinzig[1], Robert Manthey[1], Daniel Richter[1], Gisela Susanne Bahr[3], and Maximilian Eibl[2]

[1] Junior Professorship Media Computing, Technische Universität Chemnitz, D-09107 Chemnitz, Germany
[2] Chair Media Informatics, Technische Universität Chemnitz, D-09107 Chemnitz, Germany
[3] Florida Institute of Technology, Department of Biomedical Engineering, Melbourne, Florida 32901, USA

Abstract. This contribution presents our second appearance at the TRECVID *Instance Search* task (Over et al., 2015; Smeaton et al., 2006). We participated in the evaluation campaign with four runs (one interactive and three automatic) using audiovisual concepts. A combination of different methods is used in every run. Our basic approach is based on probabilistic assumptions about the occurrences of instances. A deep learning convolutional neural network (CNN) is used in connection with the classification of filming locations and the analysis of audio tracks. The extraction of SIFT features is combined with K-Nearest Neighbors (KNN) clustering and matching to search for instances. In addition, we applied a sequence clustering method that incorporates visual similarity calculations between all corresponding shots in the omnibus episodes provided. Throughout all approaches, we make use of our adaptable and easy-to-use keyframe extraction scheme from the previous evaluation period (Ritter et al., 2014).

1 Structured Abstract

1. *Briefly, list all the different sources of training data used in the creation of your system and its components.*

- For training issues, we solely used the given master shot reference, and the audio and video tracks of the first video with ID 0 from the provided *BBC EastEnders* video footage.

2. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- Within the first interactive run I_E_TUC_1, we are using CNN & visual Bag-of-Word as well as SIFT & KNN based approaches in combination with audio-based indoor/outdoor detection and a probabilistic shot composition (PRNA) that is based on around 1.1 million extracted keyframes and thus shrinks the keyframe pool with respect to this years queries to around 18,000 available frames.

- Our first automatic run F_E_TUC_2 combines CNN & visual Bag-of-Word approaches with audio analysis of the three different classes indoor, outdoor, and crowd & machine.

- The automatic run F_A_TUC_3 combines SIFT features with K-Nearest Neighbors (KNN) matching and deals as a baseline.

- Our last automatic run F_A_TUC_4 combines our approach to partially semantic sequence clustering (SC) as input to the Probabilistic Run-length weighted Neighborhood Algorithm (PRNA) from the previous year that is built on probabilistic assumptions about the occurrences of instances.

3. *What if any significant differences (in terms of what measures) did you find among the runs?*

- We present an adaptable and easy-to-use keyframe extraction scheme in order to reduce the large amount of 42 million frames to 1.1 million keyframes that were used for indexing or instance comparison at I_E_TUC_MI_1.

- As expected, and in terms of MAP, there is a significant difference of more than 13% between the interactive and the best fully automatic run.

Correspondence to: Marc Ritter
marc.ritter@informatik.tu-chemnitz.de

- The results of the run F_A_TUC_4 with SC & PRNA are promising within Precision at rank 30 (P30). Since the sequence clustering did not finish, some optimization potential is left to increase the resulting scores.

4. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

- The reduction scheme of extracting representative keyframes via preprocessing or even SC & PRNA is crucial to an efficient further processing.
- I_E_TUC_MI_1 and F_A_TUC_3 showed reasonable results for topics containing sharp edges using SIFT features.
- The usability of our interactive GUI was significantly improved while allowing to review approximately 3,500 instance candidates on average per topic within the evaluation time frame leading to fast rejections of a large number of false positives.

5. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

- The SC & PRNA method seems to be an usable heuristic for finding a set of new shots containing an instance based on some detected samples in the direct or indirect neighborhood, especially to boost the top 5 result entries at a Precision of almost 40%.
- SIFT features deliver promising results for topics with specific properties.
- An appropriate ranking algorithm needs to be developed in order to create stable results in the first 1,000 appearances above P(30). Additional preliminary tests with similarity measures like PSNR, structured similarity index and histogram correlation indicated insufficient ranking capabilities while being applied to 75 million image patches of the size 48×48 and thus were discontinued. Incorporation with machine learning methods might solve these aspects.

The remainder of the paper is organized as follows: Section 2 provides a general view about the basic concepts and more common components of our system architecture and the underlying workflow for both run types. The specific algorithms that were used within the system, are described in section 3. Remarks regarding the official evaluation results are given in section 4 followed by some conclusions in section 5.

2 System Architecture

The following section describes the overall system architecture and their components as well as the software and toolkits used to accomplish the instance search task. The preprocessing steps and keyframe extraction process applied to the original video footage and sample queries of the topics are discussed in section 2.1. In the section 2.2, the tools used for feature extraction and classification of filming locations based on audio tracks are illustrated. Our approach to deep learning is described in section 2.3. Another methods that are based on SIFT features and the MPEG-7 feature extraction library are described in section 2.4 and 2.5, respectively.

2.1 Preprocessing and Keyframe Extraction

Our different approaches for feature extraction demand an abundant preprocessing on the given data. The underlying video collection from the *BBC EastEnders* series consists of 244 MPEG-4 omnibus video files each containing four episodes of around 30 minutes plus short additional video sequences like advertisements. As the data collection for the task Instance Search (INS) was maintained, we mostly retained the sequence of preprocessing steps described in our report from the previous TRECVID evaluation campaign (Ritter et al., 2014).

We used the already built collection of 471,526 shots according to the given master shot reference table. Due to the anamorphic format provided, we applied deinterlacing routines and a pixel aspect ratio correction to square pixel (resulting in a resolution of $1{,}024\times576$ pixel by utilizing FFMPEG[1]. To further reduce the information that needs to be processed by our image processing chains, we decided to extract representative frames from each shot that we refer to as keyframes, according to our adaptive keyframe extraction scheme from last year; see Figure 2 in (Ritter et al., 2014). By selecting up to five frames per shot, the method is capable of reducing the number of frames from 42 million to 1.15 million. Instead of extracting full size images, we cropped each image at its full resolution by 48 pixels in horizontal and 32 pixels in vertical direction resulting in a resolution of 928×512 pixels. This is expected to reduce or even prevent statistical corruptions in the latter feature extraction processes by black borders or other artifacts at the margins of the pictures.

As the query images and the corresponding masks of the test set were also given with an anamorphic equalization of pixels, we stretched them to squared pixels, too. This results in both query and mask ending up with the same aspect ratio as the index pictures in the corpus. When finished, we process the masks with a customized MATLAB function which delivers the coordinates and size of the bounding box that surrounds the marked white space which denotes the searched object in the full-size query image. As a final step, the coordinates are being mapped to the original picture to provide cut out object patches resulting in query images containing the searched object and a small part of the surrounding environment.

[1] http://www.ffmpeg.org, 06/02/2015

For audio processing, we also used the same collection of audio-only versions of all shots, which were created at sampling rates of 16 kHz mono channel in 16 bit PCM format.

2.2 openSMILE & Weka

The openSMILE feature extraction tool (Eyben et al., 2013) contains general audio signal processing functions which extract several speech- and music-related features. The Low-Level Descriptors (LLDs) as well as the statistical functionals can be calculated with this tool. The LLDs include energy, spectral, cepstral (Mel Frequency Cepstral Coefficients—MFCC) features as well as logarithmic harmonic-to-noise ratio (HNR), spectral harmonicity, and psychoacoustic spectral sharpness. The statistical functionals contain for example means, extremes and percentiles. We used the openSMILE tool to extract large features from audio tracks of sample videos in order to classify the shots according to their filming locations.

The Weka toolkit (Hall et al., 2009) is a machine learning and data mining software which we used for the classification of filming locations based on audio features. Therefore, a series of classifiers that have shown promising results for classification in the literature were selected.

2.3 Deep Learning

Our approach of deep learning is based on Convolutional Neural Networks (CNN) and consists of three main components:

- The generation of training data sets.
- The training of the convolutional neural net.
- The classification of the entire *BBC EastEnders* corpus.

We assembled our training data with Python and FFMPEG scripts using the extracted keyframes mentioned above. The NVIDIA Deep Learning GPU Training System (DIGITS)[2] provided us with a front end for CNN training and quick screening and testing. Based on the deep learning framework *Caffe* (Jia et al., 2014), DIGITS is also implemented in Python. We run the training process on a single Linux machine with one NVIDIA GTX 980 graphics card in GPU mode. We used the same host for the classification process based on a customized *Caffe* implementation. We used *OpenCV* (Bradski, 2000) and a simple *LibSVM* implementation with Python bindings to support the process of image classification with additional *visual Bag-of-Words* (BoW) training.

2.4 OpenCV SIFT & KNN

The approach designed for the automatic run is shown in Figure 1. The basic pipeline from Content-Based Image Retrieval (CBIR), i.e. preprocessing, processing, and postprocessing of the given content is followed to develop this approach. In addition to the aforementioned keyframe extraction process (see section 2.1) from the given video data as preprocessing step, the keyframes are further resized by half into a size of 464×256 pixels on the fly. Afterwards, we extract the widely used Scale Invariant Features Transform (SIFT) features from the resized keyframes and also from bounding box regions of query topics. As the SIFT features are computationally expensive to apply to this very large scale database, the feature extraction from resized keyframes greatly accelerates the extraction process.

Unlike our last year's approach using a database as the main repository for storing features which had significantly helped in performing faster in-database calculations using User Defined Functions (UDF), our approach in this year stores the features and descriptors in normal text files. Because of choosing SIFT features with 128-dimensional descriptions around each keypoint, a simple way is to use open source OpenCV SIFT implementation and its K-Nearest Neighbors (KNN) searching and matching functions that are utilized rather than handling these extracted features in any relational database. The keyframes with their video shot ID numbers and their matching scores for each query topic are stored in text files each containing 1,145,774 entries. In the postprocessing step, the keyframes with high scores are sorted in descending order to retrieve the first most relevant 1,000 video shot IDs.

2.5 MPEG-7 LIRE Library

For the calculation of the MPEG-7 visual descriptors, we used an open source implementation for C# of the LIRE library (Lux and Chatzichristofis, 2008).

3 Methods

Different methods are used this year for the task of instance search. An approach is based on probabilistic assumptions about the occurrences of instances is described in section 3.1. The recognition of filming locations based on classification of audio signals is illustrated in section 3.2. The section 3.3 presents the deep learning approach based on the CNN. The approach based on extraction of SIFT features and KNN is described in section 3.4 and the sequence clustering based on video segmentation in section 3.5.

3.1 Probabilistic Run-length-weighted Neighborhood Algorithm (PRNA)

As already indicated by the name, this approach is rather based on statistics than on image processing. It is built up on two assumptions:

[2] https://developer.nvidia.com/digits, 06/15/2015

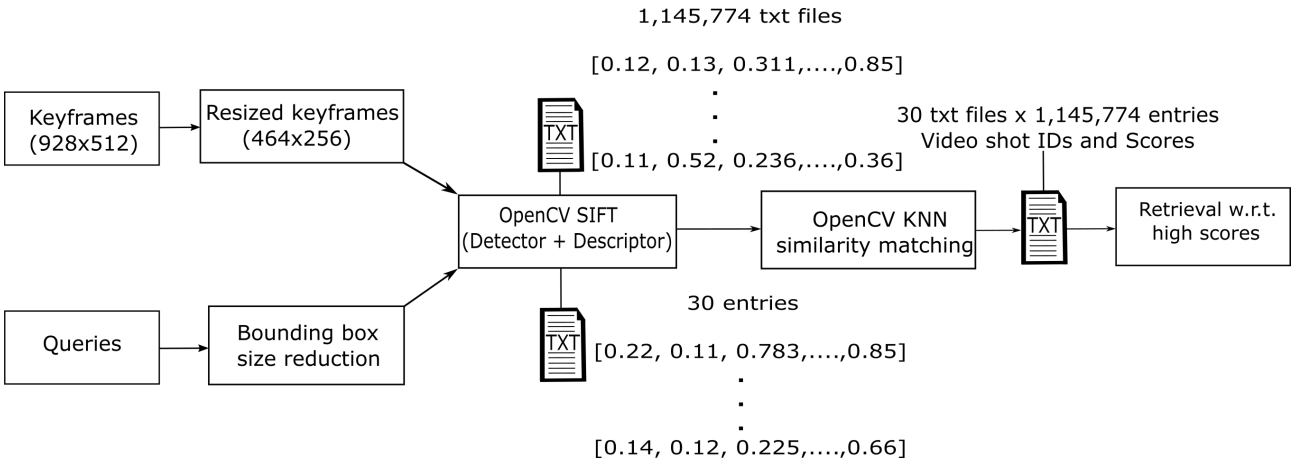

Figure 1. Our workflow for using SIFT and KNN classifier.

1. Longer shots tend to be accompanied by containing a higher probability of a searched instance than shorter shots.

2. Let Π denote the target instance (shot number of the query) of a given sample shot in the test set. Similar object instances are more likely to be contained in the neighborhood Ω around Π, whereas the probability $P \propto \Delta(\Pi,\omega)^{-1}$ decreases while enlarging the distance between Π and a specific location $\omega \in \Omega$.

Hence, this approach takes all shots in the neighborhood around sample shots known to contain a searched instance, weighting them by distance (number of intermediate shots) and run-length to create an ordered list. If there are no available samples in the test set or there are not enough shots connected to the sample shots, the second assumption becomes invalid. Hence, only the first assumption is taken into account by heuristically using a list of the longest shots in descending order to fill up the result lists to 1,000 shots.

3.2 Audio Processing

The recognition of indoor/outdoor scenes is very important in several areas, for example, content based image retrieval and digital photography. The techniques of image processing, such as edge analysis in images, can be used to classify indoor and outdoor images (Payne and Singh, 2005). The acoustic signal classification can also be used to detect indoor and outdoor scenes, since properties of audio signals which are generated during filming differ between indoor/outdoor locations. The audio-based indoor/outdoor detection has been previously used in combination with image processing techniques for instance search (Ritter et al., 2014).

3.2.1 Acoustic Classes

The shots of the development database were categorized to 32 different locations according to filming locations. The locations comprise Albert Bar, Beale Outside, Betting Office, Bridge Street Cafe, Carter Bedroom, Carter Livingroom, Dots Kitchen, Market, Park, Walford East Station Outside, Launderette, Mitchell's Car, etc. There are different numbers of shots in every location in the development database. The minimum and maximum number of shots is 5 and 175, respectively. Several acoustic signals were detected in the audio tracks of sample videos such as speech, music, background music, background noise, baby crying, cutlery, machine, footstep, opening/closing doors, and birds sounds as well as roadway and street noise. The audio-based classification of the 32 locations is difficult, since audio tracks in every location have overlapping sounds. For example, there are speech signals, music and background noise in a bar or cafe. Preliminary detections of single locations did not provide crucial information. Therefore, we regrouped locations according to properties of audio signals into four acoustic classes:

- Crowd: Albert Bar, Bridge Street Cafe,
- Indoor: Betting Office, Carter Bedroom, Carter Livingroom, Dots Kitchen,
- Outdoor: Beale Outside, Market, Park, Walford East Station Outside,
- Machine: Launderette, Mitchell's Car, etc.

The number of shots is in crowd (242), indoor (1,081), outdoor (394), and in machine (118). We used only 115 shots in every class as a balanced data set for training. The audio tracks of sample shot videos were extracted with the WinFF/FFMPEG tool. The audio tracks are downsampled to a sampling rate of 16 kHz and 16 bits per sample.

3.2.2 Feature Extraction and Classification

We assume that the type of speech differ according to a certain location (e.g., shouting in a bar or whispering in a park). Therefore, our approach for location discrimination is rather related to the actor's voices and its paralinguistic features than to background noise. The extracted features on shot-level are the baseline feature set of the Computational Paralinguistics Challenge (ComParE) (Schuller et al., 2013). The feature set contains 6,373 features of LLDs and their statistical functions. The features were extracted using the openSMILE toolkit with the corresponding configuration file.

It is known that feature selection methods can lead to promising results. There are two main approaches: wrappers and filters. Wrapper methods from a machine learning perspective can outperform a classifier by evaluating subsets of features but may lead to overfitting. A filter method in contrast uses a metric to rank features and criteria for the selection. In this work, we prefer the second approach with a discriminant analysis and a ranking by computing the correlated adjusted T-Scores (CAT) (Ahdesmki and Strimmer, 2010) between the group centroids and the pooled mean. Additionally, we assigned a threshold of 50 for the number of the top ranked features in which predominantly MFCC based features are associated with 66%.

Preliminary experiments were conducted by using the prepared development set with the four assigned labels and the proposed feature sets. The goal of the experiments was to determine a suitable feature set with its classifier method. The following machine learning algorithms with default values from the Weka toolkit were applied:

- SMO (support vector machine)
- J48 (decision tree)
- BayesNet (bayes network)
- Random Forest (forest of trees, random inputs)

3.2.3 Audio Classification Results

As a metric we used the unweighted average recall (UAR) in a 10 fold cross-validation as shown in Figure 2. It can be seen that the feature selection method increases the performance for all chosen classifiers. The highest difference could be reached for the BayesNet classifier with 22.2% to a value of 71.3%. However, this result is equal to the SMO without feature selection. The best performance in the experiment was achieved by the SMO including feature selection with an UAR of 76.5%. Finally, this constructed model as well as the selected features were applied on the test set to contribute location discrimination as indicators.

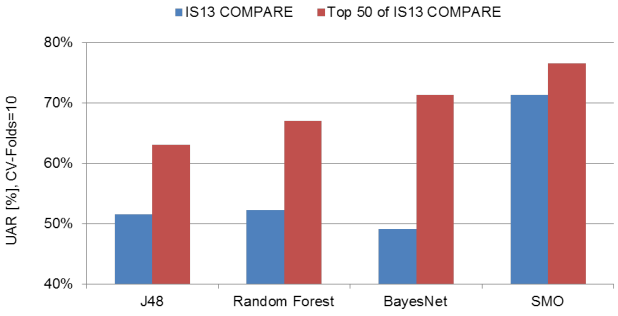

Figure 2. Experimental results of the UAR by applying different machine learning algorithms in connection with the ComParE feature set and the top 50 ranked features.

3.3 CNN & BoW

In the following, the data set and steps for training and classification of instances using available deep learning frameworks are described.

3.3.1 Extending the Data Set

Generating good data sets for CNN training is crucial and massively impacts the outcome and success of image classification. Those sets of training images must cover a wide variety of instances for each class and often consist of hundreds of images. The given set of examples from the instance search task only contained four different images that are insufficient for successful deep learning and classification. We had to find a way to improve the variety of the training data and boost the number of example images without interfering manually. Therefore, we identified the exact frame of the given topic images and extracted additional ten frames before and after the initial occurrence of that topic example image (less if the source shot consists of a small number of frames). We used the given mask files to find the important region in each extracted frame, expanded the mask by 10% and normalized the cropped region to squares of 256×256 pixels. Our final training set contained 30 categories (=30 topics) with a total amount of 1,711 images. During the CNN training process 453 images (20%) were used for validation and 116 images (5%) were used for testing. Although some categories of our training data set contained examples without the relevant instances due to pans and zooms in the original shot (see Figure 3), results of the classification of the example episode 0 were promising, as almost instances from the previous TRECVID evaluation period were contained in the top 5 classification results.

3.3.2 CNN & BoW Training

Our trained image classification model uses the GoogLeNet (Szegedy et al., 2014) approach of network convolutions at a

Figure 3. Our training data set for topic 9133.1 (lava lamp) containing the normalized crop region of the mask of the given example frame as well as the ten leading and trailing frames taken from the same shot.

base learning rate of 0.01. We used the Nesterov accelerated gradient as solver. The trained network showed significant improvements in classification accuracy after the 10th epoch and finished with at least 97% accuracy in class prediction for the top 5 results from the tested images.

We followed a very basic approach for the extraction of visual words from the *EastEnders* episodes. Based on SIFT features (Lowe, 2004), a support vector machine (using the LibSVM library) was trained with 800 visual words separated by *k*-means clustering (MacQueen, 1967).

3.3.3 Classification

We performed the keyframe classification of the *EastEnders* episodes on the whole image as well as on extracted sub-images (patches). For performance reasons, we divided every keyframe into a 3×2 grid resulting in six patches. We did not uses sliding windows to reduce computation time. In order to avoid distortions, all patches were normalized to squares of 256×256 pixels. Every patch of one keyframe was classified with a score for each of the 30 categories. Only the highest rated patch was chosen as the salient part of the keyframe most likely to contain the wanted object instance and representative sub-image for one category. This approach is backed by the visual BoW classification based on SIFT matching performed on the whole source keyframe. Although we only used one class classification with no confidence rating, this approach improved our results especially for object instances with clear outlines. All classification scores were saved in a XML file for further retrieval.

3.3.4 Parsing Results

All of our CNN classification results were stored in XML files in order to add other layers of classification, sorting and rule-based ranking. The most important additional classification approach utilized the acoustic signal analysis. The description of the environment of wanted instances was likely to improve the overall instance search results when combined with the CNN classifications. For that, we retrieved the audio classification of every subtopic from the audio signal analysis and calculated an additional score representing the "noise activity" (e.g. "indoor" classification = low noise activity = low score) in those example shots. This score was later used to determine shots from the entire collection that show similar noise characteristics and are therefore likely to contain an instance (e.g. a lava lamp is more likely to be placed indoors, although image classification could detect otherwise).

We used additional rule-based constraints to reduce the number of false positive detections. For that, we automatically analysed differences in patch scores, the top 3 categories of CNN classification of each shot, average scores of all topics combined and multiple deviations from that. All additional measures were stored in an index, which facilitate the retrieval of varying numbers of final results and, most important, rank those results based on a combination of all measures.

3.4 SIFT & KNN

This approach is based on OpenCV SIFT features and KNN matching functions. SIFT appears as one of the most influential scale invariant features being used in wide variety of publications in instance search tasks during recent years which indeed yielded to promising results. The SIFT features are extracted from 1,145,774 resized keyframes of 464×256 pixels and stored in text files in *OpenCV* file storage format, i.e. in the YAML format structure with the same keyframe names. That means each resized keyframe is represented as a text file with SIFT features. During the extracting of SIFT features using *OpenCV*, there is an opportunity to select the number of keypoints to be detected and described. A partial improvement of the matching accuracy is achieved extracting keypoints without any constraints from the resized keyframes. This feature extraction occupied disk storage space of around 1 TB and consumed around 19 hours of processing time by a workstation PC with a configuration of 16 GB RAM and eight 3 GHz CPU cores where all applications work in parallel utilizing 80 to 90% of its CPU usage.

In the similar way, SIFT features are extracted from bounding box regions of 30 topics from the provided 120 query examples. One example query is randomly selected from the four examples. Once the features are extracted and stored, the 30 queries can be executed at once as a single application against 1,145,774 text files to perform similarity matching. Since this would exceed the available submission period, we had chosen another opportunity in this scenario by executing 15 different queries with 2 topics each in parallel. The extracted SIFT features from the queries are matched against all keyframes features stored in text files. All the text files are read in a sequential order and stored in OpenCV Mat object which facilitates to perform KNN search and matching. KNN search is executed to retrieve the 2 nearest neighbors for each query keypoint in keyframes according to the Euclidean distance metric. In addition to this

matching criteria, Lowe's proposed ratio test (Lowe, 2004) with parameter setting of 0.85 is also applied to prune the ambiguous matched keypoints that are more distant than the mentioned threshold which are referred to as good matches. The matching score is calculated as the ratio between good matches that are left after Lowe's ratio test in comparison to the total number of matching keypoints in a keyframe. This matching score is calculated for each query and all available keyframes. The time taken for similarity matching is around 43 hours of CPU time for 15 applications executed at once. At the end, there are 30 text files in total with 1,145,774 entries in each text file containing scores and keyframe names. These text files are further read to sort the scores in a descending order and retrieving the first 1,000 keyframes and video shot IDs with the highest scores.

3.5 Sequence Clustering

The Sequence Clustering (SC) is based on an alienate video-segmentation approach. It consists of visual similarity calculations between all corresponding shots for each video in the test corpus, followed by sequence clustering. This strategy is led by the idea that many instance search topics are querying for objects which mostly can be found at the same location. For example, the topic 9156 is querying for the symbol of a beer brand, this symbol can probably be found in any sequence located at the pub. The pub has a recognizable combination of colors and textures. Hence, finding all shots with a similar color- and texture-combination raises the probability of finding the beer symbol. In detail, we don't only use a color similarity calculation to find our results. We perform a full scene and location segmentation. This has the advantage to find not only a certain keyframe which is very similar to the topic example, but to return a sequence of continuous shots representing a location. This raises the chance to find the query object even if it is placed in shots with different camera angles.

3.5.1 Workflow

The workflow of the sequence clustering as shown in Figure 4 can be summarized as followed:

1. Selection and extraction of one keyframe per shot (frame in the middle of the shot).
2. Calculation of MPEG-7 (Sikora, 2001) visual descriptors: Color Layout Descriptor (CLD), Edge Histogram Descriptor (EHD) and Scalable Color Descriptor (SCD) for each keyframe and each topic example.
3. Using a *hierarchical agglomerative clustering* (HAC) for each video to find groups of shots with high visual similarity.
4. Clustering sequences on a second level by linking the found similarity groups by their temporal dependencies algorithm.

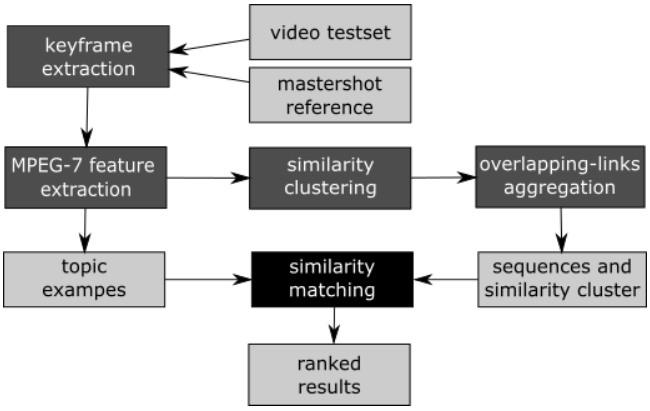

Figure 4. Analysis workflow of the sequence clustering approach.

5. Selecting a keyframe for each similarity group and sequence, plus calculation of MPEG-7 visual descriptors.
6. For each topic:
   - Calculation of all similarity distances between the examples and the similarity-group-keyframes. Returning a list of similarity groups ordered by visual distance.
   - Removing all illegal results, like shots from the test video or shots equal to the topic examples.
   - For each entry in the lists of each example: Insert all shots which are member of the same sequence of results.

3.5.2 Similarity Clustering with HAC

Clustering algorithms can be divided into two strategies: top-down and bottom-up. For example, one of the most used supervised machine learning algorithms in computer vision research is *k*-means clustering. It is a well-known approach and uses a top-down strategy. Its aim is to assign a set of data to a number of predefined clusters and to find an optimal assignment. It has a good performance, but a big disadvantage for the use in video segmentation tasks since the number of clusters has to be already known prior. For our approach we needed to use a more flexible solution, because it is not possible to know how many locations or scenes can be found in a video. Hence, we use a bottom-up strategy by using the HAC. The HAC starts with one cluster per element (shot). In an iterative process, the distances between all clusters are calculated and the closest two clusters are merged into one combined cluster. With each loop the number of clusters is reduced by one until an exit condition is complied. The advantages of the HAC are:

- Customized distance metrics can be implemented.
- Several fusion strategies for the distance calculation of clusters with multiple elements can be selected.

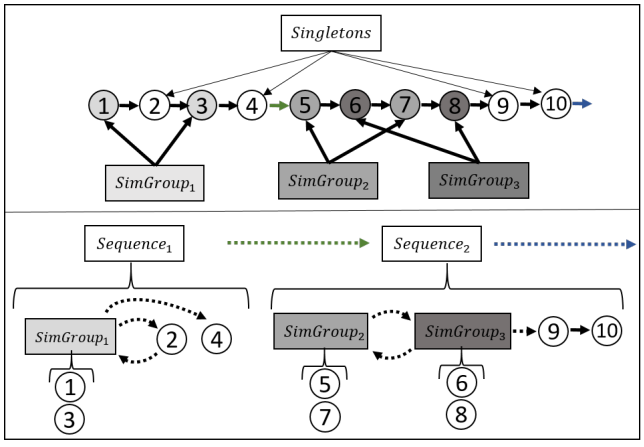

Figure 5. Example for the clustering of shots into sequences.

- The exit condition can be configured freely.

The main disadvantage is the high computational complexity of up to O(n^3). We used the HAC to calculate groups of visual similar shots. This similarity groups represent the reoccuring camera angles and recordings in a similar location. The *BBC EastEnders* show a good example for the shot and reverse-shot technique in film production. The show used in most of the scenes with a low number (in most cases not more than $3-5$) of different camera angles. But these angles are used multiple times. Therefore, each scene consists of a low number of visual similar shot groups. The identification of these similarity groups builds the key element of a scene or sequence detection in a subsequent process (see Figure 5).

For the HAC, a distance metric is needed. In our case, we want to measure the visual distance between two representative keyframes of two shots. There is a large number of available visual features. We focus on the MPEG-7 visual descriptors: CLD, EHD and SCD. For each descriptor, the distance is calculated as defined in the MPEG-7 standard. This results in three distance values. To calculate one single distance metric, we used a linear combination of all three distances as well as a fourth part for the transition distance.

The transition resistance (TR) is defined by the number of shot-boundaries or transitions between two shots. It provides a correction factor to the other visual distances. It reduces the visual similarity depending of the temporal distance of two shots. This reduces the probability of two very similar shots to be merged together, if they are more than a few transitions away from each other. Without this correction step, most clustering analysis would end up with a distribution of clusters unusable for the later step of sequence analysis. In case of the *EastEnders* omnibus episodes with two hours duration, it is very likely that a location occurs several times.

In order to gain reasonable results by the HAC, we needed to perform a parameter optimization on the similarity metrics. The quality of the results depend on the individual weights of the visual descriptors. We use the training video for a parameter optimization. Therefore, the sequences are intellectually annotated as ground truth reference. We found 127 individual sequences in the training video. Based on the ground truth data, we were able to evaluate the individual parameter combinations and count the number of merge errors that occurr during the iterations of the HAC. A merge error can be defined as a merge of two clusters containing elements that belong to separate ground truth sequences. The parameter optimization is performed by the evaluation of the results with different weights of the visual descriptors and the TR. All four factors are used as relative percentage weights with a sum of 100%. The linear combination is given by:

$$a \cdot CLD + b \cdot SCD + c \cdot EHD + d \cdot TR\,, \quad a+b+c+d = 1.0 \qquad (1)$$

We tested all combinations of the factors in steps of 5 percent, which results in 1,680 runs. The finally used parameter set is:

$$0.15 \cdot CLD + 0.55 \cdot SCD + 0.15 \cdot EHD + 0.15 \cdot TR \qquad (2)$$

The evaluation results are a minimal Differential Edit Distance (DED) of 0.4972, while Coverage/Overflow (CO) (see section 3.5.4) are 0.661/0.339 at HAC iteration 1792.

A linkage criterion defines how the distance of two clusters is calculated if the cluster consists of more than one element. Each element is one shot of a video represented as the feature vectors of CLD, SCD, and EHD as well as the number of the shot for the calculation of the transition resistance. If a cluster consists of multiple elements, the distance between two clusters has to be interpolated. The single linkage criterion calculates the distance between each element of cluster A and each element of cluster B and uses the minimal distance between both clusters. This leads in most cases to a long chain of elements, because this strategy tends to build a small number of big clusters. From our experiments we learned that this criterion is not the optimal solution for building similarity groups of shots. On the opposite, the complete linkage criterion takes the maximum element distance for the distance of both clusters. This leads to a better distribution with a larger number of relatively small clusters. A third option could be to use an average distance or to calculate a virtual centroid as a representative for each cluster.

Defining a correct exit criterion for the HAC iteration is a crucial component. It has to match the best segmentation before the HAC clustering algorithm begins to merge shots which are too dissimilar. In other applications of HAC, the exit condition terminates when a certain distance threshold is reached. In our approach this threshold is unknown. Hence, we needed to find a different criterion which provides a good segmentation for the subsequent sequence analysis. It needs a strong differentiation between shots which are not similar. It suffers from weak definite similarity groups, but tolerates a relative high over-segmentation with large numbers of small similarity groups. With our empirical experiences in intellectual sequence segmentation, we concluded that a typical

Condition	Description
Criterion 1	There are more clusters with two elements than singular clusters.
	The number of singular clusters is smaller than the elements of the three largest clusters.
	There are less than 15 singular cluster left.
Criterion 2	The largest cluster size class is less than 10.
	The number of singular clusters is less than 100.
Criterion 3	The number of singular clusters is less than 3.

Table 1. The criteria of the exit condition.

daily soap TV show could use a very small number of single shots where location views or camera angles appear multiple times. Indeed, most shots tend to reoccur between 3 to 6 times but never for more than 10 times. Accordingly, we defined the exit criterion by the distribution of the similarity group sizes or respectively the cluster sizes. To guarantee a sufficient progress in the clustering process, we expect the majority of all clusters to contain more than one element (shot) but less than 10 elements. In addition, the result should provide relatively small groups. We defined that the iteration shall be terminated when the number of elements belonging to the three largest clusters becomes greater than the number of elements in all smaller cluster. The criteria of the exit condition are described in Table 1.

3.5.3 Sequence Analysis

The sequence analysis uses the resulting similarity groups from the HAC. These are mostly representing similar camera angles. It is a common technique in TV production to record a sequence of the plot from multiple fixed camera positions. The cameras are running simultaneously. The final video sequence is a result of the video editing process in the post-production. During this editing the different camera angles are reorganized. Especially in dialog sequences, the different cameras are showing the location from various positions focused in the involved actors. The goal of the sequence analysis is to reconstruct the editing process to find the semantic connections between the camera angles or similarity groups. This makes it possible to determine whether a couple of similarity groups belongs together to form a plot sequence or an action taking place at the same location. Therefore, the algorithm uses the position of the individual shot in the video in combination with the positions of other shots in its similarity group. The following points were considered in the sequence analysis:

1. Beginning with the fist similarity group in the video all shots are ordered by its position in the video.
2. If there are gaps in this sequence, the missing shots are belonging to different similarity groups. This groups have to be member of the same sequence. These groups are clustered together.
3. There is no overlapping in the similarity groups at certain points and there for a number of separate sequences is the result.
4. These sequences are comparable to story sequences showing a continuous action like a dialog or taking place at the same location.

This strategy is inspired by the concept of overlapping links and Scene Transition Graphs (STG) introduced by (Hanjalic et al., 1999) and (Yeung et al., 1998). To evaluate the accuracy of the clustering process, we compared the results with our ground truth data of test videos from the intellectual annotation. The evaluation metrics are described in section 3.5.4. Our implementation provides a result in a hierarchical data structure. Every video file consist of 1,2,...,m sequences. Each sequence consist of 1,2,...,n shots. Each shot can only be connected to one sequence and only one similarity group. At the end, we are able to find the most similar shot in the test set for each example of a topic. For this reason, the data structure delivers all shots in the same similarity group. It delivers the corresponding sequence with all dissimilar shots, which are probably taking place at the same location or belong to the same plot action. We assume, that these resulting lists of shots entail (depending on the selected topic) a higher probability of showing the queried object than other randomly selected shots from the test set.

3.5.4 Evaluation Metrics

Common evaluation metrics at *TRECVID* are Precision and Recall. Unfortunately, these metric proved to be insufficient for the evaluation of our parameter optimization. When it comes to the rating of a segmentation or clustering result, it is hard to decide whether a solution's result is positive or negative. Results are commonly never absolute right nor absolute wrong. Some authors like (Rasheed and Shah, 2003) compensate this by defining gradual measures like a degree of acceptable errors. But in the context of video segmentation there are other more suitable metrics like coverage/overflow or differential edit distance. Both are based on comparing a result with ground truth data measuring the degree of distance between the result and the optimal solution.

The *Coverage/Overflow (CO)* metric was introduced by (Vendrig and Worring, 2002). It consists of two components. For each ground truth sequence the coverage searches the largest overlapping within a calculated sequence and counts the number of correct assigned elements (here shots). The results based on the sequence are aggregated to the arithmetical mean for the complete video in the interval [0,1]. The optimal solution would result in a coverage of 1.0. The overflow counts for each ground truth sequence the number of shots belonging to other sequences, which are mistakenly assigned to this sequence. The result is aggregated over the video. The overflow has the same scope as coverage, but the best result has an overflow of 0.0. Hence the optimal solution would have a coverage/overflow of 1.0/0.0. Generally, it

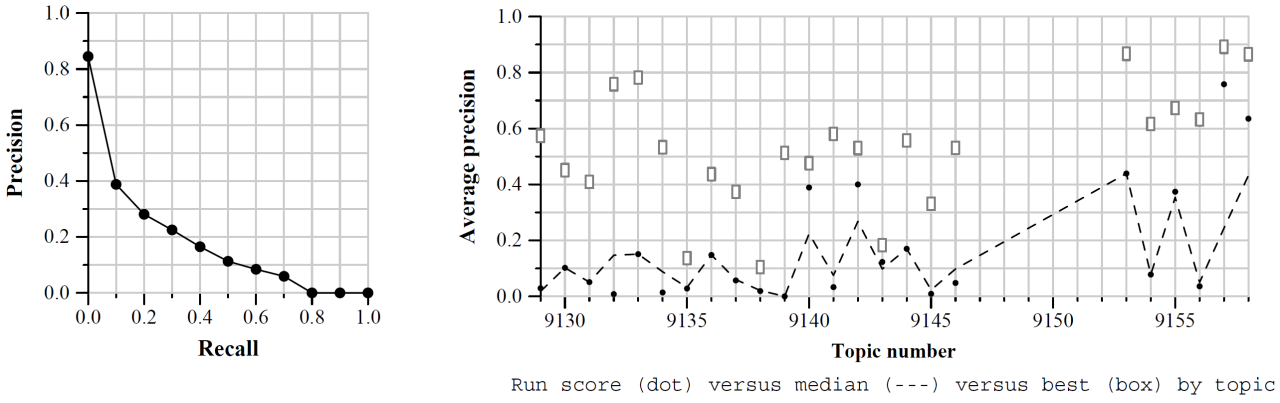

Figure 6. Evaluation result of the first run "I_E_TUC_1".

can be concluded that coverage measures the degree of over-segmentation in a solution. A result of too many small segments results in a bad coverage. The overflow measures the under-segmentation and is sensitive to solutions where too many elements were merged and the found sequences exceed the boundaries of their ground truth equivalent.

The coverage/overflow metric is very useful, but it is not able to determine the accuracy of a result in a single value. An additional metric is defined as DED by (Sidiropoulos et al., 2012). It directly measures the distance between a result candidate and the ground truth reference. It simply counts the number of editing cuts a human editor would have to perform to transform the result into an optimal solution. The DED is measured as a percentage where 100% denotes the number of elements (shots) in the video, because this would be the number of operations to be done in the worst case. Thus, the optimal solution has a DED of 0.

3.5.5 Processing

The calculation is distributed on seven workstation servers due to the computational effort. Each calculation instance works in a virtual machine regardless if it is part of the parameter optimization, descriptor calculation, clustering or similarity comparison. This enabled us to scale up the system performance using a parallel execution with up to 15 parallel instances.

For the evaluation and processing of similarity groups and sequences we use a MS SQL 2012 Database as central data warehouse. This enables us to scale up the processing and distribute it on a server cluster. The database stores and aggregates the following data sets:

- Master shot reference provided by NIST.
- Visual descriptors for each shot.
- Relations between shots and similarity groups.
- Relations between similarity groups and sequences.
- Ground truth sequences data.
- Evaluation presets and results.
- Ordered similarity lists for each topic sample.

4 Results

We participated in four different runs: One interactive and three automatic ones.

4.1 Run 1: Interactive Run

In our first interactive run ("I_E_TUC_1"), we chose a result set from either the combination of "CNN & BoW & Audio" or the combination of "SIFT & KNN" classifiers when the specific topic was presented at the start of each interactive evaluation period. The result sets were evaluated with our graphical evaluation tool (Ritter et al., 2015, 2014; Ritter and Eibl, 2011) which presented up to 4,500 instance candidates per topic. Within the 15 minutes period, we were capable of intellectually examining 3,500 candidates in average grouping them into positive and negative result sets. If necessary, the remaining positive set was taken as input for the instant PRNA algorithm that filled the final results lists up to 1,000 examples.

In 9140, 9153, 9157 and 9158, we identified a high number of true positive matches compared to the whole pool of relevant shots, which leads to a higher recall of objects with sharp edges and SIFT descriptor preprocessing. Comparing the results of our run with the other six interactive runs submitted for the evaluation, with a result of 0.17, we scored a place in the midfield with respect to Mean Average Precision (see Figure 6).

4.2 Run 2: CNN & BoW & Audio

The fully automatic run ("F_E_TUC_2") used the "CNN & BoW & Audio" classifiers. An initial screening with the DIGITS front end revealed very promising results for our deep learning approach. The generated data sets seemed sufficient and examples of wanted instances from episode 0 were regularly ranked among the top 5 results. We assumed, that keyframes, that did not contain any of the given topic objects would score significantly lower relevance ratings. Unfortunately, the final results of *TRECVID* evaluation cam-

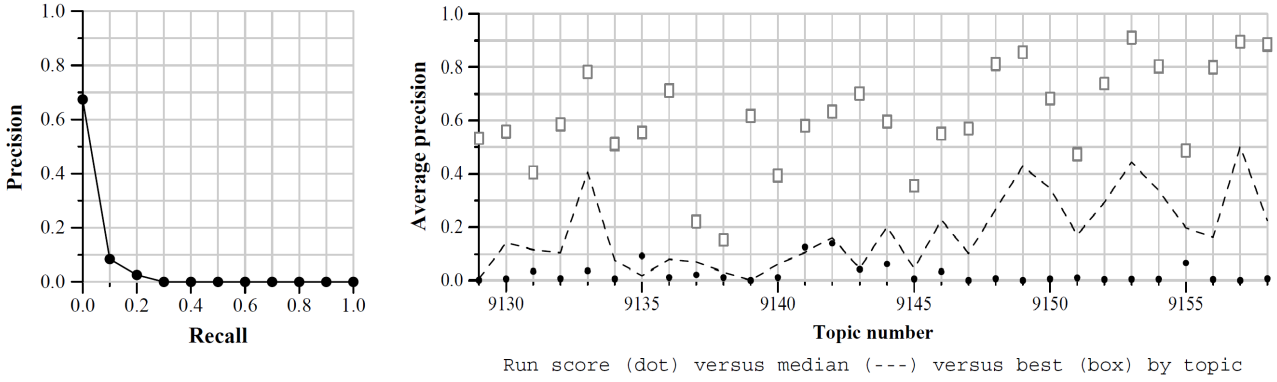

Figure 7. Evaluation result of the fourth run "F_A_TUC_4".

paign, did not confirm this assumption. In addition, the scored ratings for a significantly large number of false detections were often as high as for correct matches.

The best topic is 9157 with 51 matches in the evaluation pool. By comparing the results with the evaluation of our interactive run 1, where the same classifiers were used as a base, we found that our ranking algorithm turned to be insufficient. However, our interactive evaluation showed that we are capable of identifying more true positive hits in larger sets containing multiple thousands of results. As a result, our Mean Average Precision scored very low at 0.004. Hence, we assume that our instance of the DIGITS classification framework using default configuration settings with small amounts of training data appears to be insufficient for large scale data sets, especially when the trained categories do not support object classification for unknown instances ("no class").

4.3 Run 3: SIFT & KNN

Our second fully automatic run ("F_A_TUC_3") uses the "SIFT & KNN" classifiers. We achieved positive hits for half of the topics. Many relevant shots were found in the following three topics: 9157 with 431 hits, 9153 with 231 hits, and 9158 with 182 hits resulting in a low overall Mean Average Precision of 0.023.

4.4 Run 4: SC & PRNA

Another fully automatic run ("F_A_TUC_4") uses the sequence clustering approach. Unfortunately, we were not able to finish the calculation of "F_A_TUC_4" in time forcing us to abort the final similarity parameter optimization at about 60 percent. This led to some topics with less than 1,000 available results. Furthermore, there were even some topics without any results. Therefore, the remaining slots were filled up again by PRNA. The results for topic numbers 9134, 9135, 9137, 9139, 9145, 9146, 9154, 9155 and 9158 were filled up by PRNA, whereas for topic numbers 9136, 9150 and 9153 no results were found and all slots had to be filled by PRNA only with the example shots as input. Figure 7 shows the evaluation results for this run. With a Mean Average Precision of 0.025 for this run, this is our best automatic run this year. We expect a completed parameter optimization to deliver improved results.

5 Conclusions and Future Work

This work introduced our approaches for instance search. We submitted results in four different runs (one interactive and three automatic). In addition to the adaptable keyframe extraction schemes and the PRNA approach which were presented last year, different methods are developed or rather combined to achieve the task of instance search in this year.

Our fully automated shot oriented approach (SC & PRNA) performed slightly better than our instance-based approaches (CNN and SIFT). While we were able to identify relevant shots for almost all topics, the CNN and SIFT methods did not succeed in 20 to 50 percent of the topics. However, there are some topics with respectable results using SIFT classifiers (9140, 9153, 9157, and 9158) for structured objects with sharp edges. A remaining challenge is to create a reasonable ranking algorithm that incorporates multiple features.

The results of our probabilistic approach (PRNA) showed, that the assumptions concerning the spatio-temporal connection between the shots and the corresponding occurrences of searched instances are valid for the evaluated data set. Especially in our interactive run, we recorded a massive improvement in the recall, when there were only a few but further distributed identified instances. The application of the sequence clustering (SC) algorithm as a preprocessing step significantly improved the results. However, it is likely to achieve further improvements by including additional features beyond *MPEG-7*. Especially, centroid linkage and average linkage are supposed to be promising strategies. A solution of an audio-based shot boundary detection could be a useful extension, as well as a face detection for the main characters and supporting actors. But the most important task is to extend this approach to construct a preprocessing component to support the instance search solutions.

We entered the domain of deep learning strategies for instance search for the first time this year. Currently available frameworks like DIGITS and *Caffe* provided us with the tools needed for the training of convolutional neural networks. We built our deep learning workflow from the scratch with focus on the instance search task. Large scale classification with neural networks is a demanding challenge and results are largely dependent on the quality of training data sets as well as the selection of distinct categories for the classification process. Preprocessing and enhancement of data for those training sets can have significant impact on the overall quality of the classification. Postprocessing of classification results and the combination of those results with additional measures might help us to reduce the number of false positive detections in the future, which is important for the classification of mass data. Utilizing different models of neural networks for specific purposes (e.g. the detection of people, machines or locations) and additional high-level information (e.g. shot compositions or global visual features) might help to further improve the search for certain classes of objects besides the search for specific instances of objects.

The matching and retrieval in the "SIFT & KNN" classifiers is based on SIFT features matching and Lowe's ratio test without any spatial verification. To further improve the matching accuracy, carefully chosen spatial verification strategies with less computationally expensive approaches could be integrated into the matching process. By using efficient re-ranking techniques, the retrieval rate could also be improved.

Acknowledgements. This work was partially funded by the program of Entrepreneurial Regions InnoProfile-Transfer in the project group localizeIT (funding code 03IPT608X). Programme material is copyrighted by BBC.

References

Ahdesmki, M. and Strimmer, K.: Feature selection in omics prediction problems using cat scores and false non-discovery rate control, Annals of Applied Statistics, Vol. 4 (1), 503–519, 2010.

Bradski, G.: The OpenCV Library, Dr. Dobb's Journal of Software Tools, 25, 120–126, 2000.

Eyben, F., Weninger, F., Gross, F., and Schuller, B.: Recent Developments in openSMILE, the Munich Open-source Multimedia Feature Extractor, in: Proceedings of the 21st ACM International Conference on Multimedia, MM '13, pp. 835–838, ACM, New York, NY, USA, 2013.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H.: The WEKA Data Mining Software: An Update, SIGKDD Explor. Newsl., 11, 10–18, 2009.

Hanjalic, A., Lagendijk, R. L., and Biemond, J.: Automated High-level Movie Segmentation for Advanced Video-retrieval Systems, IEEE Transactions Circuits and Systems for Video Technology for Video Technol., 9, 580–588, 1999.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093, 2014.

Lowe, D. G.: Distinctive image features from scale-invariant keypoints, International journal of computer vision, 60, 91–110, 2004.

Lux, M. and Chatzichristofis, S. A.: Lire: Lucene Image Retrieval: An Extensible Java CBIR Library, in: Proceedings of the 16th ACM International Conference on Multimedia, MM '08, pp. 1085–1088, ACM, New York, NY, USA, doi:10.1145/1459359.1459577, 2008.

MacQueen, J.: Some Methods for Classification and Analysis of MultiVariate Observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, edited by Cam, L. M. L. and Neyman, J., vol. 1, pp. 281–297, University of California Press, Berkeley, Calif., 1967.

Over, P., Awad, G., Michel, M., Fiscus, J., Kraaij, W., Smeaton, A. F., Quenot, G., and Ordelman, R.: TRECVID 2015 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics, in: Proceedings of TRECVID 2015, NIST, USA, 2015.

Payne, A. and Singh, S.: Indoor vs. Outdoor Scene Classification in Digital Photographs, Pattern Recognition, 38, 1533–1545, 2005.

Rasheed, Z. and Shah, M.: A Graph Theoretic Approach for Scene Detection in Produced Videos, Multimedia Information Retrieval Workshop, 2003.

Ritter, M. and Eibl, M.: An Extensible Tool for the Annotation of Videos Using Segmentation and Tracking, in: Proceedings of DUXU at HCI International, pp. 295–304, 2011.

Ritter, M., Heinzig, M., Herms, R., Kahl, S., Richter, D., Manthey, R., and Eibl, M.: Technische Universität Chemnitz at TRECVID Instance Search 2014, in: Proceedings of TRECVID Workshop, Orlando, Florida, USA, 2014.

Ritter, M., Storz, M., Heinzig, M., and Eibl, M.: Rapid Model-Driven Annotation and Evaluation for Object Detection in Videos, in: UAHCI at HCI International, vol. 9175 of *LNCS*, pp. 464–474, 2015.

Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., and Kim, S.: The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism, in: Proceedings of Interspeech, Lyon, France, 2013.

Sidiropoulos, P., Mezaris, V., and Kompatsiaris, I.: Differential edit distance as a countermeasure to video scene ambiguity, in: IEEE International Workshop on Machine Learning for Signal Processing, MLSP, pp. 1–6, 2012.

Sikora, T.: The MPEG-7 Visual Standard for Content Description—An Overview, IEEE Transactions on Circuits and Systems for Video Technology, 11, 696–702, 2001.

Smeaton, A. F., Over, P., and Kraaij, W.: Evaluation campaigns and TRECVid, in: MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 321–330, 2006.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: Going deeper with convolutions, arXiv preprint arXiv:1409.4842, 2014.

Vendrig, J. and Worring, M.: Systematic Evaluation of Logical Story Unit Segmentation, IEEE Transactions on Multimedia, 4, 492–499, 2002.

Yeung, M., Yeo, B.-L., and Liu, B.: Segmentation of Video by Clustering and Graph Analysis, Computer vision and image understanding, 71, 94–109, 1998.