Supervised Learning Approaches for Automatic Structuring of Videos

Danila Potapov

UNIVERSITÉ GRENOBLE ALPES

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Danila POTAPOV

Thèse dirigée par **Cordelia SCHMID** et
codirigée par **Zaid HARCHAOUI**

préparée au sein de **Inria Grenoble Rhône-Alpes**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Supervised Learning Approaches for Automatic Structuring of Videos

Thèse soutenue publiquement le **« 22 Juillet 2015 »**,
devant le jury composé de :

Prof. Cordelia SCHMID
Inria Grenoble Rhône-Alpes, France, Directeur de thèse
Dr. Zaid HARCHAOUI
Inria Grenoble Rhône-Alpes, France, Co-encadrant de thèse
Prof. Patrick PEREZ
Technicolor Rennes, France, Rapporteur
Prof. Ivan LAPTEV
Inria Paris Rocquencourt, France, Rapporteur, Président
Dr. Florent PERRONNIN
Facebook AI Research, Paris, France, Examinateur
Dr. Matthijs DOUZE
Inria Grenoble Rhône-Alpes, France, Examinateur

Université Grenoble Alpes

Doctoral Thesis

Supervised Learning Approaches for Automatic Structuring of Videos

Author:
Danila Potapov

Supervisors:
Zaid Harchaoui
Cordelia Schmid

Inria Grenoble
Ecole doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

October 2015

Abstract

Automatic interpretation and understanding of videos still remains at the frontier of computer vision. The core challenge is to lift the expressive power of the current visual features (as well as features from other modalities, such as audio or text) to be able to automatically recognize typical video sections, with low temporal saliency yet high semantic expression. Examples of such long events include video sections where someone is fishing (TRECVID Multimedia Event Detection), or where the hero argues with a villain in a Hollywood action movie (Action Movie Franchises). In this manuscript, we present several contributions towards this goal, focusing on three video analysis tasks: summarization, classification, localization.

First, we propose an automatic video summarization method, yielding a short and highly informative video summary of potentially long videos, tailored for specified categories of videos. We also introduce a new dataset for evaluation of video summarization methods, called MED-Summaries, which contains complete importance-scoring annotations of the videos, along with a complete set of evaluation tools.

Second, we introduce a new dataset, called Action Movie Franchises, consisting of long movies, and annotated with non-exclusive semantic categories (called beat-categories), whose definition is broad enough to cover most of the movie footage. Categories such as "pursuit" or "romance" in action movies are examples of beat-categories. We propose an approach for localizing beat-events based on classifying shots into beat-categories and learning the temporal constraints between shots.

Third, we overview the Inria event classification system developed within the TRECVID Multimedia Event Detection competition and highlight the contributions made during the work on this thesis from 2011 to 2014.

Keywords: video analysis, video classification, video summarization, computer vision, machine learning

Resumé

L'Interprétation automatique de vidéos est un horizon qui demeure difficile à atteindre en utilisant les approches actuelles de vision par ordinateur. Une des principales difficultés est d'aller au-delà des descripteurs visuels actuels (de même que pour les autres modalités, audio, textuelle, etc) pour pouvoir mettre en oeuvre des algorithmes qui permettraient de reconnaitre automatiquement des sections de vidéos, potentiellement longues, dont le contenu appartient à une certaine catégorie définie de manière sémantique. Un exemple d'une telle section de vidéo serait une séquence où une personne serait en train de pêcher; un autre exemple serait une dispute entre le héros et le méchant dans un film d'action hollywoodien. Dans ce manuscrit, nous présentons plusieurs contributions qui vont dans le sens de cet objectif ambitieux, en nous concentrant sur trois tâches d'analyse de vidéos: le résumé automatique, la classification, la localisation temporelle.

Tout d'abord, nous introduisons une approche pour le résumé automatique de vidéos, qui fournit un résumé de courte durée et informatif de vidéos pouvant être très longues, résumé qui est de plus adapté à la catégorie de vidéos considérée. Nous introduisons également une nouvelle base de vidéos pour l'évaluation de méthodes de résumé automatique, appelé MED-Summaries, où chaque plan est annoté avec un score d'importance, ainsi qu'un ensemble de programmes informatiques pour le calcul des métriques d'évaluation.

Deuxièmement, nous introduisons une nouvelle base de films de cinéma annotés, appelée Action Movie Franchises, constitué de films d'action hollywoodiens, dont les plans sont annotés suivant des catégories sémantiques non-exclusives, dont la définition est suffisamment large pour couvrir l'ensemble du film. Un exemple de catégorie est "course-poursuite"; un autre exemple est "scène sentimentale". Nous proposons une approche pour localiser les sections de vidéos appartenant à chaque catégorie et apprendre les dépendances temporelles entre les occurrences de chaque catégorie.

Troisièmement, nous décrivons les différentes versions du système développé pour la compétition de détection d'événement vidéo *TRECVID Multimédia Event Detection*, entre 2011 et 2014, en soulignant les composantes du système dont l'auteur du manuscrit était responsable.

Mots-clés: analyse de vidéos, classification de vidéos, résumé automatique de vidéos, vision par ordinateur, apprentissage statistique

Acknowledgements

It was an exceptional chance to work with my supervisors, Dr. Zaid Harchaoui and Dr. Cordelia Schmid, and, starting from the third year, Dr. Matthijs Douze.
I thank the supervisors for their effort and for the knowledge that I got during this long way.

I thank the jury members: Dr. Ivan Laptev, Dr. Patrick Pérez, and Dr. Florent Perronnin, for evaluating this work.

Many thanks to friends and colleagues from the LEAR team for the professional discussions and the multiple sports events. Special gratitude to Nathalie Gillot for the help with administrative questions.

Finally, I thank my family for their patience and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic interpretation and understanding of video data is a major field of active research within computer vision. A simplistic way to highlight the progress in this area over the last decades is to look at state-of-the-art computer vision systems that win TRECVID competitions. Focusing on the visual modality, state-of-the-art systems proceed by first computing low-level visual feature representations (static visual features, such as SIFT [Lowe, 2004, Zhang et al., 2007] or Deep Convolutional Net features [Bengio, 2009]; dynamic visual features, such as optical flow [Szeliski, 2010, Forsyth and Ponce, 2003]). This first step actually corresponds to mature computer vision approaches, which stood the test of time and can be deployed reliably at a large-scale with little fine-tuning. Then, mid-level feature representations, such as bag-of-visual-words (BoW) [Csurka et al., 2004] or Fisher vector (FV) [Perronnin et al., 2010], aggregate the information from the lower-level features such as SIFT or MBH [Wang et al., 2011, 2013]. Such feature representations already convey enough information to classify objects or actions into different categories. Finally, higher-level feature representations, such as attributes (higher-level properties of an object, shared across multiple classes), are used and complement the final feature representation that incomes a classifier such as Support Vector Machines (SVM) [Hastie et al., 2009].

Thus, as one moves forward along the pipeline, before the final classification stage, the visual feature representation that is computed by the system gets progressively higher-level, capturing more subtle properties of the video stream and grazing more "semantic" information. There are currently computer vision systems that are able to classify short chunks of real-world videos into pre-defined action or activity categories, with rather high classification accuracy [Laptev et al., 2008, Gaidon et al., 2013]. However, several tasks consisting of more automatic interpretation and understanding of video data still lie at the research frontier, where performance is still unsatisfactory. Automatically

generating short and informative video summaries currently can only be applied to very distinctive video streams, and may fall short when applied to real-world user-generated videos. Similarly, classification of long video chunks into more "semantic" categories, such as "romance" or "battle preparation" when skimming through a Hollywood action movie, remains a challenging problem.

In this PhD thesis, we propose effective approaches for these two latter problems, and present winning systems at the TRECVID Multimedia Event Detection challenges. First, we start by reviewing state-of-the-art approaches for these problems, and detail the current open problems. Second, we outline our contributions to the supervised video summarization task. Third, we outline our contributions to the classification of long semantic events, called "beat-event", in stylized Hollywood action movies. Finally, we present our contributions to the winning systems of the TRECVID Multimedia Event Detection challenges, describing the impact of each component in winning the competition.

1.1 Context

Image understanding has been an active research area during the past decade. The interest was supported by numerous practical applications, such as face detection and recognition, human detection and pose estimation, image retrieval and categorization. For example, a prominent success of computer vision, face detection, is based on several simultaneous ground-breaking works that leverage large collection of digital image data to train machine learning algorithms to detect faces in images [Viola and Jones, 2004, Rowley et al., 1998]. Face detection is now used in almost every commodity digital camera. The progress in image understanding could be explained as follows. First, the exponential growth of computing power allows to capture, store and process large collections of digital photographs with higher quality than previous film-based photographic technology. Second, machine learning algorithms [Hastie et al., 2009, Duda et al., 2012] and models have become mature enough to leverage visual information from large collections of digital photographs, allowing them to be applied to real-world problems.

Digital photos mostly convey the static information about the world, such as salient visual patterns (faces, objects, etc.). Some applications, like gesture recognition, video surveillance, sport analytics (Figure 1.1) require the visual dynamics to be analysed. Then, temporal sequences of images, *videos*, are used. A *video* is a sequence of images, captured at regular temporal rate (frame-rate). Early research on automatic video analysis focused on surveillance applications, which assist humans in preventing hazardous situations [Szeliski, 2010, Forsyth and Ponce, 2003]. For example, airport surveillance

FIGURE 1.1: Examples of sports actions from UCF 50 dataset [Reddy and Shah, 2013]. Analysis of the dynamics through temporal information gives additional cues to accurately recognize which technical element happens.

system must be able to detect suspicious activities such as "person leaves a bag" and "person puts a bag in a trash bin". Surveillance systems are also used in nursing homes for the elderly, at medical care institutions, in traffic control systems, and for many other security tasks [Aggarwal and Ryoo, 2011].

The human body can achieve an incredibly large number of poses, both static and in motion. Automatic analysis of dynamic human movements, *action recognition*, focuses on detecting human actions in video data, by means of the computer. Historically, first works focused on *gesture recognition*, one of the directions towards more intuitive human-computer interaction. In that scenario, gestures can be designed to be easily distinguished from each other. Besides understanding hand/arm gestures and postures, ongoing research focuses on recognition of facial emotions and reactions, and other subtle body movements known in whole as "body language". See [Weinland et al., 2011] for a survey on resp. action and gesture recognition.

Later works focused on recognizing more natural human actions, such as everyday actions ("open door", "sit down", etc.) or longer sports activities ("running", "weight lifting", "skateboarding", etc.). A recent trend is to focus on even higher level concepts (*events*) like "landing a fish", "birthday party", "sewing", "wedding", etc. These categories are quite diverse, especially in terms of activities performed and tools that are used. For instance, "sewing" can be done using a machine, or by hand, and "wedding" ceremonies differ in different countries. Common applications of action recognition include video indexing and categorization, surveillance, and sports annotation.

Video data is often lengthy and, therefore, takes significant amount of time to analyse. It is tempting to identify the parts that are most relevant to the desired goal. *Video summarization* searches for ways to represent a video in a more compact form, keeping only the important data. Known applications of video summarization [Truong and Venkatesh, 2007] include semi-automated database search [Truong and Venkatesh, 2007], video surveillance, detecting highlights in sports, influential moments in egocentric videos [Lu and Grauman, 2013], and culmination points [Truong and Venkatesh, 2007] in movies.

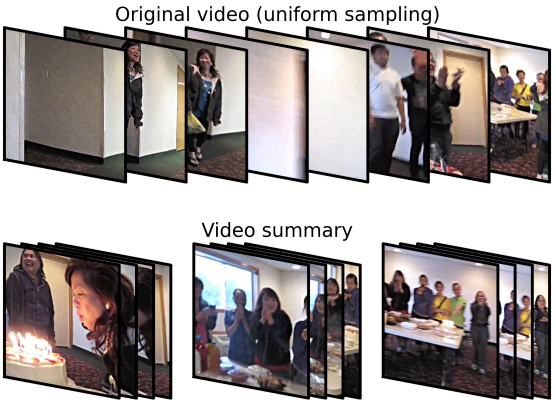


FIGURE 1.2: Example of a video summary output by Kernel Video Summarisation [Potapov et al., 2014].

Movies provide a large quantity of realistic video data, being a useful "experimentation lab" for computer vision researchers [Sivic and Zisserman, 2003, Laptev and Pérez, 2007]. An interesting property of the *movie data* is the involvement of hundreds or even thousands of people during the movie creation. Therefore, literally every second of movie contains implicit sense: movie lovers need to watch a movie many times to achieve a complete understanding. Movie data is therefore quite challenging. Also, in dynamic scenes, the camera angle can change after a fraction of second, which is nearly impossible to perceive from the first time. Therefore, we see the movie data as a way to develop more sensitive tools for video analysis.

1.2 Goals

The long-term goal of our work is automatic understanding of real-world videos. This goal implies answering such questions as:

- what happens in a video?
- what happened before and what will happen next?
- when does the most important event happen?
- what is the link between the events in the video?
- what suggests that the video represents a certain event?

More precisely, we formulate the following goals, that are specific to this dissertation.

Recognize events in videos accurately and efficiently. The amount of video data has been growing fast during the last decade. For example, 100 hours of video are uploaded to YouTube every minute [1]. On one hand, this is a problem, because it requires using scalable algorithms, which can absorb more data with a reasonable increase in computational resources. On the other hand, for a given learning model, more training examples allow to learn more accurate predictors, as we proceed along the learning curve [Hastie et al., 2009]. The trade-off between the prediction quality and the required computational resources is present in many real-world recognition tasks.

Identify the most important moments in videos. Analysing long videos is a time-consuming and resource-demanding process, both when done automatically or by a human. A convenient tool with a graphical user-interface, that localizes in time all the crucial points in a given video, can make such a task much easier. Such tools already exist for specialized tasks, for example, it is possible to identify "goal" moments in football matches. Our goal, however, is a general-purpose approach, which can learn the importance criterion from a set of weakly annotated videos of a given domain.

Adapt the classifiers to a specific dataset. The performance of machine learning algorithms depends on the distributions of training and test data [Hastie et al., 2009]. On one hand, when the training data comes from the same source as the test data, better results are often expected. On the other hand, training data from the same source is often scarce and it is necessary to use data coming from a different distribution. The goal is to understand how to adapt classifiers trained on a *source domain* to another, —*target domain*.

Quantitative evaluation of video analysis algorithms. Although many video understanding tasks are interrelated, it is often easier to focus on a single problem formulation, with a formally-defined evaluation criterion. Additionally, learning tasks require a dataset, with formally defined resp. training and evaluation procedures, along with proper performance metrics. Constructing a new dataset implies collecting the videos, setting up an annotation protocol and ensuring the annotation consistency.

1.3 Contributions

We outline here our main contributions.

[1] https://www.youtube.com/yt/press/statistics.html

FIGURE 1.3: A 5-minute extract from our Action Movie Franchises dataset, ground truth annotation and output of different methods. Each colors stands for a different event category: green —*pursuit*, blue —*battle*, yellow —*victory-good*, green —*despair-good*, pink —*romance*, gray —*victory-bad*, cadet blue —*good-argue-good*. Hashes mark difficult examples. The color code for the classifier evaluation is: white = true positive, gray = ignored, black = false positive.

Category-specific video summarization. (Chapter 3, Fig. 1.2)

- We propose a novel approach, for supervised video summarization of realistic videos, that uses state-of-the-art image and video features. It consists of two parts: temporal video segmentation and supervised importance scoring. The proposed Kernel Temporal Segmentation approach is efficient with high-dimensional frame descriptors and allows automatic calibration of the number of change-points.
- We introduce a new dataset, MED-Summaries, along with a clear annotation protocol, to evaluate video summarization. It consists of more than $10,000$ temporal segments annotated with importance ratings.
- We obtain experimental results on the MED-Summaries dataset, showing that the proposed approach delivers video summaries with higher overall importance, as measured by two performance metrics.

Event Detection in Action Movie Franchises. (Chapter 4, Fig. 1.3)

- We introduce the **Action Movie Franchises** dataset, which features dense annotations of 11 beat-categories in 20 action movies at both shot and event levels. Our categories have a higher semantic level than in most of the existing datasets. To the best of our knowledge, a comparable dense annotation of videos does not exist.
- We define several evaluation protocols, to investigate the impact of franchise-information, that is testing with or without previously seen movies from the same franchise. We study the impact for both classification and localization tasks.

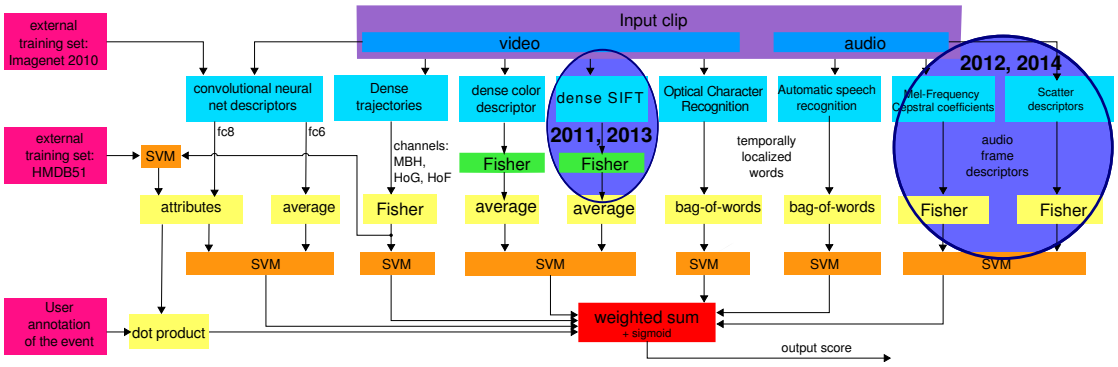


FIGURE 1.4: Overview of the whole INRIA-LIM-VocR and AXES system for TRECVID MED 2014 [Douze et al., 2014]. Circles denote areas of our contributions through years 2011-2014.

- We propose an approach for classification of video shots into beat-categories based on a state-of-the-art pipeline for multimodal feature extraction, classification and fusion. The approach for localizing beat-events uses a temporal structure inferred by a conditional random field (CRF) model learned from training data.

Contributions to TRECVID Multimedia Event Detection submissions. (Appendix A, Fig. 1.4)

TRECVID Multimedia Event Detection (MED) [Over et al., 2014] is a competition on event retrieval and categorization in real-world videos, held annually by NIST (National Institute of Standards and Technology) starting with a pilot study in 2010. The most challenging part is the large evaluation set of 8000 hours of video. In Appendix A we summarize the contributions made during the participations in the challenge from 2011 to 2014. Note, that these contributions are more technical than the contributions presented in Chapters 3 and 4.

- We present static (image) visual descriptors for video classification, which achieve state-of-the-art results. The description technique is efficient for videos captured in unconstrained filming conditions and with various durations.
- We report experimental results with two types of audio descriptors, Mel Frequency Cepstral Coefficients (MFCC) and Scattering Transform Coefficients (ScatNet), which achieve state-of-the-art results.
- We describe different versions of the dataset and show the evolution of performance over the years.

Chapter 2

Related work

In this chapter, we give a general overview of related works. We give more specific references in the "Related Work" sections of the following chapters.

2.1 Video summarization

In this section, we mention only the most prominent approaches for video summarization. A broader overview can be found in [Truong and Venkatesh, 2007]. Note that there exist two types of video summarization forms [Truong and Venkatesh, 2007]: *keyframes* and *video skims*. Figure 2.1 highlights the main difference between these two types of video summaries.

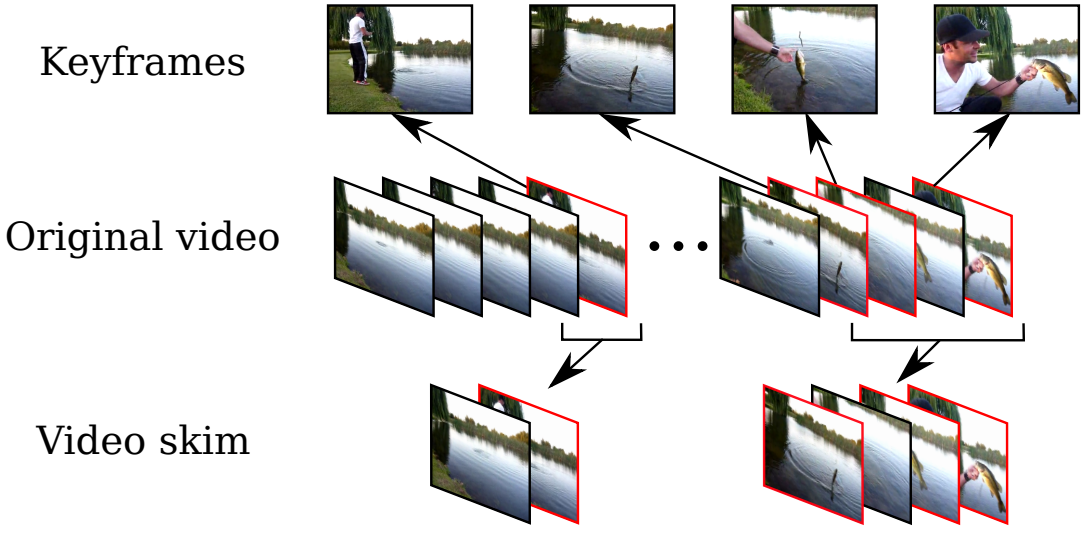


FIGURE 2.1: Illustration of the two forms of video summarization. *Keyframes* are better viewed in static, while the *video skims* preserve the dynamics.

A *video skim* is a representative subset of temporal segments in a given video. Since the two forms share many similarities, we review the works for both of them.

The idea of *summarization* was initially proposed for text data. The goal of *automatic document summarization* [Manning et al., 2008], is the automatic construction of text summaries, similar to what humans can do. A summary should *identify and convey the main points* of the text, be *concise* and *grammatically correct*. Consequently, the summary should *avoid non-important* and *repetitive* content, which only deteriorates the quality of the summary. Video summarization aims for similar goals.

Video summaries are constructed in relation to different use cases. In one scenario, a summary simplifies browsing a single video. In that case coverage is more important than conciseness. In a different scenario, a summary is a way to speed up retrieval in a large dataset. In that case a summary can be either an unsupervised or a query-specific shortened version of the video in the database.

We focus here on 4 main aspects of a video summary:

- saliency,
- diversity,
- temporal consistency,
- and duration.

Figure 2.2 illustrates the diversity-duration trade-off in video summarization.

The notion of **saliency**, also known as importance or "interestingness", comes from uneven distribution of the content over the video. Important details are often concentrated in small chunks, while the rest of the video can be omitted. Ngo et al. [2005] model the user attention, as a cue to saliency in movie data. In most cases, (temporal) saliency can be learned in a supervised way. In the context of video skimming, we shall use the term "saliency" to refer to temporal saliency, in contrats to spatial saliency which is more common for image recognition tasks. Rui et al. [2000], Xie et al. [2004] learn to detect interesting moments of baseball and soccer videos, such as goal attempts and corner kicks. For egocentric videos it is useful to learn the saliency of objects from spatial segmentation masks of important objects [Lee et al., 2012]. When the full annotation is not available, weak supervision can be utilized, e.g. using the multiple instance learning (MIL) framework [Wang et al., 2012, Duda et al., 2012]. For keyframe summaries, images retrieved by a search engine can add more supervision for summarization of car advertisements [Khosla et al., 2013] and many other classes of user

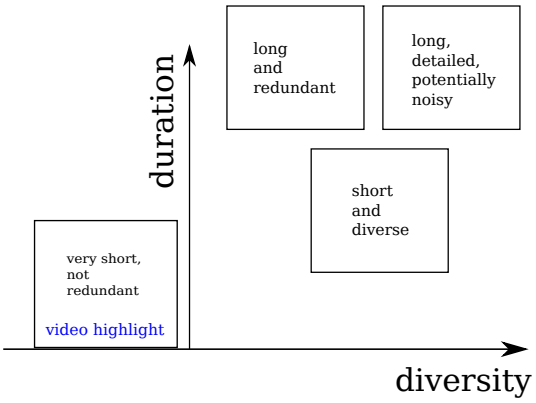


FIGURE 2.2: Diversity-duration trade-off in video summarization.

generated videos [Kim et al., 2014]. Sun et al. [2014] leverage edited videos to rank video chunks for the presence of a highlight. While early methods focused on a single domain, a recent trend is to automatically learn the importance for each video category [Li et al., 2012, Kim et al., 2014, Gygli et al., 2014, Sun et al., 2014].

Diverse summaries do not contain repetitive temporal segments. Given a fixed duration limit, the more diverse the summary is, the more information it preserves. The traditional approach for eliminating redundancies is to cluster frames or chunks of a video in an unsupervised way, and then select the most representative ones [Truong and Venkatesh, 2007, Lee et al., 2012, Ngo et al., 2005, Vermaak et al., 2002]. Multiple works proposed better ways to accomplish the temporal clustering. Vermaak et al. [2002] propose an efficient keyframe extraction algorithm, for browsing of generic video sequences, which maximizes the difference between consecutive keyframes. This group of approaches is known as "Minimum correlation among keyframes" [Truong and Venkatesh, 2007]. Online sparse reconstruction approach [Zhao and Xing, 2014] can summarize potentially infinite videos. A dictionary of local features is constructed on-the-fly. A new chunk is included in the summary, once it cannot be accurately approximated by the current dictionary. Khosla et al. [2013] apply discriminative clustering, such that both diversity and saliency are optimized at the same time. Redundancies can be penalized by including a diversity prior in the optimization objective [Lu and Grauman, 2013]. Note, that maximizing the diversity alone suffers greatly from outliers [Truong and Venkatesh, 2007].

Temporal consistency of a summary is often dictated by the application. In movie data it can be the rhythm [Sundaram et al., 2002] or the culmination points [Truong and Venkatesh, 2007]. Kim et al. [2014] model the typical storylines in user videos of each event category. The story progress is also preserved in [Lu and Grauman, 2013] to create story-driven summaries of egocentric videos. Additional temporal restrictions, such as

cinematic rules can apply in some domains [Gygli et al., 2014, Li et al., 2012, Truong and Venkatesh, 2007]. Other works model the distribution of important moments in time [Xie et al., 2004, Kim et al., 2014], leveraging the temporal context and the domain specifics.

Duration of a summary is either limited in advance or should be determined automatically, optionally for a given level of detail. In contrast to keyframe summaries, the duration of a video skim is not proportional to the number of temporal segments. More details can be found in [Truong and Venkatesh, 2007].

A recent work [Wang et al., 2014] directly gathered the desired aspects of a summary within a user study. For user generated videos, these aspects are: 1) strong connection to the dominant semantics of the original video, 2) comprehensibility of the original story and little redundancy, 3) high quality of the segments.

To achieve the summarization goal, most methods rely on an optimization-based approach [Truong and Venkatesh, 2007]: define an objective function and then optimize it. While saliency can be learned before seeing the video to summarize, diversity is inferred for the video at hand.

Features. Automatic video summarization methods use different low-level features. Early methods relied on hand-crafted features [Lee et al., 2012, Sundaram et al., 2002], color and motion [Lee et al., 2012, Wang et al., 2012, Xie et al., 2004, Rui et al., 2000, Vermaak et al., 2002], and audio [Rui et al., 2000] features. Later methods utilized global features [Kim et al., 2014, Lu and Grauman, 2013], bag-of-features approach [Wang et al., 2012, Kim et al., 2014], and its extended counterparts [Sun et al., 2014, Khosla et al., 2013] such as Fisher Vectors [Perronnin et al., 2010] and Locality-constrained Linear Coding [Wang et al., 2010, Chatfield et al., 2011]. These works suggest that insufficiently discriminative features prevent the generalisation to new domains [Khosla et al., 2013]. Another alternative is to directly reconstruct the local features using learned codewords [Zhao and Xing, 2014], without the aggregation over the chunk. A recent approach [Gygli et al., 2014] automatically predicts multiple high-level features, such as attention, aesthetics, location notability, etc. For efficiency some methods work directly in the compressed domain [Ngo et al., 2005, Truong and Venkatesh, 2007].

2.1.1 Datasets

For some time, there have been no common datasets for video summarization publicly available to the research community. An important body of works reports experiments

on the Open Video Archive [Ope], which lacks dynamics and semantics in the video, because most of the information is communicated through audio.

From 2006 to 2008, TRECVID has been running the BBC Rushes summarization competition. Unedited video data from the BBC Archive was provided to participants. It contained the scenes for BBC drama programs, with multiple takes and the footage between the actual play. The expectation of the competition was that summarizing such rushes might significantly simplify the overall rushes management process. However, the methods were mostly specific to the domain, i.e. they focused on detecting redundant shots of a scene and clapperboards.

In 2012 TRECVID started the Multimedia Event Recounting (MER) competition [Over et al., 2012, 2013, 2014]. The goal of the competition is to present an *important evidence* for the videos classified to a particular event, to allow the users accurately and rapidly find the videos of interest via the *recountings*. Participants were asked to provide video clips together with the textual descriptions. The best overall participant in 2013 achieved 63% accuracy with summaries of 44% of the total video duration. In 2014, for all the teams the evidence recounted by the systems was more convincing for the videos containing an instance of the event, than for the videos that did not [Over et al., 2014].

We review recent publicly available datasets in Chapter 3.

2.1.2 Evaluation metrics

Existing evaluation approaches can be classified into 4 categories:

- concept coverage (mostly objective),
- user studies of summaries (subjective),
- comparison to multiple summaries (mostly subjective),
- problem-specific evaluation (mostly objective).

We now discuss each of the categories in detail.

Concept coverage The main idea is to measure the percentage of captured concepts (or events). It requires manual annotation of the concepts in the test videos.

For example, Lee et al. [2012] use, as a ground truth, spatial annotations of important objects, collected with Amazon Mechanical Turk. Important region prediction is the

major part of their summarization pipeline, and it is first evaluated separately. A region r is considered to be a true positive, if and only if

$$\exists \text{ ground truth region } g \quad s.t. \quad \frac{\text{Area}(r \cap g)}{\text{Area}(r \cup g)} > 0.5 \tag{2.1}$$

i.e. its overlap score with any ground truth region is greater than 0.5. Aggregation over the whole dataset is done using the precision-recall curves and the *average precision* (area under the precision-recall curve). Finally, for the whole summarization pipeline, Lee et al. [2012] report the *importance-based summarization accuracy*, which is the recall rate, as a function of the summary length N:

$$f_N(S) = \frac{\#\text{ objects } \in S}{\#\text{ objects total}} \quad s.t. \quad \text{length}(S) = N. \tag{2.2}$$

Multiple instances of the same object are counted as one object.

Wang et al. [2012] first validate the correctness of concept classification using the standard accuracy metric. The whole summarization approach is evaluated using NDCG@10 ranking metric, that expects graded ground truth scores. For a given video, key-shots are selected by a human. Instead of ranking the shots by the annotators, near-duplicate detection is applied to the key-shots and the group sizes are used as a reference. NDCG@k ranking metric is defined as [Manning et al., 2008]:

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{\log_2(1+m)}, \tag{2.3}$$

where $R^{(j,m)}$ is the ground truth score of the m-th element in the ranked list of query j; Z_{kj} is the normalization factor that makes a perfect ranking's NDCG at k for query j equal to 1; k is the size of the retrieval shortlist.

Additionally, Wang et al. [2012] validate the tag localization part of their approach using the standard accuracy metric for classification.

The authors of Sun et al. [2014] collected highlight annotations in a user generated videos from 5 people, with one temporal window per video. Only the videos where the annotators reached consensus are used for evaluation. Each video is partitioned into regular 100 frames chunks with 50 frames overlap. Chunks of a video have binary labels, with $+1$ for highlights and -1 for the rest of the video. Since the summarization algorithm of [Sun et al., 2014] outputs a ranking of the chunks, average precision metric is computed for each video, like a small retrieval problem. Finally, the *mean average precision* (mAP) is reported for all videos.

The advantage of the concept coverage method is that the manual annotation is required only once. Further evaluation of summaries on the annotated dataset is fully automatic. On the other hand, annotation of videos is often more difficult than comparing summaries as it is done in user studies.

User studies of summaries Summaries are mostly created for humans. Therefore a user study is the direct way to evaluate an automatic summarization method.

In [Lu and Grauman, 2013] the user is showed a sped-up original video and two different summaries. The user selects the preferred summary, among the baseline and the proposed method. The summaries are shown in random order, such that the user is not aware which method created the summary. While making his/her choice, the user is asked to focus on the progress of the story, redundancy and representativeness of each sub-event. For every video and every baseline, 5 user votes are collected. The overall figure is the percentage of user loyalty with the proposed method versus the baseline. A similar method is employed in [Lee et al., 2012].

In the setup of [Khosla et al., 2013], the user is asked to rate on a scale from 1 to 10 each of 4 summaries. One of the summaries is constructed by a user from Amazon Mechanical Turk, the others are automatically generated. Finally, user grades of summaries serve as an input to the average precision.

In the user study of Ngo et al. [2005], the user is asked to rate the summaries in terms of enjoyability and informativeness. The summaries with different level of detail are shown, from shortest to longest, followed by the original video.

User studies of summaries take into account multiple aspects of the summary: importance, diversity, temporal consistency, duration and also the application specifics. Therefore this is the most common evaluation approach. User studies are, however, time-consuming and hard to reproduce.

Comparison to multiple summaries In the document summarization literature, it was noted [Lin, 2004] that counting the number of common n-grams in automatically generated summary and a set of ground truth summaries highly correlates with human evaluation [Lin, 2004].

Similarly, to automate the comparison of keyframe summaries, previous works relied on such frame descriptors, as: color and edge histograms [de Avila et al., 2011, Wang et al., 2012], Bag of Visual Words [Wang et al., 2012] and SIFT flow [Khosla et al., 2013].

More specifically, Khosla et al. [2013] define a distance measure between 2 frames, based on the SIFT flow warping algorithm. For a pair of frames F_1 and F_2, the algorithm outputs a warped image F_1^w. Function d is defined as the average of squared pixel differences:

$$d(F_1, F_2) = \frac{|F_1^w - F_2|^2}{P}, \tag{2.4}$$

where P is the number of pixels, such that $d(F_1, F_2) \in [0, 1]$. Based on this measure, the optimal bipartite matching is computed between an automatic summary $S = \{s_1, \ldots, s_k\}$ and a user summary $H = \{h_1, \ldots, h_n\}$. Let $\sigma_1, \ldots, \sigma_k$ be the best matches for S and $\xi_1, \ldots, \xi_n$ be the best matches for H. Based on this matches, the analogues of precision and recall metrics are defined as,

$$\text{precision} = \frac{1}{k}\sum_{i=1}^{k}(1 - d_{\sigma_i,i}), \qquad \text{recall} = \frac{1}{n}\sum_{i=1}^{n}(1 - d_{i,\xi_i}), \tag{2.5}$$

where the matrix $[d_{j,\ell}] \in \mathbf{R}^{n \times k}$ is the distance matrix between all pairs of frames of the summaries. Finally, the average precision, that is the area under the precision-recall curve, is computed using the summaries of different length.

To the best of our knowledge, comparison to multiple summaries was never applied to video skims.

Problem-specific evaluation When the goal is to simplify video search or navigation, the gain in efficiency (time required to accomplish the target task) can be directly used as a quality metric. TRECVID MER competition [Over et al., 2012, 2013, 2014] is aimed to simplify the search of relevant videos. For each retrieved video, the algorithm is required to output a set of video segments, containing the evidence of the queried category. Then, during a stage called *triaging*, a user verifies whether a video corresponds to the query or not (only by watching the summary). The MER task defines two video summarization metrics:

1. average time required to triage a video,
2. accuracy of triaging decisions.

Similar to user studies, this process requires multiple human judges.

To increase the evaluation precision, most of existing works on video summarization use at least two different evaluation approaches, usually one objective and one subjective.

2.2 Video classification

Originally, the research on video classification mostly focused on action recognition. Many relevant works are summarized in a survey [Poppe, 2010]. In this section, we briefly summarize main structural elements of the action and the event recognition pipelines.

The difference between the action [Schuldt et al., 2004, Rodriguez et al., 2008] and the event [Over et al., 2014, Potapov et al., 2015] is the following. The action is short and is usually performed by a single human. The event is a complex activity often involving multiple humans, possibly related to a specific location or time, and comprising a set of characteristic actions [MED]. Regardless the difference between the action and the event, some of recent pipelines achieve state-of-the-art performance for both of them [Wang et al., 2013].

2.2.1 Datasets

Table 2.1 shows the quantitative statistics of some of the recent video datasets.

	#clips	average length	background length	#categories	data source
KTH [Schuldt et al., 2004]	2391	4.8 s	-	6	Lab recording
UCF Sports [Rodriguez et al., 2008]	300	6.1 s	-	10	TV
Hollywood2 [Marszalek et al., 2009]	1707	11.6 s	-	12	Movies
HMDB51 [Kuehne et al., 2011]	6766	3.15 s	-	51	Movies/Internet
UCF101 [Soomro et al., 2012]	13320	7.2 s	-	101	Internet
Sports1M [Karpathy et al., 2014]	1058888	4 m 8 s	-	487	Internet
Coffee&Cigarettes [Laptev and Pérez, 2007]	264	$\sim$ 2 s	$\sim$ 0.5 h	2	Movies
DLSBP [Duchenne et al., 2009]	266	$\sim$ 2.5 s	$\sim$ 4 h	2	Movies
TRECVID MED [Over et al., 2014]	207000	2 m 30 s	$\sim$ 7000 h	20	Internet

TABLE 2.1: Selection of a few representative event and action recognition datasets.

2.2.2 Evaluation metrics

Let us define the *supervised classification problem* [Duda et al., 2012, Hastie et al., 2009]. Suppose we have a *training set*

$$\boldsymbol{x}_1^{tr}, \dots \boldsymbol{x}_{n_{tr}}^{tr}, \quad \boldsymbol{x}_i^{tr} \in \mathcal{X}, \quad i = 1, \dots, n_{tr}; \tag{2.6}$$

$$y_1^{tr}, \dots y_{n_{tr}}^{tr}, \quad y_i^{tr} \in \mathcal{Y}, \quad i = 1, \dots, n_{tr}, \tag{2.7}$$

where $\boldsymbol{x}_i^{tr}$ is the descriptor for i^{th} training example, and y_i^{tr} is its category label. In practice, $\mathcal{X}$ is often a Euclidean space $\mathbb{R}^d$ and $\mathcal{Y}$ is a finite set $\{1, \ldots, m\}$. Solving the supervised classification problem implies constructing a classifier function (*classifier*) $f : \mathcal{X} \to \mathcal{Y}$ that distinguishes examples of different categories.

The quality of the classifier is evaluated on a separate *test set*:

$$\boldsymbol{x}_1^{te}, \ldots, \boldsymbol{x}_{n_{te}}^{te}, \quad \boldsymbol{x}_i^{te} \in \mathcal{X}, \quad i = 1, \ldots, n_{te}; \tag{2.8}$$

$$y_1^{te}, \ldots, y_{n_{te}}^{te}, \quad y_i^{te} \in \mathcal{Y}, \quad i = 1, \ldots, n_{te}. \tag{2.9}$$

Ideally, $f(\boldsymbol{x}_i^{te})$ should be equal to y_i^{te} in as many cases as possible.

The simplest metric, *accuracy*, directly measures the frequency of correct predictions on the test set:

$$\text{Accuracy} = \frac{\left|\{i \quad s.t. \quad f(\boldsymbol{x}_i^{te}) = y_i^{te}\}\right|}{n_{te}}, \tag{2.10}$$

where the numerator is the number of correctly predicted test labels.

Let us now consider the case of *binary classification*: $\mathcal{Y} = \{+1, -1\}$. The examples with label $+1$ are called *positive examples*, and the examples with label -1 are called *negative examples*. When a classifier predicts a label $\hat{y} = f(\boldsymbol{x})$ and the ground truth label is y, there can be four cases:

- **True Positive (TP)**: $\hat{y} = +1, y = +1$,
- **True Negative (TN)**: $\hat{y} = -1, y = -1$,
- **False Positive (FP)**: $\hat{y} = +1, y = -1$,
- **False Negative (FN)**: $\hat{y} = -1, y = +1$.

First two cases correspond to correct predictions, while last two cases are mistakes. In practice false alarm (false positive) and missed detection (false negative) often have different significance. For example, in the *retrieval* problem, where the negative examples prevail in the quantity, a classifier that always predicts -1 will have the accuracy close to 1.0, while it does nothing useful. Therefore a better metric is required.

Let $\#\text{TP}, \#\text{TN}, \#\text{FP}, \#\text{FN}$ be the total counts for each of four cases.
Precision is defined as

$$\text{Precision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}}, \tag{2.11}$$

that is the fraction of true positive examples among all the examples predicted as positive.

Recall is defined as

$$\text{Recall} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}}, \tag{2.12}$$

that is the fraction of correctly classified examples among all positive examples.

There is often a trade-off between precision and recall. Let us define a thresholded classifier as

$$f_t(\boldsymbol{x}) = \begin{cases} +1, & s(\boldsymbol{x}) \geq t, \\ -1, & s(\boldsymbol{x}) < t. \end{cases} \tag{2.13}$$

where $s : \mathcal{X} \to \mathbb{R}$ is the *classification score*, and $t \in \mathbb{R}$ is a *threshold*. Changing the threshold changes the predicted labels. Therefore, we can define a precision-recall curve in a parametric form:

$$\{(\text{Recall}(t), \text{Precision}(t)) \quad t \in \mathbb{R}\}. \tag{2.14}$$

The area under the curve on the 2D plane, limited by X and Y axes, is called **average precision** (**AP**). Average precision shows the performance of a classifier for multiple threshold settings.

Values of the accuracy, the precision, the recall and the AP lie in $[0, 1]$ interval. For the classifier with no mistakes, all these metrics equal to 1.

A multiclass classification problem with m categories can be partitioned into m binary problems: category i is treated as positive class, while the others count as negative. Correct solution of each binary problem implies correct solution of the full problem. *Mean average precision* (**mAP**), defined as the average AP over the m binary problems, is a common way to aggregate performance in the multiclass setting.

For problems with a larger number of categories, the classification accuracy metric is also common [Soomro et al., 2012, Karpathy et al., 2014].

Compared to other metrics, the average precision does not require to specify the classification threshold and only relies on the ranking of the test examples. Meanwhile, the accuracy, the precision and the recall, and many other metrics, all depend on the threshold. Normalized Detection Cost [Over et al., 2011], which is defined as a weighted linear combination the False Alarm rate and the Missed Detection rate, with a much higher cost for False Alarms, requires to estimate the threshold given the target ratio of false alarms versus missed detections.

2.2.3 Classification pipeline

Support Vector Machines The *Bag of Visual Words* model was inherited from the document classification literature, and is now widespread in the computer vision

community for a larger number of recognition tasks on images and videos [Csurka et al., 2004, Tuytelaars and Mikolajczyk, 2007].

Training a linear SVM classifier [Hastie et al., 2009] consists in solving the following convex optimization problem:

$$\underset{\boldsymbol{w},b}{\text{Minimize}} \quad \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\max\big(0, 1 - y_i^{tr}(\boldsymbol{w}^T\boldsymbol{x}_i^{tr} + b)\big), \tag{2.15}$$

where $\boldsymbol{w}$ is the weight vector, b is the bias term, $\boldsymbol{x}_i^{tr}$ is the i'th vector in the training sample and y_i^{tr} is the corresponding label (+1 or -1). At test time, the predicted label for a new example $\boldsymbol{x}_i^{te}$ is $f(\boldsymbol{x}_i^{te}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x}_i^{te} + b)$.

Kernel SVM [Hastie et al., 2009, Shawe-Taylor and Cristianini, 2004] is a modification of model (2.15), which better aligns with the input space structure. It was successfully applied to the tasks of image and video classification [Zhang et al., 2007, Laptev et al., 2008].

Other classification pipelines Since the seminal work of Csurka et al. [2004], the SVM classification pipeline has been the *de-facto* state-of-the-art in image and video classification for almost a decade [Zhang et al., 2007, Laptev et al., 2008, Wang et al., 2013]. In the SVM-based pipeline, the video description and classification stages are completely separated. Therefore numerous works have explored how to add supervision to the description stage: by supervised learning of the local feature quantizer [Krapac et al., 2011a], through mid-level features [Maji et al., 2011], by discriminative modeling of spatial saliency [Sharma et al., 2012], etc.

Recent advances in Deep Convolutional Networks (DCN) [Krizhevsky et al., 2012, Bengio, 2009] made it possible to learn descriptors and classifiers within a single framework. The model consists of a hierarchy of layers with raw image pixels as input and the category label as output. Main structural elements of the DCN model are: convolution, non-linearity and pooling.

A few representative approaches that explicitly model the temporal structure of the video are reviewed in Chapter 4.

2.2.4 Video descriptors

In this section we describe the top-performing local image and video descriptors, their aggregation techniques, and briefly mention other state-of-the-art visual and audio descriptors.

Local patterns in the video are very important for discriminating its content [Zhang et al., 2007]. *Static descriptors*, such as SIFT [Lowe, 2004], are good for describing the shape and the appearance of objects, but lack the motion information. *Motion descriptors* are specifically constructed to capture the shape and the appearance changes over time. The Dense Trajectories approach [Wang et al., 2013, Wang and Schmid, 2013] (see Figure 2.3) relies on the optical flow to track each local keypoint and then describes its neighbourhood with 4 descriptors: Histogram of Gradients (HoG), Histogram of Flow (HoF), Motion Boundary Histogram (MBH) and the normalized trajectory. HoG, HoF and MBH descriptors compute histograms of directions on a spatial grid, but take different inputs: the spatial gradients for HoG, the optical flow for HoF, the spatial gradients of the optical flow for MBH. The normalized trajectory is a vector of displacements of the local keypoint across a few frames (usually 15). For motion descriptors it is important to stabilize the video by compensating the camera motion [Wang and Schmid, 2013, Gaidon et al., 2014, Jain et al., 2013b].

Local descriptors can be extracted at salient points or on a dense grid. The latter is better in general [Wang et al., 2013, Fei-Fei and Perona, 2005], while careful feature pruning can give additional gains [Wang and Schmid, 2013].

Local descriptors are often redundant and their number may change. Therefore it is a common practice to *aggregate* the local descriptors into a fixed-size global descriptor. In the *Bag of Visual Words (BoW)* model [Csurka et al., 2004], locals descriptors are quantized using a codebook and a video is described by a histogram of codewords. A more advanced *Fisher Vectors (FV)* model relies on Gaussian Mixture Model (GMM) [Hastie et al., 2009, Bishop, 2009] soft assignment and encodes an image (or a video) with a vector of derivatives of the log-likelihood function [Perronnin et al., 2010]:.

$$\mathcal{G}_{w,k} = \frac{1}{T w_k} \sum_{t=1}^{T} \gamma_t(k),$$

$$\mathcal{G}_{\mu,k} = \frac{1}{T \sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{\xi_t - \mu_k}{\sigma_k} \right),$$

$$\mathcal{G}_{\sigma,k} = \frac{1}{T \sqrt{2 w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{(\xi_t - \mu_k)^2}{\sigma_k^2} - 1 \right),$$

where ξ_t denotes t'th local descriptor of the image (or the video), and $\gamma_t(k)$ is the posterior probability of t'th local descriptor to belong to k'th Gaussian.

It was shown that normalizing the Fisher Vectors with a signed power-transform, followed by L2-normalization increases its robustness [Perronnin et al., 2010].

Both BoW and FV models discard the spatial and the temporal positions of the descriptors. A number of spatio-temporal extensions have been proposed for these models [Lazebnik et al., 2006, Laptev et al., 2008, Krapac et al., 2011b].

Other descriptor pipelines Other methods for video description include mid-level representations, semantic attributes and deep networks.

Maji et al. [2011] introduce a mid-level model for human actions. They encode each detection with a vector of *poselet activations*. *Poselets* are part detectors, learned in a semi-supervised way from joint annotations. Jain et al. [2013a] and Raptis and Sigal [2013] extend the idea to videos.

Object Bank [Li et al., 2010] and *Classemes* [Torresani et al., 2010] are the models for high-level image description. The image is described by a vector of classification scores of pre-trained classifiers. The models accept adding new categories at little cost and are designed to suit well for high-level computer vision tasks.

Deep Convolutional Network (DCN) are currently quite popular for joint supervised image description and classification [Krizhevsky et al., 2012, Bengio, 2009]. It bridges the gap between image pixels and class labels, filtering the information through multiple interconnected layers. The idea is currently being extended to videos [Karpathy et al., 2014]. As described in Section 2.3, the DCN approach is also used within the attribute classification pipeline.

2.2.5 Audio descriptors

The audio channel often complements the information from the visual channel. Although the audio signal is one-dimensional, there are still many challenges in automatic audio recognition. A major difficulty comes from the multi-scale nature of the signal: pitch and timbre, rhythm, and music progression, — all belong to different scales [Andén and Mallat, 2013a].

Mel Frequency Cepstrum Coefficients (*MFCC*) capture rather short-term frequency-based features of audio signals [Rabiner and Schafer, 2007]. MFCC is defined as the inverse Fourier Transform of the log magnitude spectrum of a signal. MFCC descriptors proved to be efficient for multiple audio recognition tasks such as genre classification [Tzanetakis and Cook, 2002] and multimedia event detection [Douze et al., 2014].

Scattering Transform Coefficients (*ScatNet*) capture longer-term temporal dependencies in the audio signal, for windows typically longer than 1 second. ScatNet descriptors are based on a cascade of wavelet-like convolutional operators and modulus transforms [Andén and Mallat, 2011].

Both MFCC and the ScatNet descriptors are computed within local windows slided over the audio signal. Aggregation along the temporal axis can be done using the Bag of Words and the Fisher Vectors models [Bao et al., 2011, Oneata et al., 2012], similar to the video classification pipeline described above.

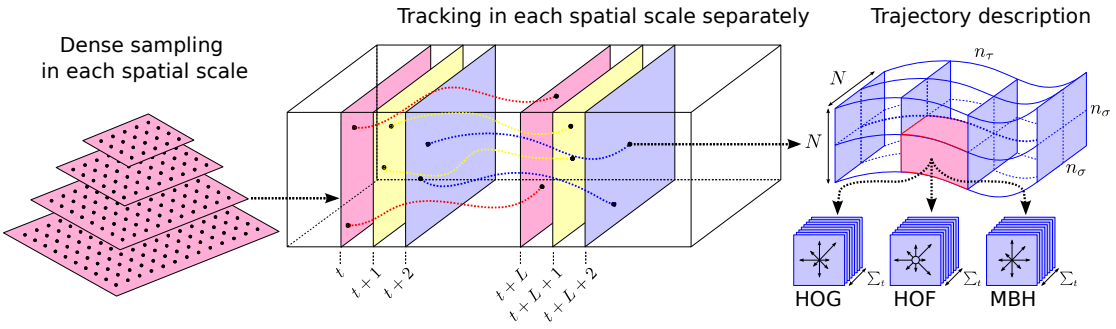


FIGURE 2.3: Illustration of the Dense Trajectories approach, which takes into account the local motion in the video. Courtesy of [Wang et al., 2013]

Figure 2.3 illustrates the video description approach of Wang et al. [2013].

2.3 TRECVID Multimedia Event Detection

In this section we review the systems that performed best in TRECVID Multimedia Event Detection (MED) competition from 2011 to 2014. The INRIA-AXES submissions are described in Appendix A.

TRECVID MED 2011 All top-ranked systems included local descriptors for image, motion and audio channels. Another standard practice was to use semantic concepts. They provide high-level information, which helps to close the semantic gap. Semantic models usually rely on local descriptors and their training requires the annotation of concepts, possibly on a separate dataset. Third type of information, textual, was often considered of low utility, because it is present only in a small fraction of videos.

Participants came up with novel ways for fusion of different modalities and pooling of the local features along the video. All the systems relied on variants of the Support Vector Machines (SVM) [Duda et al., 2012, Hastie et al., 2009] classifier.

Early fusion means concatenation of multiple descriptors of the same video into a single vector. Assuming a video is described by a set of n descriptors

$$\boldsymbol{x}^1 \in \mathbb{R}^{d_1}, \quad \boldsymbol{x}^2 \in \mathbb{R}^{d_2}, \quad \ldots, \ \boldsymbol{x}^i = (x^i_1, \ldots, x^i_{d_i}) \in \mathbb{R}^{d_i}, \quad \ldots, \ \boldsymbol{x}^n \in \mathbb{R}^{d_n} \tag{2.16}$$

the early fusion descriptor for this video will be

$$\boldsymbol{x}^{early} = (w_1 x^1_1, \ldots, w_1 x^1_{d_1}, \ w_2 x^2_1, \ldots, w_2 x^2_{d_2}, \ \ldots, w_n x^n_1, \ldots, w_n x^n_{d_n}). \tag{2.17}$$

In the context of kernel methods, it relates to (weighted) summation of kernel matrices, computed for different descriptors [Shawe-Taylor and Cristianini, 2004].

Late fusion implies combining the scores of classifiers, trained separately for each channel. Let $s_1(\boldsymbol{x}) \in \mathbb{R}, s_2(\boldsymbol{x}) \in \mathbb{R}, \ldots, s_n(\boldsymbol{x}) \in \mathbb{R}$ be the classification scores for a video $\boldsymbol{x}$. Then, the late fusion score is computed as

$$s_{late}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i s_i(\boldsymbol{x}). \tag{2.18}$$

Note that for linear classifiers, the early and late fusions can be applied interchangeably. However, the function minimized during training is different, so the classifier will be different. The late fusion is computationally more efficient, although the relative weighting of descriptor components remains the same for different combinations of descriptors.

Natarajan et al. [2011] (see also [Natarajan et al., 2012b]) use multiple methods for the early and late fusions of different channels. For the early fusion, they rely on Multiple Kernel Learning (MKL). MKL is a technique to select the early fusion weights in a discriminative manner such that the classification performance is maximized. For the late fusion, Natarajan et al. [2011] utilize two different approaches: a Bayesian decision theoretic approach and a weighted average fusion.

Cao et al. [2011] (see also [Cao et al., 2012]) also experimented with multiple fusion techniques. Firstly, they show that using AdaBoost to select the most informative channels, during the early fusion, improves over the uniform average. Secondly, they train SVMs [Duda et al., 2012, Hastie et al., 2009] from scores generated by 780 visual, 113 action and 56 audio high-level concepts. The concepts themselves are learned using SVM improved with Robust Subspace Bagging. Further analysis suggests that the performance saturates after 200 best performing visual concepts. Thirdly, multiple late fusion models are investigated, including weighted average, Ada Boost, linear regression and linear SVM [Duda et al., 2012, Hastie et al., 2009].

Cao et al. [2011] propose to pool descriptors along each scene using a *Scene Aligned Model*, such that the classification models are adaptive to different scenes. In this model, scenes are detected using the K-means clustering [Bishop, 2009], applied to GIST descriptors of all training data. At test time, the GIST descriptor of a frame contributes to each pool proportionally to its soft-assignment to scene clusters.

Bao et al. [2011] (see also [Lan et al., 2012]) experimented with three fusion schemes. For the early fusion, they prefer the uniform averaging of the kernel matrices over the MKL. The former is significantly faster, while the performance is almost the same in their experiments. For the late fusion, Bao et al. [2011] use both uniform weights and a logistic regression classifier. However, due to overfitting, only the uniform weights were used in the primary run. Additionally, Bao et al. [2011] propose *double fusion* — a method that chains early fusion with late fusion. First, combinations of channels are fused at the early fusion stage. Late fusion then combines the scores of these combinations with the scores of individual channels. Double fusion improves the classification performance over the results of the early and late fusions.

For pooling Bao et al. [2011] rely on Spatial Pyramid Matching [Lazebnik et al., 2006], that preserves the information about locations of local descriptors. In addition to non-linear SVM, Kernel Regression and Sequential Boosting showed competitive performance.

TRECVID MED 2012 Common conclusions of 2012 year participants were the following. Firstly, low-level descriptors excellently suit for multimedia event detection. Secondly, semantic concepts provide complementary information, especially in the setting with few training examples. Thirdly, text information, although being rarely present in videos, allows to achieve high precision. Therefore, positive evidences for Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR) channels improve the overall retrieval performance.

Most works focused on improving the spatial pooling of descriptors. Besides, the focus of the competition itself included training with few positive examples (10 videos), Ad-Hoc evaluation, and a concurrent Multimedia Event Recounting task. In the Ad-Hoc task, the features are computed before the 5 additional categories are revealed.

Natarajan et al. [2012a] report high quality results based on 18 robust low-level features and complementary high-level concept classifiers. In addition to standard local features for appearance, color and motion, they rely on kernel descriptors that generalize hand-designed features. Three types of event concepts are used: video-level and segment-level, and Classemes [Torresani et al., 2010] models. Video-level concepts are automatically

extracted from textual video descriptions using NLP techniques. Concept classifiers are then learned using visual and audio descriptors. Segment-level concepts are trained in a similar way, but relied on manual frame annotations. In the end, Natarajan et al. [2012a] use the vector of concept classifier scores for training event detectors. The same fusion techniques are used as for TRECVID 2011 competition [Natarajan et al., 2011].

Yu et al. [2012] use 6 standard local descriptors together with novel Motion SIFT and Acoustic Unit Detectors. They investigate two different kinds of local feature aggregation and two types of pooling. In the basic scheme, local features are encoded with the BoW model, combined with two pooling variants: standard Spatial Pyramids, and feature and event-specific pooling. The latter improves the performance, especially for particular events. Yu et al. [2012] also experiment with Gaussian Mixture Model Super Vector [Chatfield et al., 2011] feature aggregation. Fusion and classifier techniques are almost identical to the previous year's system [Bao et al., 2011]. The system generalizes well to new categories, showing similar performance on Pre-Specified and Ad-Hoc tasks.

Cheng et al. [2012] propose Fixed-Pattern spatial feature pooling, which is a fast alternative of the Spatial Pyramid model. A separate classifier is trained on each spatial cell, and the video score is built on the scores of individual classifiers. Apart from local features, Cheng et al. [2012] rely on 1800 concepts and investigate different techniques for concept pooling. They consider use both data-relevant (extracted from the event category definitions) and data-irrelevant concepts (taken from existing datasets). ASR and OCR information is integrated using the video-level fusion, such that only high scores contribute to the final score.

TRECVID MED 2013 The 2013 year competition had the setting similar to the previous year: 20 Pre-Specified events, 10 Ad-Hoc events, training with few examples, complementary MER task. The evaluation metric was changed to a simpler Average Precision, which does not require threshold selection. Additionally to Full evaluation, participants reported results separately for each of four modalities: Visual, Audio, ASR, OCR. We omit the discussion of methods used for 0-examples scenario, since they mostly rely on NLP and retrieval techniques.

Compared to previous year, the top performing systems 1) use Fisher Vectors to aggregate local descriptors, 2) report improved performance of semantic concepts, comparable to that of the low-level descriptors, 3) empirically combine multiple late fusion techniques.

The evaluation of different modalities suggests that the visual content provided the majority of the evidence, while the audio, ASR and OCR provided complementary evidence.

Natarajan et al. [2013] improve their previous year system [Natarajan et al., 2012a] by adding the dense trajectory-based local feature. They focus on learning semantic concepts in a weakly-supervised way. Semantic concepts are detected in category definitions using NLP techniques and the videos are labeled based on textual video descriptions. Irrelevant concepts are pruned two times: first, based on the number of positive examples, and second, based on the classification performance. Additional concepts are detected by crawling web resources like Flickr and Youtube, ranking the images with event classifiers. Tags from relevant images are included into concept pool. For each event category, relevant concepts are selected using an SVM.

Another sources of high-level information include Classemes detectors, video-adapted ASR and OCR. Fusion of multiple features is similar to double fusion [Yu et al., 2012], but also includes weighted average fusion.

In addition to SVM models, Natarajan et al. [2013] also experimented with query-based detections. Each concept is described by its mean in the feature space. During search, the distance to the mean of relevant concepts is then used to rank the search data collection.

Lan et al. [2013] use a new type of semantic concepts based on deep networks, which outperforms the best low-level features. The Deep Convolutional Neural Network [Bengio, 2009] model is trained for 1000 image categories on the ImageNet 2012 database. The video-level scores are computed as a sum of scores for keyframes.

Lan et al. [2013] perform a rigorous comparison of multiple late fusion techniques. They improve the previous year's double fusion [Yu et al., 2012] by learning the weights as compared to the uniform average. The best performing method uses leave-one-out performance to rank the features. Finally, the average of 10 different fusion methods gives additional improvement.

An ablation study suggests that MFCC is highly complementary to other channels. On the contrary, new types of audio features are of less utility.

For feature aggregation, Spatial Bag of Features (BoF), Super Vectors (SV) and Fisher Vectors (FV) are used. FVs improve the performance for dense trajectory descriptors, but they are not complementary to BoF and SV when used with multiple features.

Average Z-score fusion integrates the scores of multiple representations and classification methods.

In total, the whole feature extraction is about 2 times faster than realtime on 300 cores [Lan et al., 2013].

Chapter 3

Category-specific video summarization

Abstract

In large video collections with clusters of typical categories, such as "birthday party" or "flash-mob", category-specific video summarization can produce higher quality video summaries than unsupervised approaches that are blind to the video category.

Given a video from a known category, our approach first efficiently performs a temporal segmentation into semantically-consistent segments, delimited not only by shot boundaries but also general change points. Then, equipped with an SVM classifier, our approach assigns importance scores to each segment. The resulting video assembles the sequence of segments with the highest scores. The obtained video summary is therefore both short and highly informative. Experimental results on videos from the multimedia event detection (MED) dataset of TRECVID'11 show that our approach produces video summaries with higher relevance than the state of the art.

Publication

Danila Potapov, Matthijs Douze, Zaid Harchaoui, Cordelia Schmid. *Category-specific video summarization.* European Conference on Computer Vision (ECCV), Zürich, 2014

Dataset:

http://lear.inrialpes.fr/people/potapov/med_summaries

3.1 Introduction

Most videos from YouTube or DailyMotion consist of long-running, poorly-filmed and unedited content. Users would like to browse, i.e., to *skim through* the video to quickly get a hint on the semantic content. Video summarization addresses this problem by providing a short video summary of a full-length video. An ideal video summary would include all the important video segments and remain short in length. The problem is extremely challenging in general and has been the subject of recent research [Liu et al., 2010, de Avila et al., 2011, Lee et al., 2012, Wang et al., 2012, Khosla et al., 2013, Lu and Grauman, 2013].

FIGURE 3.1: Original video, and its video summary for the category "birthday party".

Large collections of videos contain clusters of videos belonging to specific categories with typical visual content and repeating patterns in the temporal structure. Consider a video of a "birthday party" (see Figure 3.1). It is unclear how an unsupervised approach for video summarization would single out the short segments corresponding to "blow the candles", "applause", etc. Such a summary serves as a proof to the user that the video comes from the "birthday party" category, without the need of watching the whole video.

In this chapter, we describe a category-specific summarization approach. A first distinctive feature of our approach is the temporal segmentation algorithm. While most previous works relate segment boundaries to shot boundaries, our temporal segmentation algorithm detects general change points. This includes shot boundaries, but also sub-shot boundaries where the transitions between sub-shots are gradual. A second feature is the category-specific supervised importance-scoring algorithm, which scores the *relative importance* of segments within each category, in contrast to video-specific importance [Liu et al., 2010, de Avila et al., 2011, Truong and Venkatesh, 2007].

Our approach works as follows (see Figure 3.2). First, we perform an automatic kernel-based temporal segmentation based on state-of-the-art video features that automatically selects the number of segments. Then, equipped with an SVM classifier for importance scoring that was trained on videos for the category at hand, we score each segment in terms of importance. Finally, the approach outputs a video summary composed of the segments with the highest predicted importance scores. Thus, our contributions are threefold:

- we propose a novel approach, Kernel Video Summarisation (**KVS**), for supervised video summarization of realistic videos, that uses state-of-the-art image and video features
- we introduce a new dataset, **MED-Summaries**[1], along with a clear annotation protocol to evaluate video summarization
- we obtain excellent experimental results on MED-Summaries, showing that KVS delivers video summaries with higher overall importance, as measured by two performance metrics.

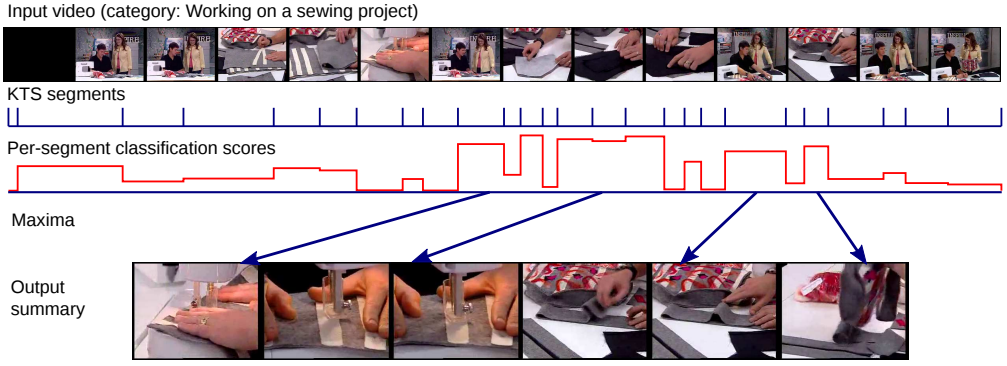


FIGURE 3.2: Overall scheme of Kernel Video Summarization (KVS).

3.2 Related work

Video summarization. Truong & Venkatesh [Truong and Venkatesh, 2007] present a comprehensive overview and classification of video summarization methods. The task is difficult to define and many methods are domain-specific (sports, news, rushes, documentary, etc.). However, to our knowledge, there are no publicly available implementations or datasets, for eg. sports videos summarization, that could be used for comparison

[1]The annotations and the evaluation codes are available at http://lear.inrialpes.fr/people/potapov/med_summaries.

with more recent approaches. Summaries may focus on dominant concepts [Over et al., 2008], relate to the video's story [Lu and Grauman, 2013], the user's preferences, the query context [Wang et al., 2012], or user attention [Ma et al., 2005]. A video is either summed up as a sequence of keyframes [Lee et al., 2012, Khosla et al., 2013, de Avila et al., 2011] or by video excerpts [Lu and Grauman, 2013].

Video summarization received much attention when NIST was running the TRECVID Rushes summarization task (2006-2008). The evaluation was conducted on a dataset of significant size, with an expensive manual annotation of the ground-truth [Over et al., 2008]. However, the methods were mostly specific to the domain, i.e. they focused on detecting redundant shots of a scene, and clapperboards.

For professional and low-dynamic TV broadcast videos (e.g. from [Over et al., 2008, Wang et al., 2012] or Open Video Archive), shot boundaries naturally split a video into "visual sentences". Early summarization methods [Truong and Venkatesh, 2007] extract one or more keyframes to represent a shot, often independently from the other shots. Recent works, including this one, focus on user-generated data [Lee et al., 2012, Khosla et al., 2013, Lu and Grauman, 2013, Li et al., 2012], which typically do not contain shot boundaries.

Without supervision, summarization methods must rely on low-level indices to determine the relevance of parts of a video [Ma et al., 2005, Ngo et al., 2005, Divakaran et al., 2003]. When the video domain is known, summarization can be strongly supervised. For example, soccer games [Xie et al., 2004, Rui et al., 2000] or feature films [Sundaram et al., 2002] have standard phases that can be manually identified. A few previous works [Lee et al., 2012, Lu and Grauman, 2013, Khosla et al., 2013] produced summaries using features crafted for specific visual categories. In contrast to these works, our approach builds short yet highly informative category specific video summaries, using generic state-of-the-art visual features.

In [Zhao and Xing, 2014, Cong et al., 2012], the main task is to remove redundant video footage, which is detected as easy to reconstruct based on sparse coding from the rest of the video. A recent work [Li et al., 2012] also segments a video at a finer level than shots and relies on supervised mutual information to identify the important segments. The main difference of our work is the use of state-of-the-art video features and the quantitative evaluation of the approach. Leveraging crawled internet photos is another recent trend for video summarization [Kim et al., 2014, Khosla et al., 2013].

There are several ways of evaluating video summarization methods [Truong and Venkatesh, 2007]. Most works [Lee et al., 2012, Lu and Grauman, 2013, Khosla et al., 2013, Ngo et al., 2005] conduct *user studies* to compare different summaries of the same video.

	year	#videos	#users/video	duration	annotation	#classes	data type
UT Egocentric (1)	2012	4	2	17 h	important region masks	-	Egocentric
SumMe (2)	2014	25	15-18	1.1 h	multiple summaries	-	User videos
Youtube Highlight (3)	2014	≈600	5	24 h	best highlight, selected content	6	User videos
MED Summaries (4)	2014	140	2-4	6 h	temp. segments and importance	10	User videos

TABLE 3.1: Publicly available video summarization datasets. (1) [Lee et al., 2012], (2) [Gygli et al., 2014], (3) [Sun et al., 2014], (4) [Potapov et al., 2014].

The *concept coverage* metric evaluates the number of important objects or actions included in the summary [Lee et al., 2012, Over et al., 2008]. Although it requires time-consuming manual annotation of videos, the annotations can be reused to evaluate multiple approaches. When the goal is to simplify video navigation, the time it takes a user to perform some data exploration task can be used as a quality metric [Over et al., 2008]. *Automatic comparison to reference summaries* comes from text summarization literature [Lin, 2004]. It relies on a user-generated summary of a video and a metric to compare it to the algorithm's summary [Khosla et al., 2013, de Avila et al., 2011, Kim et al., 2014]. The protocol used in this work combines concept coverage with a comparison to multiple reference summaries.

Video summarization datasets.

Lee et al. [2012] proposed a dataset of egocentric videos. Such videos are filmed with a camera mounted on a person's head, to imitate the video stream seen by humans. There are 4 test videos publicly available, 17 hours in total. The videos were captured by 4 different people while performing daily activities. Additionally, the dataset provides approximately 1660 spatial important object segmentations and a set of negative frames, which are non-important for the summary.

Recently Sun et al. [2014] introduced a new summarization dataset with around 600 videos (1430 minutes in total), from 6 domains: "skating", "gymnastics", "dog", "parkour", "surfing", and "skiing". For nearly 60% of videos the dataset provides highlight annotations, with a consensus reached by at least 3 out of 5 annotators. Sun et al. [2014] also proposed to rely on edited videos, which are supposed to contain only the interesting parts of the raw videos. Interestingly, such videos can be queried among the videos created with the "Youtube video editor" using the Youtube API.

Gygli et al. [2014] proposed another dataset for summarization of user generated videos. There are 25 videos, from 1 to 6 minutes, with 15-18 ground truth annotations from different people. Human consistency was validated using the Cronbach psychometric test [Gygli et al., 2014], and is reported as good on average over the dataset.

Table 3.1 compares recent publicly available summarization datasets.

In a related task of unsupervised discovery of action segments / parts, existing approaches rely on discriminatively trained part based models [Niebles et al., 2010, Raptis and Sigal, 2013]. Similar temporal modeling could be useful for the problem of category-specific video summarization, though a lack of obvious temporal structure was reported for Trecvid videos [Wang et al., 2015].

Temporal video segmentation. Computer vision methods often utilize spatial or temporal segmentation to raise the abstraction level of the problem and reduce its dimensionality. Segmentation can help to solve image classification, scene reconstruction [Hoiem et al., 2005] and can serve as a basis for semantic segmentation [Tighe and Lazebnik, 2010]. Similarly, video segmentation usually implies dividing a video into spatio-temporal volumes [Lezama et al., 2011, Grundmann et al., 2010]. Temporal video segmentation often means detecting shot or scene boundaries, that are either introduced by the "director" through editing or simply correspond to filming stops.

The proliferation of user-generated videos created a new challenge for semantic temporal segmentation of videos. Lee et al. [Lee et al., 2012] used clustering of frame color histograms to segment temporal events. In [Lu and Grauman, 2013] a video is split in sub-shots depending on the activity of the wearer of a head-mounted camera: "static", "moving the head" or "in transit". Similar to these works we focus on the content of the segment rather than its boundaries.

Most shot boundary detection methods focus on differences between consecutive frames [Massoudi et al., 2006], relying on image descriptors (pixel color histograms, local or global motion [Truong and Venkatesh, 2007], or bag-of-features descriptors [Chasanis et al., 2009]). Our temporal segmentation approach takes into account the differences between *all pairs of frames*. Therefore, the approach allows to single out not only shot boundaries but also *change points* in general that correspond to non-abrupt boundaries between two consecutive segments with different semantic content.

3.3 Kernel video summarization

We start by giving definitions of the main concepts and building blocks of our approach.

3.3.1 Video summary

A video is partitioned into segments. A *segment* is a part of the video enclosed between two timestamps. A *video summary* is a video composed of a subset of the temporal segments of the original video.

FIGURE 3.3: Illustration of the importance notion on the "Changing a vehicle tire" category. These frames come from a 1-minute video where a support car follows a cyclist during a cycle race. The main event — changing a bicycle tire — takes less than one third of the video. The figure shows central frames of user-annotated segments together with their importance score.

A *summary* is a condensed synopsis of the whole video. It conveys the most *important* details of the original video. A segment can be non-informative due to signal-level reasons like abrupt camera shake and dark underexposed segments commonly present in egocentric videos [Lee et al., 2012, Lu and Grauman, 2013].

A segment can be considered *important* due to multiple reasons, depending on the video category and application goals: highlights of sport matches, culmination points of movies [Truong and Venkatesh, 2007], influential moments of egocentric videos [Lu and Grauman, 2013].

We make the assumption that the notion of importance can be learned from a set of videos belonging to the video category. This point of view stems from the Multimedia Event Recounting task at TRECVID: selecting segments containing evidence that the video belongs to a certain event category. Similarly, we define importance as a *measure of relevance to the type of event*. Figure 3.3 shows an example video together with the importance of its segments.

Our definition of importance spans an ordinal scale, ranging from 0 "no evidence" to 3 "the segment alone could classify the video into the category". More details are given in Sec. 3.4.1.

The proposed method, Kernel Video Summarisation (**KVS**), decomposes into three steps: i) kernel temporal segmentation; ii) importance-scoring of segments; iii) summary building. Figure 3.2 summarizes our approach.

3.3.2 Overview of Kernel Temporal Segmentation

Kernel Temporal Segmentation (KTS) allows split the video into a set of non-intersecting temporal segments. The method is fast and accurate when combined with high-dimensional

descriptors.

Our temporal segmentation approach is a kernel-based change point detection algorithm. In contrast to shot boundary detection, change point detection is a more general statistical framework [Kay, 1998]. Change point detection usually focuses on piecewise constant one-dimensional signals corrupted by noise, and the goal is to detect the jumps in the signal. It is able to statistically discriminate between jumps due to noise and jumps due to the underlying signal. Change-point detection has been subject of intense theoretical and methodological study in statistics and signal processing; see [Kay, 1998, Harchaoui et al., 2008] and references therein. Such methods enjoy strong theoretical guarantees, in contrast to shot boundary techniques that are mostly heuristic and tuned to the types of video transitions at hand (cut, fade in/out, etc.). We propose here a retrospective multiple change-point detection approach, based on [Harchaoui and Cappé, 2007], that considers the whole signal at once. A similar Sequence Reconstruction Error approach is well known for constructing keyframe summaries [Truong and Venkatesh, 2007]. In contrast to this group of methods, while we also search for the best piecewise approximation in the feature space, the proposed approach only targets the best temporal segmentation, regardless of the keyframe positions within the segments.

Given the matrix of frame-to-frame similarities defined through a positive-definite kernel, the algorithm outputs a set of optimal "change points" that correspond to the boundaries of temporal segments. More precisely, let the video be a sequence of descriptors $\mathbf{x}_i \in \mathbf{X}$, $i = 0, \ldots, n-1$.

Let $K : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ be a kernel function between descriptors. Let $\mathcal{H}$ be the feature space of the kernel $K(\cdot, \cdot)$. Denote $\phi : \mathbf{X} \to \mathcal{H}$ the associated feature map, and $\|\cdot\|_{\mathcal{H}}$ the norm in the feature space $\mathcal{H}$. We minimize the following objective

$$\underset{m;\, t_0,\ldots,t_{m-1}}{\text{Minimize}} \quad J_{m,n} := L_{m,n} + Cg(m,n) \tag{3.1}$$

where m is the number of change points and $g(m,n)$ a penalty term (see below). $L_{m,n}$ is defined from the within-segment kernel variances $v_{t_i,t_{i+1}}$:

$$L_{m,n} = \sum_{i=0}^{m} v_{t_{i-1},t_i}, \quad v_{t_i,t_{i+1}} = \sum_{t=t_i}^{t_{i+1}-1} \|\phi(x_t) - \mu_i\|_{\mathcal{H}}^2, \quad \mu_i = \frac{\sum_{t=t_i}^{t_{i+1}-1} \phi(x_t)}{t_{i+1} - t_i} \tag{3.2}$$

Automatic calibration. The number of segments could be set proportional to the video duration, but this would be too loose. Therefore, the objective of Equation (3.1) decomposes into two terms: $L_{m,n}$ which measures the overall within-segment variance, and $g(m,n)$ that penalizes segmentations with too many segments. We consider a BIC-type penalty [Hastie et al., 2009] with the parameterized form $g(m,n) = m(\log(n/m) +$

1) [Arlot et al., 2012]. Increasing the number of segments decreases $L_{m,n}$ (3.2), but increases the model complexity. This objective yields a trade-off between under- and over-segmentation. We propose to cross-validate the C parameter using a validation set of annotated videos. Hence we get kernel-based temporal segmentation algorithm where the number of segments is set automatically from data.

Algorithm 1 Kernel temporal segmentation

Input: temporal sequence of descriptors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$	Cost
1. Compute the Gram matrix A : $a_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$	$dn^2/2$
2. Compute cumulative sums of A	n^2
3. Compute unnormalized variances $v_{t,t+d} = \sum_{i=t}^{t+d-1} a_{i,i} - \frac{1}{d}\sum_{i,j=t}^{t+d-1} a_{i,j}$ $t = 0, \ldots, n-1, \quad d = 1, \ldots, n-t$	$2n^2$
4. Do the forward pass of dynamic programming $L_{i,j} = \min_{t=i,\ldots,j-1}\big(L_{i-1,t} + v_{t,j}\big), \quad L_{0,j} = v_{0,j}$ $i = 1, \ldots, m_{\max}, \quad j = 1, \ldots, n$	$2m_{\max}n^2$
5. Select the optimal number of change points $m^\star = \arg\min_{m=0,\ldots,m_{\max}} L_{m,n} + Cg(m,n)$	$2m_{\max}$
6. Find change-point positions by backtracking $t_{m^\star} = n, \quad t_{i-1} = \arg\min_t\big(L_{i-1,t} + v_{t,t_i}\big)$ $i = m^\star, \ldots, 1$	$2m^\star$
Output: Change-point positions $t_0, \ldots, t_{m^\star-1}$	

Algorithm. The proposed algorithm is described in Algo. 1. First, the kernel is computed for each pair of descriptors in the sequence. Then, the segment variances are computed for each possible starting point t and segment duration d. It can be done efficiently by precomputing the cumulative sums of the matrix [Crow, 1984, Viola and Jones, 2004]. After that, dynamic programming is used to minimize the objective (3.2). It iteratively computes the best objective value for the first j descriptors and i change points. Finally, the optimal segmentation is reconstructed by backtracking. The total runtime cost of the algorithm is in $O(m_{\max}n^2)$. The penalization introduces a minimal computational overhead because dynamic programming already computes $L_{i,n}$ for all possible segment counts.

Step 3 in Algo. 1 is efficiently computed as follows. First we compute the cumulative sums.

$$s'_i = \sum_{i'=0}^{i-1} a_{i',i'}, \quad i = 1, \ldots, n, \quad s'_0 = 0 \tag{3.3}$$

$$s''_{i,j} = \sum_{i'=0}^{i-1}\sum_{j'=0}^{j-1} a_{i',j'}, \quad i,j = 1, \ldots, n, \quad s''_{\cdot,0} = 0, \quad s''_{0,\cdot} = 0. \tag{3.4}$$

An efficient way to compute the cumulative sums is described in [Crow, 1984, Viola and Jones, 2004]. Then, step 2 in Algo. 1 requires only 6 array element accesses:

$$v_{t,t+d} = s'_{t+d} - s'_t - \frac{1}{d}(s''_{t+d,t+d} - s''_{t,t+d} - s''_{t+d,t} + s''_{t,t}) \tag{3.5}$$

3.3.3 Properties of Kernel Temporal Segmentation

We refer to Algorithm 1 for Kernel Temporal Segmentation as A_1 for brevity. A_1 has the following properties:

1. **(Optimality)** Let $\theta^* = \{m^*; t_0, \ldots, t_{m^*-1}\}$ be a solution returned by A_1. Optimization objective (3.1) attains in θ^* the exact global optimum.
2. **(Termination)** Algorithm A_1 stops after $O(n^2)$ operations.

Proof (Optimality):
To disambiguate the notation, we will refer to $L_{m,n}$ as defined in Step 4 of A_1, not in formula (3.2). Let us define for $j = 1, \ldots, n$ (we assume $t_{-1} = 0$):

$$f_j(t_0, \ldots, t_{m-1}) := \sum_{i=0}^{m-1} v_{t_{i-1},t_i} + v_{t_{m-1},j} \tag{3.6}$$

$$f_j^*(m) := \min_{\substack{t_0,\ldots,t_{m-1} \\ 0<t_0<\ldots<t_{m-1}<j}} f_j(t_0, \ldots, t_{m-1}) \tag{3.7}$$

Then the objective (3.1) writes as:

$$\min_{\substack{m;\, t_0,\ldots,t_{m-1} \\ 0<t_0<\ldots<t_{m-1}<n}} f_n(t_0, \ldots, t_{m-1}) + Cg(m, n) \tag{3.8}$$

To prove the optimality property we will show that Step 3 of A_1 computes the function $f_j^*(i)$.

From Definitions (3.6)-(3.7): $\forall j = 1, \ldots, n \quad \forall m = 1, \ldots, m_{\max} \quad$ s.t. $\quad m < j$:

$$f_j^*(m) = \min_{m-1<t_{m-1}<j} \quad \min_{\substack{t_0,\ldots,t_{m-2} \\ 0<t_0<\ldots<t_{m-1}}} f_j(t_0, \ldots, t_{m-1}) \tag{3.9}$$

$$= \min_{m-1<t_{m-1}<j} \quad \min_{\substack{t_0,\ldots,t_{m-2} \\ 0<t_0<\ldots<t_{m-1}}} \sum_{i=0}^{m-2} v_{t_{i-1},t_i} + v_{t_{m-2},t_{m-1}} + v_{t_{m-1},j} \tag{3.10}$$

$$= \min_{m-1<t<j} \quad \min_{\substack{t_0,\ldots,t_{m-2} \\ 0<t_0<\ldots<t}} \sum_{i=0}^{m-2} v_{t_{i-1},t_i} + v_{t_{m-2},t} + v_{t,j} \tag{3.11}$$

and we finally get

$$f_j^*(m) = \min_{m-1<t<j} f_t^*(m-1) + v_{t,j} \ . \tag{3.12}$$

We also note that by definition $f_j^*(0) = v_{0,j}$. Therefore we state that Step 3 of A_1 computes $L_{i,j} = f_j^*(i)$, because reccurrent functions $L_{i,j}$ and $f_j^*(i)$ are identical, which proves the optimality property for the fixed m.

At Step 4, A_1 selects the optimal m^*, because the penalty term $Cg(m,n)$ in (3.8) does not depend on change-point positions. Finally, the optimal change-points are identified for the fixed value m^* at Step 5.

Proof (Termination):
On the right of Algorithm 1 we show the computational complexity of each step. We assume the constant $m_{\max}$ to be fixed. Then the algorithm stops after $O(n^2)$ operations.

3.3.4 Learning to predict importance scores

For each category, we train a linear SVM classifier from a set of videos with video-level labels, assuming that a classifier originally trained to classify the full videos can be used to score importance of small segments. This assumption is reasonable for videos where a significant proportion of segments have high scores. The opposite case, when a very small number of segments allow to classify the video ("needle in a haystack"), is outside the scope of this work.

At training time, we aggregate frame descriptors of a video as if the whole video was a single segment. In this way a video descriptor has the same dimensionality as a segment descriptor. For each category we use videos of the category as positive examples and the videos from the other categories as negatives. We train one binary SVM classifier per category.

At test time, we segment the video using the KTS algorithm and aggregate Fisher descriptors for each segment. The relevant classifier is then applied to the segment descriptors, yielding a 1D signal which is the *importance map* of the video.

In order to evaluate the summarization separately from the classification, we assume that the category of the video is known in advance. While recent methods specifically targeted at video classification [Oneata et al., 2013, Cao et al., 2012] are rather mature, relying on them for our evaluation would introduce additional noise.

3.3.5 Summary building with Kernel Video Summarisation

Finally, a summary is constructed by concatenating the most important segments of the video. We assume that the duration of the summary is set a priori. Segments are included in the summary by the order of their importance until the duration limit is achieved (we crop the last segment to satisfy the constraint).

3.4 MED-summaries dataset

Most existing works evaluate summaries based on user studies, which are time-consuming, costly and hard to reproduce.

We introduce a new dataset, called **MED-summaries**. The proposed benchmark simplifies the evaluation by introducing a clear and automatic evaluation procedure, that is tailored to category-specific summarization. Every part of the video is annotated with a category-specific importance value. For example, for the category "birthday party", a segment that contains a scene where someone is blowing candles is assigned a high importance, whereas a segment just showing children around a table is assigned a lower importance.

We use the training set of the TRECVID 2011 Multimedia Event Detection dataset (12,249 videos) to train the classifier for importance scoring. Furthermore, we select 60 videos from this training set as a validation set and annotate them. To test our approach we annotate 100 videos from the official test set (10 per class), where most test videos have a duration from 1 to 5 minutes. Annotators mark the temporal segments and their importance; the annotation protocol is described in section 3.4.1. To take into account the variability due to different annotators, annotations were made by several people. In the experimental section we evaluate our results with respect to the different annotations and average the results. The different metrics for evaluation are described in section 3.4.3. See the dataset's website for details.

3.4.1 Annotation protocol

3.4.1.1 Segment annotation.

The annotation interface shows one test video at a time, which can be advanced by steps of 5 frames. First, we ask a user to annotate temporal segments. Temporal segments should be *semantically consistent*, i.e. long enough for a user to grasp what is going on, but it must be possible to describe it in a short sentence. For example it can be "a

group of people marching in the street" for a video of the class "Parade", or "putting one slice of bread onto another" for the class "Making a sandwich".

Some actions are repetitive or homogeneous, e.g. running, sewing, etc. In that case we ask to specify the "period" — minimum duration of a sub-segment that fully represents the whole segment. For example, watching 2-3 seconds of a running person is sufficient to describe the segment as "a person is running".

We require all shot boundaries to be annotated as change points, but change points do not necessarily correspond to shot boundaries. Often a shot contains a single action, but the main part is shorter than the whole segment. In this case we ask to localize precisely the main part.

3.4.1.2 Importance annotation.

For each semantic segment we ask a user *"Does the segment contain evidence of the given event category?"*. The possible answers are:

0: No evidence
1: Some hints suggest that the whole video could belong to the category
2: The segment contains significant evidence of the category
3: The segment alone classifies the video to the category

TRECVID 2011 Multimedia Event Detection [Over et al., 2011] dataset provides textual descriptions for each event category. The descriptions were shown to users as a reference for importance annotation. As an example the "Birthday party" category is defined as follows:

Definition: An individual celebrates a birthday with other people
Explication: A birthday in this context is the anniversary of a person's birth. Less commonly, the term "birthday" can be used to refer to the anniversary of an organization's establishment, but a celebration for an organization does not satisfy the event definition.
A birthday celebration is a gathering of people who have been invited by the host or hosts to come to a set location (often a private home, sometimes a restaurant, bar, nightclub, park, or other public venue) to socialize in honor of the person(s) whose birthday it is (the birthday celebrant(s)).
Birthday parties, as with other parties/celebrations, will typically feature an assortment of food and beverages. Birthday parties are often accompanied by colorful decorations, such as balloons and streamers, and some people may wear cone-shaped "birthday hats". [...]
Evidential description:
scene: indoors (a home, a restaurant) or outdoors (backyard, park); day or night
objects/people: decorations (balloons, streamers, conical hats, etc), birthday cake (often with candles), birthday celebrant, guests, gifts
activities: singing, blowing out candles on cake, playing games, eating, opening gifts
audio: singing "Happy Birthday to You"; saying happy birthday; laughing; sounds of games being played

While audio can be used during annotation, we specify that if something is only mentioned in onscreen text or speech, then it should not be labeled as important.

In preliminary experiments we found that annotators tend to give too high importance to very short segments, that often have ambiguous segmentation and importance score. Therefore, we preprocess the ground-truth before the evaluation — we decrease the annotated importance for segments smaller than 4 seconds proportionally to the segment duration.

3.4.2 Annotation interface

In order to get the ground-truth annotation of temporal segments and importance, we developed a web-based interface. Figure 3.4 shows a screenshot right after the annotation is finished. The interface allows quick and precise navigation within the video, which is essential both for the segment annotation and the importance annotation. We defined multiple keyboard shortcuts to speed up the annotation. However, navigating the video during the segment annotation is much easier using a computer mouse.

3.4.3 Evaluation metrics

We represent the manually annotated ground-truth segments $\mathbf{S} = \{S_1, \ldots, S_n\}$ of a video by:

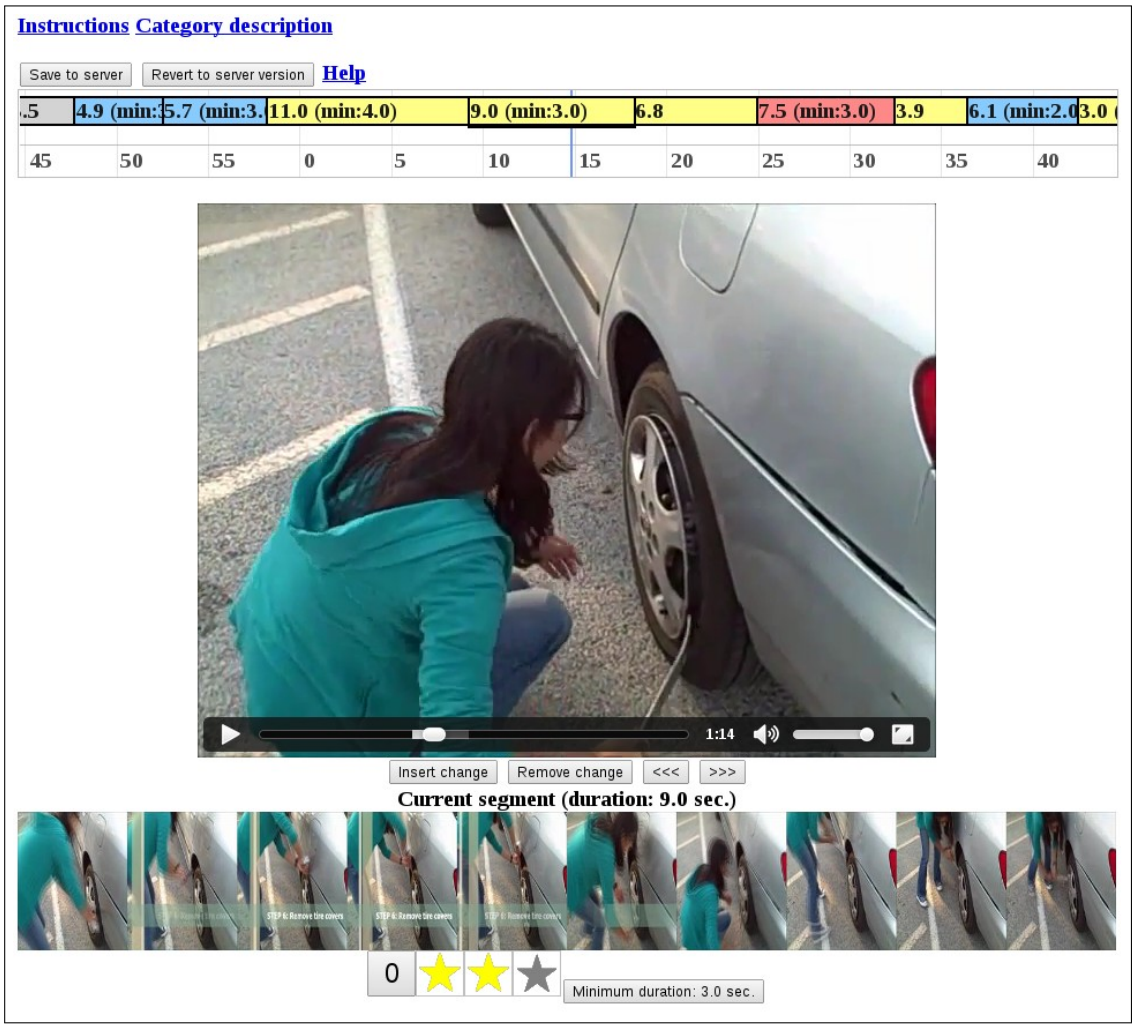
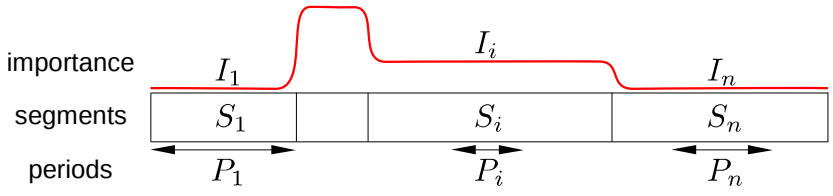


FIGURE 3.4: Interface for the annotation of temporal segments and importance. Video from "Changing a vehicle tire" category.



An automatic temporal segmentation is represented by the sequence of segments $\mathbf{S}' = \{S'_1, \ldots, S'_m\}$.

To evaluate **segmentation** we define a symmetric f-score metric as:

$$f(\mathbf{S}, \mathbf{S}') = \frac{2 \cdot p(\mathbf{S}, \mathbf{S}') \cdot p(\mathbf{S}', \mathbf{S})}{p(\mathbf{S}, \mathbf{S}') + p(\mathbf{S}', \mathbf{S})}, \tag{3.13}$$

where the similarity of two segmentations $\mathbf{A}$ and $\mathbf{B}$ is

$$p(\mathbf{A},\mathbf{B}) = \frac{1}{|\mathbf{A}|}|\{A \in \mathbf{A} \text{ st. } \exists B \in \mathbf{B} \text{ matching } A\}| \tag{3.14}$$

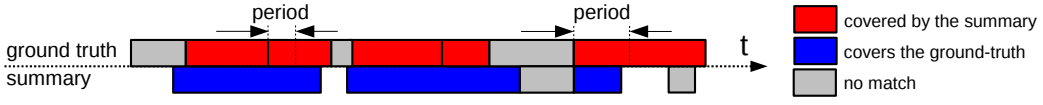
where $|\mathbf{A}|$ is the number of segments in $\mathbf{A}$. We consider segments A and B are matching if the temporal overlap over the union ratio is larger than 0.75, and when a segment has an annotated period, it is reduced to a sub-segment no shorter than the period, that maximizes the overlap over the union.

To evaluate **summarization** we define two metrics: the importance ratio and the meaningful summary duration.

A computed summary is a subset of the segments $\widetilde{\mathbf{S}} = \{\widetilde{S}_1, \cdots, \widetilde{S}_{\tilde{m}}\} \subset \mathbf{S}'$. We say a ground truth segment S_i is *covered* by a detected segment $\widetilde{S}_j$ if

$$\text{duration}\big(S_i \cap \widetilde{S}_j\big) > \alpha P_i \tag{3.15}$$

When the period equals the segment duration this means that a fraction α of the ground truth segment is covered by the detected segment. We use $\alpha = 80\%$ to enforce visually coherent summaries, which was validated using the ground-truth. Note that this definition allows covering several ground truth segments by a single detected segment, as in the following example:



Let $C(\widetilde{\mathbf{S}}) \subset \mathbf{S}$ be the subset of ground truth segments covered by the summary $\widetilde{\mathbf{S}}$. Given the duration of the summary $\mathcal{T}(\widetilde{\mathbf{S}}) = \sum_{j=1}^{\tilde{m}} \text{duration}(\widetilde{S}_j)$ and its total importance $\mathcal{I}(\widetilde{\mathbf{S}}) = \sum_{i \in C(\widetilde{\mathbf{S}})} I_i$, we define the *importance ratio* as

$$\mathcal{I}^*(\widetilde{\mathbf{S}}) = \frac{\mathcal{I}(\widetilde{\mathbf{S}})}{\mathcal{I}^{\max}(\mathcal{T}(\widetilde{\mathbf{S}}))}, \quad \text{with} \quad \mathcal{I}^{\max}(T) = \max_{\substack{\mathbf{A} \subset \mathbf{S} \\ \mathcal{T}(\mathbf{A}) \leq T}} \text{s.t.} \; \mathcal{I}(\mathbf{A}) \tag{3.16}$$

We use the *maximum possible summary importance* $\mathcal{I}^{\max}(T)$ as a normalization factor. This normalization takes into account the duration and the redundancy of the video and ensures that $\mathcal{I}^*(\widetilde{\mathbf{S}}) \in [0, 1]$.

It turns out that maximizing the summary importance given the ground-truth segmentation and importance is NP-hard, as it is a form of knapsack problem. Therefore we use a greedy approximate summarization: we reduce each segment to its period, sort the segments by decreasing importance (resolving ties by favoring shorter segments), and

constructing the optimal summary from the top-ranked segments that fit in the duration constraint.

A second measure is the *meaningful summary duration*: **MSD**. A meaningful summary is obtained as follows. We build it by adding segments by order of classification scores until it covers a segment of importance 3, as defined by the ground-truth annotation. This guarantees that the gist of the input video is represented at this length and measures how relevant the importance scoring is. Summaries assembling a large number of low-importance segments first are mediocre summaries and get a low MSD score. Summaries assembling high-importance segments first get a high MSD score. In our experiments we report the median MSD score over all test videos as a performance measure.

3.5 Results

3.5.1 Baselines

As the videos are annotated by several users, we can evaluate their annotations with respect to each other in a leave-one-out manner (**Users**). This quantifies the task's ambiguity and gives an upper bound on the expected performance.

For segmentation we use a shot detector (**SD**) of Massoudi et al. [Massoudi et al., 2006] as a baseline. *For classification* we use two baselines: one with the shot detector, where shots are classified with an SVM (**SD+SVM**) and one where the segments are selected by clustering instead of SVM scores (**KTS+Cluster**).

The SD+SVM baseline is close to an event detection setup, where a temporal window slides over the video, and an SVM score is computed for every position of the window [Oneata et al., 2013, Gaidon et al., 2013]. However, we pre-select promising windows with the SD segmentation.

Clustering descriptors produces a representative set of images or segments of the video, where long static shots are given the same importance as short shots [Khosla et al., 2013]. We use a simple k-means clustering, as the Fisher Vectors representing segments (see next section) can be compared with the L2 distance [Perronnin et al., 2010]. The summary is built by adding one segment from each cluster in turn. First we add segments nearest to each centroid, ordered by increasing duration, then second nearest, etc.

Our KVS method combines the KTS segmentation with a SVM classifier.

TABLE 3.2: Evaluation of segmentation and summarization methods on the test set of 100 videos. The performance measures are average f-measure for segmentation (higher is better) and median Meaningful Summary Duration for summarization (lower is better).

Method	Segmentation Avg. f-score	Summarization MSD (s)
Users	49.1	10.6
SD + SVM	30.9	16.7
KTS + Cluster	**41.0**	13.8
KVS	**41.0**	**12.5**

3.5.2 Details of implementation

3.5.2.1 Video descriptors & classifier.

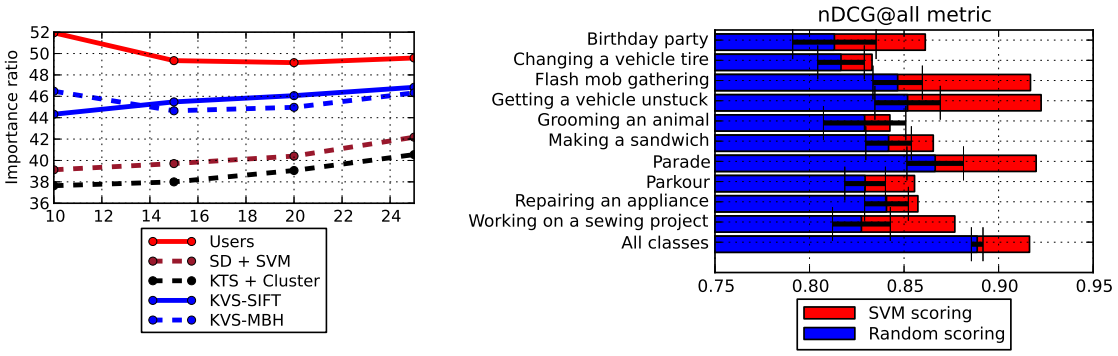
We process every 5-th frame of the video. We extract SIFT descriptors on a dense grid at multiple scales. The local descriptors are reduced to 64 dimensions with PCA. Then a video frame is encoded with a Fisher Vector [Perronnin et al., 2010] based on a GMM of 128 Gaussians, producing a d=16512 dimension vector.

For segmentation we normalize frame descriptors as follows. Each dimension is standardized within a video to have zero mean and unit variance. Then we apply signed square-rooting and L_2 normalization. We use dot products to compare Fisher vectors and produce the kernel matrix. Even though primal formulation is applicable in this case, precomputation of the kernel matrix reduces the memory usage when the features are high-dimensional.

For classification, the frame descriptors from a segment are whitened under the diagonal covariance assumption as in [Perronnin et al., 2010]. Then we apply signed square-rooting and L_2-normalization. The segment descriptor is the average of the frame descriptors. This was shown to be the best pooling method for frame descriptors [Oneata et al., 2013, Cao et al., 2012].

The linear SVM classifier for each class is built from about 150 positive and 12000 negative training videos from the MED 2011 training dataset. The C parameter of the classifier is optimized using cross-validation.

We use grid-search on the 60-video validation set to optimize the parameters of the different methods. The shot detector (**SD**) has a single threshold T. Our **KVS** method relies on a single parameter C that controls the number of segments (equation 3.1). For the clustering method, the optimal ratio of the number of clusters over the number of segments was found to be $1/5^{\text{th}}$.



(A) (B)

FIGURE 3.5: Summarization of the 100-video test dataset. (a) Importance ratio of Equation (3.16) for different durations of the summary. (b) Correlation of SVM scores and scores assigned by users.

On average, the annotated segments are 3.5 s long, and so are **SD** segments. The **KTS** method produces segments of 4.5 s on average.

3.5.3 Segmentation

Table 3.2 shows the segmentation quality of users and algorithms. For algorithms we average the f-scores of Equation (3.13) over segmentations from different users. For users we report the average f-score of the leave-one-out evaluation, i.e. we assume each user in turn to be the ground truth. The proposed approach KTS outperforms the competing method SD in terms of temporal segmentation performance. Surely, human segmentations are better than the algorithms', which means that the annotation protocol is consistent. Yet, the average f-score of users is not close to 100%, which suggests that the segment annotation task is somewhat subjective.

3.5.4 Summarization

The MSD metric in Table 3.2 shows that the temporal segmentation output by KTS has a significant impact on the summary's quality. Indeed, the SD+SVM method generally produces longer summaries than KTS+Cluster.

Figure 3.5a shows the summarization quality for different summary durations. The user curve gives an upper bound on what can be achieved, by evaluating the consensus between annotators, following the leave-one-out procedure as before. The proposed approach, KVS, is the closest to the user curve. Again, KVS clearly outperforms the competing methods KTS+Cluster and SD+SVM. Figure 3.7 illustrates our approach.

We also run an experiment where the SIFT low-level descriptor is replaced by the MBH motion descriptor [Wang et al., 2013]. We use MBH descriptors in a similar setting as SIFT — descriptors are reduced to 64 dimensions with PCA and assigned to a GMM with 128 Gaussians. We compute Fisher Vectors with derivatives w.r.t. weights, means and variances. Fisher Vectors are whitened analytically and then normalized in a standard way (signed square-rooting and L2-normalization). [2] Figure 3.5a shows that SIFT and MBH features are quite close in the summarization task. We get 2% improvement for 10 second summaries and 1% drop for longer summaries compared to SIFT. A recent work [Oneata et al., 2013] also reports little difference between SIFT and MBH on the MED 2011 dataset.

Summarization experiment with complete supervision. Using full videos for training a classifier is a simple approach for summarization that does not require additional annotation. Here we investigate how much gain in performance we can get given the segmentation and importance annotated in the training videos. The annotation is available only for test videos. Therefore we split videos into 10 folds (10 videos per class, 1 per fold) and do 10-fold cross-validation. We train a classifier using the segments with importance 3 as positives (usually more than 1 per video) and full videos as negatives. Then, the "importance ratio" is computed on the held-out video. The baseline classifier, that uses full videos as training examples, is trained and evaluated in the same cross-validation setup. We split the videos in folds in the same way, therefore the same videos are used for training in both cases. When using important parts, each positive video in the training set is replaced by its important segments. The C parameter is selected each time using 3-fold cross-validation. We use SIFT features in both cases.

Table 3.3 shows the cross-validation results. When trained from the important parts, we get around 3 points higher importance ratio.

	E006	E007	E008	E009	E010	E011	E012	E013	E014	E015	Average
full videos	33.9	29.0	37.7	29.2	40.3	29.4	37.5	37.0	30.7	46.2	35.1 +/- 4.8
important parts	35.9	38.8	40.2	33.5	41.7	33.9	37.6	36.1	35.4	44.8	37.8 +/- 4.9

TABLE 3.3: Summarization experiment on test set (100 videos). Standard deviation is computed over the cross-validation folds, after averaging over the classes. See text for details.

Further experiments showed that the gap of 3 points is not very stable and depends on the number of cross-validation folds for selecting the C parameter and the randomization seed. However, in all the experiments learning from important parts always gave better

[2] Videos are rescaled to have a width of 200 pixels and cropped to have a ratio 3:4. We use an old version of dense trajectories without foreground-background separation.

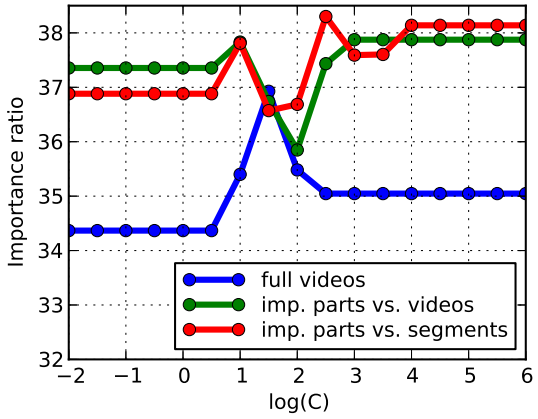


FIGURE 3.6: Importance ratio for various values of C, shared by all classes. The same test set of 100 videos is used. See text for details.

result. Figure 3.6 shows the cross-validation performance on the same test set when the SVM C parameter is fixed and shared by all classes.

There is an unusual drop when learning from important parts that can be explained by the negatives coming from a different distribution (i.e. segment descriptors versus video descriptors). To verify this hypothesis we do an additional experiment where we use all the segments from the other classes as negative examples (cf. Figure 3.6, "imp. parts vs. segments"). Although the drop in the plot is still present, it is now smaller.

Segment level evaluation. Although the *importance ratio* metric directly addresses the summarization task, it is sensitive to segmentation mismatches. Since the segmentation is not perfect (Table 3.2), it is interesting to test the *importance scoring* mechanism alone. Do SVM scores correlate with the annotated importance scores?

We sort all the segments by descending SVM score. Ideally, the segments with importance 3 should be in the top of the list, and non-relevant segments in the bottom. This is a ranking problem where segments (analogue of *images* in image retrieval) have graded importance scores (*relevance*) — from 0 to 3. We use the *normalized discounted cumulative gain* (nDCG) ranking metric [Manning et al., 2008]

$$\text{nDCG} = Z_p^{-1} \sum_{i=1}^{p} I^{(i)} (\log_2 i)^{-1},$$

where $I^{(i)}$ is the annotated importance score of the i^{th} segment in the ranked list; p is the total number of segments over all videos of the class; Z_p is the normalization factor such that a perfect ranking's nDCG is 1.

Figure 3.5b shows that, for 9 out of 10 classes, the SVM ranking is stronger than the random ranking, and also better when considering all videos.

In this experiment we use the annotations of 1 user per video on the test set of 100 videos. We use 1000 trials to get the nDCG of random scoring. Evaluation is done on all ground-truth segments — for a total 3705 segments.

3.6 Conclusion

We proposed a novel approach to video summarization, called Kernel Video Summarisation. The approach delivers short and highly-informative summaries, that assemble the most important segments for a given video category.

Kernel Video Summarisation requires a set of training videos for a given category so that the method can be trained in a supervised fashion, but does not rely on segment annotations in the training set. We also introduced a new dataset for category-specific video summarization, MED-Summaries, that is publicly available, along with the annotations and the evaluation code that computes the performance metrics introduced in this work.

FIGURE 3.7: Illustrations of summaries obtained with Kernel Video Summarization. We show the central frame in each segment with the SVM score below.

Chapter 4

Beat-Event Detection in Action Movie Franchises

Abstract

While important advances were recently made towards temporally localizing and recognizing specific human actions or activities in videos, efficient detection and classification of long video chunks belonging to semantically-defined categories such as "pursuit" or "romance" remains challenging.

We introduce a new dataset, called **Action Movie Franchises**, consisting of a collection of Hollywood action movie franchises. We define 11 non-exclusive semantic categories — called **beat-categories** — that are broad enough to cover most of the movie footage. The corresponding **beat-events** are annotated as groups of video shots, possibly overlapping. We propose an approach for localizing beat-events based on classifying shots into beat-categories and learning the temporal constraints between shots. We show that temporal constraints significantly improve the classification performance. We set up an evaluation protocol for beat-event localization as well as for shot classification, depending on whether movies from the same franchise are present or not in the training data.

Publication

Danila Potapov, Matthijs Douze, Jerome Revaud, Zaid Harchaoui, Cordelia Schmid. *Beat-Event Detection in Action Movie Franchises*, arXiv, 2015

Dataset: http://lear.inrialpes.fr/people/potapov/action_movies

FIGURE 4.1: Example frames for categories of the Action Movie Franchises dataset.

4.1 Introduction

Automatic understanding and interpretation of videos is a challenging and important problem due to the massive increase of available video data, and the wealth of semantic variety of video content. Realistic videos include a wide variety of actions, activities, scene type, etc. During the last decade, significant progress has been made for action retrieval and recognition of specific, stylized, human actions. In particular, powerful visual features were proposed towards this goal [Oneata et al., 2013, 2014, Wang and Schmid, 2013]. For more general types of events in videos, such as activities, efficient approaches were proposed and benchmarked as part of the TRECVID Multimedia Event Detection (MED) competitions [Over et al., 2014]. State-of-the-art approaches combine features from all modalities (text, visual, audio), static and motion features (possibly learned beforehand with deep learning), and appropriate fusion procedures.

In this work, we aim at detecting events of the same semantic level as TRECVID MED, but on real action movies that follow a structured scenario. From a movie script writer's point of view [Snyder, 2005], a Hollywood movie is more or less constrained to a set of standard story-lines. This standardization helps matching the audience expectations and habits. However, movies need to be fresh and novel enough to fuel the interest of the audience. So, some variability must be introduced in the story lines to maintain the interest. Temporally, movies are subdivided in a hierarchy of acts, scenes, shots, and finally, frames (see Figure 4.2). Punctual changes in the storyline give it a rhythm.

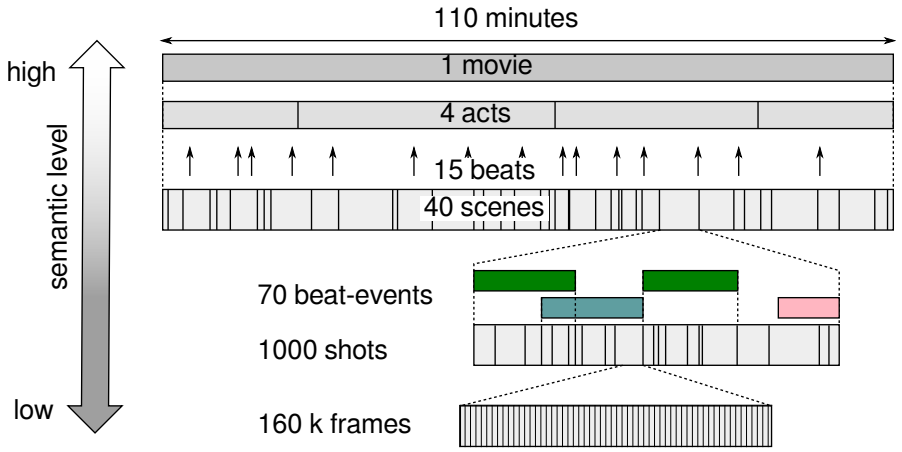


FIGURE 4.2: Temporal structure of a movie, according to the taxonomy of "Save the Cat" [Snyder, 2005], and our level of annotation, the beat-event.

They are called "beats" and are common to many films. A typical example of beat is the moment when an unexpected solution saves the hero.

From a computer vision point of view, frames are readily available and reliable algorithms for shot detection exist. Grouping shots into scenes is harder. Scenes are characterized by a uniform location, set of characters or storyline. The semantic level of beats and acts is out of reach. We propose here to attack the problem on an intermediate level by detecting "beat-events". Temporally, they consist in sequences of consecutive shots and typically last a few minutes. Shots offer a suitable granularity, because movies are edited so that they follow the rhythm of the action. Semantically, they are of a higher level than the actions in most current benchmarks, but lower than the beats, which are hard to identify even for a human.

For the purpose of research, we built an annotated dataset of Hollywood action movies, called **Action Movie Franchises**. It comprises 20 action movies from 5 franchises: *Rambo*, *Rocky*, *Die Hard*, *Lethal Weapon*, *Indiana Jones*. A movie franchise refers to a series of movies on the same "topic", sharing similar story lines and the same characters. In each movie, we annotate shots into several non-exclusive beat-categories. We then create a higher level of annotation, called beat-events, which consists of consistent sequences of shots labeled with the same beat-category.

Figure 4.1 illustrates the beat-categories that we use in the Action Movie Franchises dataset. They are targeted at action movies and, thus, rely on semantic categories that often reply on the role of the characters, such as hero (good) or villain (bad). We now briefly describe all categories. First, we define three different action-related beat-categories: *pursuit*, *battle preparation* and *battle*, shown in the first row of Figure 4.1. We also define categories centered on the emotional state of the main characters: *romance*, *despair good* (*e.g.* when the hero thinks that all is lost) and *joy bad* (*e.g.* when

the villain thinks he won the game), see second row of Figure 4.1. We also include different categories of dialog between all combinations of good and bad characters: *good argue good*, *good argue bad* and *bad argue bad* (third row of Figure 4.1). Finally, we add two more categories notifying a temporary victory of a good or bad character (*victory good* and *victory bad*, last row of Figure 4.1). We also consider a NULL category, corresponding to shots that can not be classified into any of the aforementioned beat-categories. Table 4.1 shows a mapping of the beats to beat-events.

In summary, we introduce the **Inria Action Movie** dataset, which features dense annotations of 11 beat-categories in 20 action movies at both shot and event levels. To the best of our knowledge, a comparable dense annotation of videos does not exist.

The semantic level of our beat-categories will drive progress in action recognition towards new approaches based on human identity, pose, interaction and semantic audio features. State-of-the-art methods are without doubt not sufficient for such categories. Action movies and related professionally produced content account for a major fraction of what people watch on a daily basis. There exists a large potential for applications, such as access to video archives and movie databases, interactive television and automatic annotation for the shortsighted.

Furthermore, we define several evaluation protocols, to investigate the impact of franchise-information (testing with or without previously seen movies from the same franchise) and the performance for both classification and localization tasks. We also propose an approach for classification of video shots into beat-categories based on a state-of-the-art pipeline for multimodal feature extraction, classification and fusion. Our approach for localizing beat-events uses a temporal structured inferred by a conditional random field (CRF) model learned from training data.

We make the Action Movie Franchises dataset publicly available to the community to further advance research on automatic video understanding. [1]

4.2 Related work

Related datasets. Table 4.2 summarizes recent state-of-the-art datasets for action or activity recognition. Our Action Movie Franchises dataset mainly differs from existing ones with respect to the event complexity and the density of annotations. Similarly to Coffee & Cigarettes and MediaEval Violent Scene Detection (VSD), our Action Movie Franchises dataset is built on professional movie footage. However, while the former datasets only target short and sparsely occurring events, we provide dense annotations

[1] The dataset is online: http://lear.inrialpes.fr/people/potapov/action_movies

Blake Snyder's **Beats**	Corresponding Beat-Events
Opening image (p. 1): Sets the tone for the story and suggests the protagonist's primary problem.	-
Theme is stated (p. 5): A question or statement, usually made to the protagonist, indicating the story's main thematic idea.	-
Set-up (p. 1-10): An introduction to the main characters and setting—the background.	-
Catalyst (p. 12): A major event that changes the protagonist's world and sets the story in motion.	-
Debate (p. 12-25): A question is raised about the choice now before the protagonist. Often this section lays out the stakes for the journey ahead.	**good-argue-bad** **good-argue-good** **bad-argue-bad**
Break into Act II (p. 25-30): The hero definitively leaves his old world or situation and enters a strange new one.	**pursuit (DH4,LW3,Ra1)** good-argue-good (Ro1)
B-story (p. 30): A secondary plotline that often fleshes out side characters—frequently a mentor or a love interest—who assist the hero on his journey.	**romance**
Fun and games (p. 30-55): Snyder says this section offers "the promise of the premise." It's an exploration of the story's core concept that gives the story its "trailer-friendly moments." It's usually lighter in tone, and it typically builds to a big victory at the midpoint.	**romance** good-argue-good (Ro1,Ro2) battle (Ro3)
Midpoint (p. 55): The A and B stories cross. The story builds to either a false victory or (less often) false defeat. New information is revealed that raises the stakes.	**victory-good** **victory-bad**
Bad guys close in (p. 55-75): After the victory at the midpoint, things grow steadily worse as the villains regroup and push forward.	**battle (DH4,IJ3,Ra2,Ro3)** **joy-bad (DH1,DH4,Ro1)** **despair-good (DH1,Ra2)** pursuit (DH1)
All is lost (p. 75): Mirroring the midpoint, it's usually a false defeat. The hero's life is in shambles. Often there's a major death or at least the sense of death—a reference to dying or mortality somehow.	**victory-bad** **despair-good**
Dark night of the soul (p. 75-85): A moment of contemplation in which the hero considers how far he's come and all he's learned. It's the moment in which the hero asks, "Why is all this happening?"	**despair-good (Ra4,Ro4,Ro3)**
Break into Act III (p. 85) A "Eureka!" moment that gives the hero the strength to keep going—and provides the key to success in Act III.	**romance (Ro1,Ro4)** **despair-good (LW1,LW2,LW3)** good-argue-good (Ro3)
Finale (p. 85-110) Relying on all he has learned throughout the story, the hero solves his problems, defeats the villains, and changes the world for the better.	**battle** **victory-good** **pursuit (Ra4,Ra2,DH4)**
Final image (p. 110). A mirror of the opening image that underlines the lessons learned and illustrates how the world has changed.	**victory-good** **romance (IJ2, Ro1, LW3, DH4)** **good-argue-good (IJ3, LW3)** **despair-good (DH1,DH2,LW3)** victory-bad (Ro1, Ra1)

ActionMovies **Beat-Events**
pursuit: villains are following heroes, or the opposite (it usually takes some time)
battle: confrontation between good/bad characters (usually includes fighting or shooting)
romance: between the hero and love interest
victory good: good characters win a battle
victory bad: bad characters win a battle
preparation: preparing to the battle - setting up the armor, training, jogging, etc.
despair good: desperate mood of good heroes, normally not during fight, but connected to the global battle
joy bad: villains express emotions (usually laugh)
good argue bad: heroes and villains have an oral debate
good argue good: good characters argue among each other
bad argue bad: bad characters argue among each other

TABLE 4.1: Mapping of the beats to beat-events. One of the conclusions of this work is that the global temporal structure is not constrained by a well defined beat-event order. Therefore the mapping here relies more on the definition of beats and beat-events, than on the temporal ordering. The bold beat-event matches are common and represent prominent beats; the other are more special cases. The beginning of the movie is less standardized and contains little action, therefore is hard to describe in terms of well-defined events. Note that *preparation* does not match precisely any beat, but always happens right before the final battle in the *Rocky* franchise. In parentheses we show the abbreviated movie names where each match happens.

of beat-events spanning larger time intervals. Our beat-categories are also of significantly higher semantic level than those in action recognition datasets like Coffee & Cigarettes, UCF [Soomro et al., 2012] and HMDB [Kuehne et al., 2011]. A consequence is that our dataset remains very challenging for state-of-the-art algorithms, as shown later in the experiments. Events of a similar complexity can be found in TRECVID MED

Name	# classes	example class	annotation unit	# positive train units	durations avg unit	annot	NULL	coverage
Classification								
UCF 101 [Soomro et al., 2012]	101	high jump	clip	13320	7.21s	26h39	0h	-
HMDB 51 [Kuehne et al., 2011]	51	brush hair	clip	6763	3.7s	6h59	0h	-
TRECVID MED 11	15	birthday party	clip	2650	2m54	128h	315h	29%
Action Movie Franchises	11	good argue bad	shot	16864	5.4s	25h29	15h42	57.1%
Localization								
Coffee & Cigarettes	2	drinking	time interval	191	2.2s	7m12s	3h26	3.3%
THUMOS detection 2014	20	floor gymnastics	t.i. on clip	3213	26.2s	3h22	167h54	2.0%
MediaEval VSD [Demarty et al., 2014]	10	fighting	shot/segment	3206	3.0s	2h38	55h20	4.5%
Action Movie Franchises	11	good argue bad	beat-event	2906	35.7s	28h49	14h08	61.4%

TABLE 4.2: Comparison of classification and localization datasets.
Legend: positive train units —number of positive training units (excluding NULL); annot. — total duration of all annotated parts; NULL — duration of the non-annotated (NULL or background) footage; coverage = proportion of annotated video footage.

2011–2014 [Over et al., 2014], but our dataset includes precise temporally localized annotations.

Action detection in movies. Action detection (or action localization), that is finding if and when a particular type of action was performed in long and unsegmented video streams, received a lot of attention in the last decade. The problem was considered in a variety of settings: from still images [Raptis and Sigal, 2013], from videos [Gaidon et al., 2013, Wang and Schmid, 2013], with or without weak supervision, etc. Most works focused on highly stylized human actions such as "open door", "sit down", which are typically *temporally salient* in the video stream.

Action or activity recognition can often be boosted using temporal reasoning on the sequence of atomic events that characterize the action, as well as the surrounding events that are likely to precede or follow the action/activity of interest. We shall only review here the "temporal context" information from surrounding events; the decomposition of action or activities into sequence of atomic events [Gaidon et al., 2013] is beyond the scope of this work. Early works along this line [Rui et al., 1998] proposed to group shots and organize groups into "semantic" scenes, each group belonging exclusively to only one scene. Results were evaluated subjectively and no user study was conducted.

Several works relied on movie (or TV series) scripts to leverage the temporal structure [Everingham et al., 2006, Marszalek et al., 2009]. In [Marszalek et al., 2009], movie scripts are used to obtain scene and action annotations. Aligning of the movie scripts with the movie is often not accurate, because the final version of the script is available not for all movies. Thus, we did not use movie scripts to build our dataset and do not consider this information for training and testing. However, we do use another modality, the audio track, in a systematic way, and perform fusion following state-of-the-art

approaches in multimedia [Li et al., 2004], and TRECVID competitions [Over et al., 2014].

In [Cour et al., 2008], the authors structure a movie into a sequence of scenes, where each scene is organized into interlaced threads. An efficient dynamic programming algorithm for structure parsing is proposed. Experimental results on a dataset composed of TV series and a feature-length movie are provided. More recently, in [Bojanowski et al., 2013], actors and their actions are detected simultaneously under weak supervision of movies scripts using discriminative clustering. Experimental results on 2 movies (*Casablanca* and *American beauty*) are presented, for 3 actions (*walking*, *open door* and *sit down*). The approach improves person naming compared to previous methods. In this work, we do not use supervision from movie scripts to learn and uncover the temporal structure, but rather learn it directly using a conditional random field that takes SVM scores as input features. The proposed approach is more akin to [Hoai et al., 2011], where joint segmentation and classification of human actions in video is performed on toy datasets [Hoai and De la Torre, 2012].

4.3 The Action Movie Franchises Dataset

We first describe the Action Movie Franchises dataset and the annotation protocol. Then, we highlight some striking features in the structure of the movies observed during and after the annotation process. Finally, we propose an evaluation protocol for shot classification into beat-categories and for beat-event localization.

4.3.1 Action Movie Franchises

The **Action Movie Franchises** dataset consists of 20 Hollywood action movies belonging to 5 famous franchises: *Rambo*, *Rocky*, *Die Hard*, *Lethal Weapon*, *Indiana Jones*. Each franchise comprises 4 movies, see Table 4.2 for summary statistics of the dataset.

Each movie is decomposed into a list of shots, extracted with a shot boundary detector [Massoudi et al., 2006, Potapov et al., 2014]. Each shot is tagged with zero, one or several labels corresponding to the 11 beat-categories (the label NULL is assigned to shots with zero labels). Note that the total footage for the dataset is 36.5 h, shorter than the total length in Table 4.2. This is due to multiple labels. All categories are shown in Figure 4.1.

Series of shots with the same category label are grouped together in *beat-events* if they all depict the same scene (ie. same characters, same location, same action, etc.).

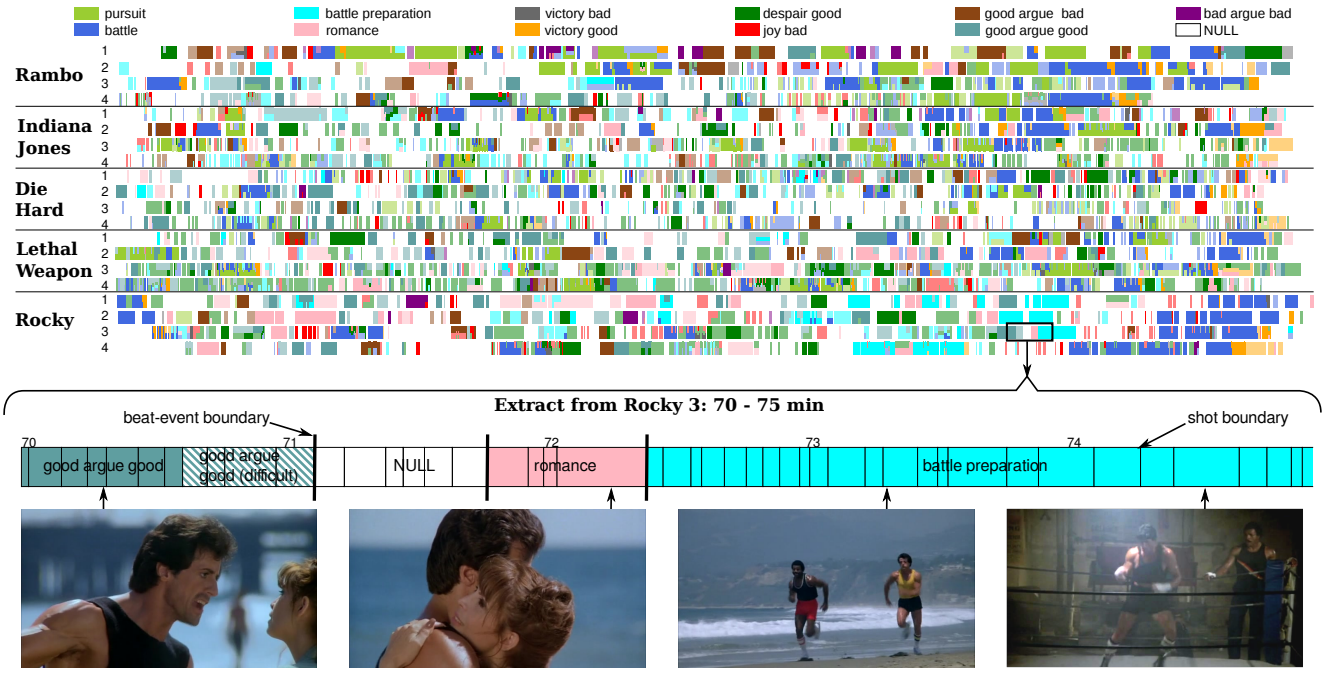


FIGURE 4.3: Top: Beat-events annotated for the Action Movie Franchises dataset, one movie per line, plotted along the temporal axis. All the movies were scaled to the same length. Bottom: zoom on a movie extract showing the shot segmentation, the annotations and the beat-events. Best viewed onscreen.

Temporally, we also allow a beat-event to bridge gaps of a few unrelated shots. Beat-events belong to a single, non-NULL, beat-category.

The set of categories was inspired by the taxonomy of [Snyder, 2005], and motivated by the presence of common narrative structures and beats in action movies. Indeed, category definitions strongly rely on a split of the characters into "good" and "bad" tags, which is typical in such movies. Each category thus involves a fixed combination of heroes and villains: both "good" and "bad" characters are present during *battle* and *pursuit*, but only "good" heroes are present in the case of *good argue good*.

Large intra-class variation is due to a number of factors: duration, intensity of action, objects and actors, and different scene locations, camera viewpoint, filming style. For ambiguous cases we used the "difficult" tag.

4.3.2 Annotation protocol

We consider the following 11 classes (*beat-categories*):

- *pursuit*: villains are following heroes, or the opposite (it usually takes some time)
- *battle*: confrontation between heroes and villains (usually includes fighting/shooting)

- *romance*: between the hero and the love interest
- *victory good*: good characters win a battle
- *victory bad*: bad characters win a battle
- *preparation*: preparing to the battle — setting up the armor, training, jogging, etc.
- *despair good*: desperate mood of heroes, normally not during fight, but connected to the global battle
- *joy bad*: villains express emotions (usually laugh)
- *good argue bad*: good and bad characters have an oral debate
- *good argue good*: good characters argue among each other
- *bad argue bad*: bad characters argue among each other

We allow beat-events of different classes to temporally overlap. If possible, beat-event boundaries are selected such that the event is recognizable from the segment alone.

Beat-event definitions are as much uniform over all the movies as possible. E.g. there are few debates annotated in Rambo, since most of the debates are less prominent than in Rocky. The division of characters into "good" and "bad" is fixed per movie. Note that it can however change within a particular franchise.

Each movie is first temporally segmented into a sequence of shots using a *shot boundary detector* [Massoudi et al., 2006, Potapov et al., 2014]. Shot boundaries correspond to transitions between different cameras and/or scene locations. The minimal temporal unit in the annotation process is a *shot*, so temporal boundaries of beat-events always coincide with shot boundaries. Shots can be annotated with zero, one, or several category labels. Shots without an annotation are assigned a NULL label. All occurrences are annotated, so NULL shots are negative instances for each of the 11 beat-event categories.

The annotation process was carried out in two passes by three researchers. Ambiguous cases were discussed and resulted in a clear annotation protocol. In the first pass we manually annotated each shot with zero, one or several of the 11 beat-category labels. In the second one we annotated the beat-events by specifying their category, beginning and ending shots. We tolerated gaps of 1-2 unrelated shots for sufficiently consistent beat-events. Indeed, movies are often edited into sequences of interleaved shots from two events, *e.g.* between the main storyline and the "B" story. We fill the gaps in the groups of shots that semantically belong to a single beat-event. Note that, in this way, temporal boundaries of beat-events always coincide with shot boundaries.

Some annotations are labeled as "difficult", if they are semantically hard to detect, or ambiguous. For instance, in *Indiana Jones 3*, Indiana Jones engages in a romance with

Dr. Elsa Schneider, who actually betrays him to the "bad guy". Romance between Indiana Jones and Dr. Elsa Schneider is therefore ambiguous. We exclude these shots at training and evaluation time, as in the Pascal evaluation protocol [Everingham et al., 2010]. More details on the annotation protocol are given in Appendix B.

Our beat-event annotations cover about 60 % of the movie footage, which is much higher than comparable datasets, see Table 4.2. This shows that the vocabulary we chose is representative: the dataset is annotated densely , in constrast to the Coffee and Cigarettes [Laptev and Pérez, 2007] and MediaEval VSD [Demarty et al., 2014] datasets where less than 5% of the footage is annotated.

4.3.3 Structure of action movies

Figure 4.3 shows the sequence of category-label annotations for several movies. Some global trends are striking: *victory good* occurs at the end of movies; *battle* is most prevalent in the last quarter of movies; there is a pause in fast actions (*battle*, *pursuit*) around the middle of the movies. In movie script terms, this is the "midpoint" beat [Snyder, 2005], where the hero is at a temporary high or low in the story. In terms of beat-event duration, *joy bad* and *victory bad* are short, while *pursuit* and *romance* are long.

After careful analysis of the annotation, we find that *battle*, *despair good* and *pursuit* are the most prevalent beat-categories, with 4145, 3042 and 2416 instances respectively. Since it is a semantically high level class, *despair good* is most often annotated as difficult. The co-occurrences of classes as annotations of the same shot follow predictable trends: *battle* co-occurs with *pursuit*, *battle preparation*, *victory good* and *victory bad*. Interestingly *romance* is often found in combination with *despair good*. This is typical for movies of the "Dude with a problem" type [Snyder, 2005], where the hero must prove himself.

Within each movie franchise, a shared structure may appear. For instance, in *Rocky*, the *battle preparation* occurs in the last quarter of the movie, and there is no *pursuit*.

4.3.4 Evaluation protocol

In the following, we propose two types of train/test splits and two performance measures for our Action Movie Franchises dataset.

Data splits. We consider two different types of splits over the 20 movies, see Figure 4.4. They both come in 5 folds of 16 training movies and 4 test movies. All movies

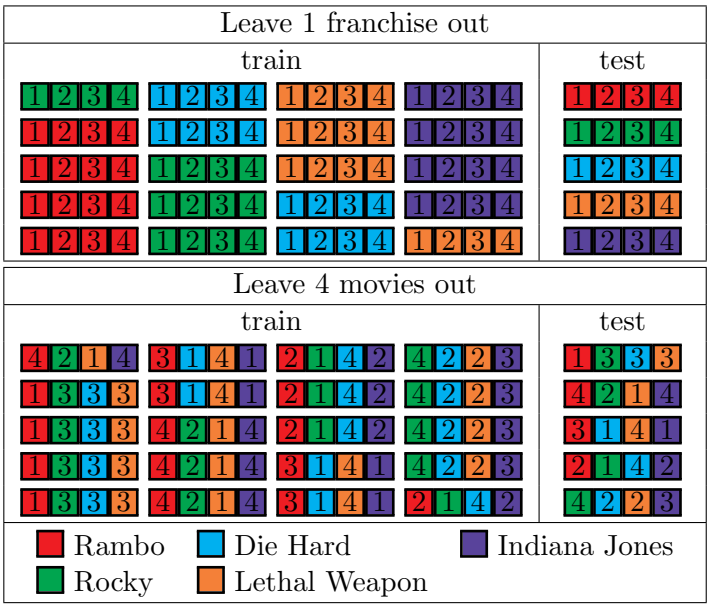


FIGURE 4.4: The two types of split for evaluation. Movie franchises are color-coded. In addition to the train/test splits, the training videos are also split in 4 *sub-folds*, that are used for cross-validation and CRF training purposes.

appear once as a test movie. In the "leave one franchise out" setting, all movies from a single franchise are used as a test set. In "leave 4 movies out", a single movie from each franchise is used as test. This allows to evaluate if our classifiers are specific to a franchise or generalize well across franchises.

Classification setting. In the classification setting, we evaluate the accuracy of beat-category prediction at the shot level. Since a shot can have several labels, we adopt the following evaluation procedure. For a given shot with $n > 0$ ground-truth labels (in general $n = 1$, but the number of labels can be up to 4), we retain the best n predicted beat-categories (out of 11, according to their confidence scores). Accuracy is then measured independently for each beat-category as the proportion of ground-truth shots which are correctly labeled. We finally average accuracies over all categories, and report the mean and the standard deviation over the 5 cross-validation splits.

Localization setting. In the localization setting, we evaluate the temporal agreement between ground-truth and predicted beat-events for each beat-category. A detection, consisting of a temporal segment, a category label and a confidence score, is tagged positive if there exists a ground-truth beat-event with an intersection-over-union score [Everingham et al., 2010] over 0.2. If the ground-truth beat-event is tagged as "difficult" it does not count as positive nor negative. The performance is measured for each beat-category in terms of average precision (AP) over all beat-events in the test fold, and the different APs are averaged to a mAP measure.

4.4 Shot and beat-event classification

The proposed approach consists of 4 stages. First, we compute high-dimensional shot descriptors for different visual and audio modalities, called *channels*. Then, we learn linear SVM classifiers for each channel. At the late fusion stage, we take the linear combination of the channel scores. Finally, predictions are refined by leveraging the temporal structure of the data and beat-events are localized.

4.4.1 Descriptors extraction

For each shot from a movie, we extract different descriptors corresponding to different modalities. For this purpose, we use a state-of-the art set of low-level descriptors [Aly et al., 2013, Oneata et al., 2013]. It includes still image, face, motion and audio descriptors:

Dense SIFT [Lowe, 2004] descriptors are extracted every 30'th frame. The SIFTs of a frame are aggregated into a Fisher vector of 256 mixture components, that is power- and L2-normalized [Perronnin et al., 2010]. The shot descriptor is the power- and L2 normalized average of the Fisher descriptors from its frames. The output descriptor has 34559 dimensions.

Convolutional neural nets (CNN) descriptors are extracted from every 30'th frame. We run the image through a CNN [Krizhevsky et al., 2012] trained on Imagenet 2012, using the activations from the first fully-connected layer as a description vector (FC6 in 4096 dimensions). The implementation is based on DeCAF [Donahue et al., 2013] and its off-the-shelf pre trained network.

Motion descriptors are extracted for each shot. We extract improved dense trajectory descriptors [Wang and Schmid, 2013]. The 4 components of the descriptor (MBHx, MBHy, HoG, HoF) are aggregated into 4 Fisher vectors that are concatenated. This output is a 108544 D vector.

Audio descriptors are based on MFCC [Rabiner and Schafer, 2007] extracted for 25 ms audio chunks with a step of 10 ms. They are enhanced by adding first and second order temporal derivatives. The MFCCs are aggregated into a shot descriptor using a Fisher aggregation, producing a 20223 D vector.

Face descriptors are obtained by first detecting faces in each frame using the Viola-Jones detector from OpenCV [Bradski, 2000]. Following the approach of Everingham et al. [2006], we join the detections into face tracks using the KLT tracker, allowing us to recover some missed detections. Each facial region is then described with a Fisher vector of dense SIFTs [Simonyan et al., 2013] (16384 dimensions) which is power- and L2-normalized. Finally, we average-pool all face descriptors within a shot and normalize

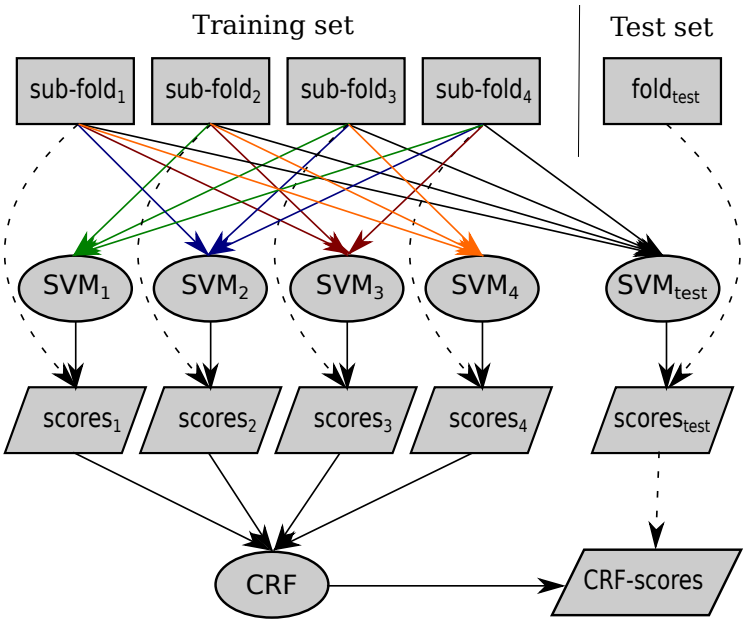


FIGURE 4.5: Proposed training approach for one fold. In a first stage, SVMs SVM_1...SVM_4 are trained in leaving one sub-fold out of the training set, and are evaluated on the left-out sub-fold. In a second stage, a CRF model is trained, taking the sub-fold SVMs scores as inputs. We then use all the training videos to train the final SVM model (SVM_{test}). The final model outputs scores on the test fold, which are then refined by the CRF model. Note that each SVM training includes calibration using cross validation.

again the result to obtain the final shot descriptor.

Overall, each 2-hour movie is processed in 6 hours on a 16-core machine.

4.4.2 Shot classification with SVMs

We now detail the time-blind detection method, that scores each shot independently without leveraging temporal structure.

Per-channel training of SVMs. The 5 descriptor channels are input separately to the SVM training. For each channel and for each beat-category, we use all shots annotated as non-difficult as positive examples and all other shots (excluding difficult ones) as negatives to train a shot classifier. We use a linear SVM and cross-validate the C parameter, independently for each channel. We compute one classifier SVM_{test} per fold, and 4 additional classifiers SVM_1...SVM_4 corresponding to sub-folds, see Figure 4.5. Classifier outputs are transformed with a sigmoid to produce probabilities for fusion at a later stage.

Late fusion of per-channel scores The per-channel probabilities are combined linearly into a shot score. For one fold, the linear combination coefficients are estimated using the sub-fold scores. We use a random search over the 5D space of coefficients, one dimension per channel, to find the one that maximizes the average precision over the sub-folds. This optimization is performed jointly over all classes (shared weights), which was found to be better to reduce the variability of the weights.

4.4.3 Leveraging temporal structure

We leverage the temporal structure to improve the performance of the time-blind detection/localization method, using a conditional random field (CRF) [Lafferty et al., 2001]. We consider a CRF that takes the SVM scores as inputs. The CRF relies on a linear chain model. Unary potentials correspond to votes for the shot labels, while binary potentials model the probability of the sequences.

We model a video with a linear chain CRF. It consists of latent nodes $y_i \in \mathcal{Y}, i = 1, \ldots, n$ that correspond to shot labels. Similar to HMM, each node y_i has a corresponding input data point $x_i \in \mathbb{R}^d$. Variables x_i are always observed, whereas y_i are known only for training data. An input data point $x_i \in \mathbb{R}^d$ corresponds to the shot descriptor, In our case, the descriptor is the 11-D vector of L2-normalized SVM scores for each beat-category. The goal is to infer probabilities of shot labels for the test video.

The CRF model for one video is defined as:

$$\log p(Y|X; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \boldsymbol{\lambda}^T \boldsymbol{f}(y_i, X) + \sum_{i=1}^{n-1} \boldsymbol{\mu}^T \boldsymbol{g}(y_i, y_{i+1}, X),$$

where the inputs are $X = \{x_1, \ldots, x_n\}$ and the outputs $Y = \{y_1, \ldots, y_n\}$. We use the following feature (in the CRF literature sense) functions $\boldsymbol{f}$ and $\boldsymbol{g}$:

$$f_k(y_i, X) = p(y_i = k|x_i)\delta(y_i, k)$$
$$g_{k',k''}(y_i, y_{i+1}, X) = \delta(y_i, k')\delta(y_{i+1}, k'')$$

where $\delta(x, y)$ is 1 when $x = y$ and 0 otherwise. Therefore, the log-likelihood becomes

$$\log p(Y|X; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{k \in \mathcal{Y}} \lambda_k \sum_{i=1}^{n} p(y_i = k \mid x_i)\delta(y_i, k) + \sum_{\substack{k',k'' \in \mathcal{Y} \\ (k',k'') \neq (c,c)}} \mu_{k',k''} \sum_{i=1}^{n-1} \delta(y_i, k')\delta(y_{i+1}, k'')$$

We estimate $p(y_i = k \mid x_i)$ from the SVM classifier trained using cross validation on the training data. The CRF is learned by minimizing the negative log-likelihood in order to estimate $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. In practice we rely on the LBFGS method, implemented in the UGM toolbox [Schmidt].

At test time, the CRF inference outputs marginal conditional probabilities $p(y_i|X), i = 1, \ldots, n$. The inference relies on the forward-backward algorithm [Schmidt].

4.4.4 Beat-event localization

The final step consists in localizing instances of a beat-event in a movie, given confidence scores output by the CRF. To that aim, shots must be grouped into segments, and a score must be assigned to the segments. We create segments by joining consecutive shots for which CRF confidence is above 30% of its maximum over the movie. The segment's score is the average of these shot confidences.

Note that the CRF produces smoother scores over time for events that occur at a slower rhythm, see Figure 4.9. For example "good argue good" lasts usually longer than "joy bad", because the villain is delighted for a short time only. The CRF smoothing modulates the length of estimated segments: smoother curves produce longer segments, as expected.

	pursuit	battle	romance	victory good	victory bad	battle preparation	despair good	joy bad	good argue bad	good argue good	bad argue bad	mean accuracy
Leave 4 movies out												
SIFT	53.8	76.4	23.9	11.7	4.4	22.1	15.0	9.5	15.1	25.5	4.0	23.76 ± *5.26*
CNN	**66.4**	60.0	16.6	6.0	2.4	9.4	21.7	6.6	17.7	30.2	4.7	21.96 ± *5.91*
dense trajectories	58.5	**85.2**	**38.0**	12.7	6.2	**28.0**	19.5	11.6	**18.8**	**40.4**	1.8	29.15 ± *6.12*
MFCC	28.1	56.3	4.5	**17.7**	**36.2**	3.8	**35.4**	**15.6**	17.3	26.5	0.0	21.95 ± *13.97*
Face descriptors	47.9	58.1	8.6	12.7	11.4	17.3	9.3	3.2	6.2	22.3	**4.7**	18.35 ± *10.50*
linear score combination	63.9	89.2	32.3	14.0	11.4	18.6	26.0	12.1	18.0	44.3	1.8	**30.15** ± *6.72*
+ CRF	76.0	91.2	57.6	19.9	1.0	41.4	43.1	9.6	25.1	44.8	0.0	**37.25** ± *9.94*
Leave 1 franchise out												
linear score combination	57.8	83.6	13.0	14.9	9.6	3.8	28.0	5.2	18.2	44.3	0.0	**25.32** ± *7.40*
+ CRF	75.4	87.4	31.3	15.8	0.0	12.7	33.4	5.7	23.2	43.7	0.0	**29.89** ± *12.11*

TABLE 4.3: Performance comparison (accuracy) for shot classification. Standard deviations are computed over folds.

	Leave 4 movies out											
CRF + thresholding	34.6	38.9	22.6	14.6	4.4	26.7	6.4	4.6	12.2	16.9	0.6	16.59 ± *6.82*
	Leave 1 franchise out											
CRF + thresholding	36.8	36.5	28.9	14.3	4.5	1.7	4.2	5.2	6.5	13.5	3.7	14.16 ± *6.84*

TABLE 4.4: Performance comparison (average precision) for beat-event localization.

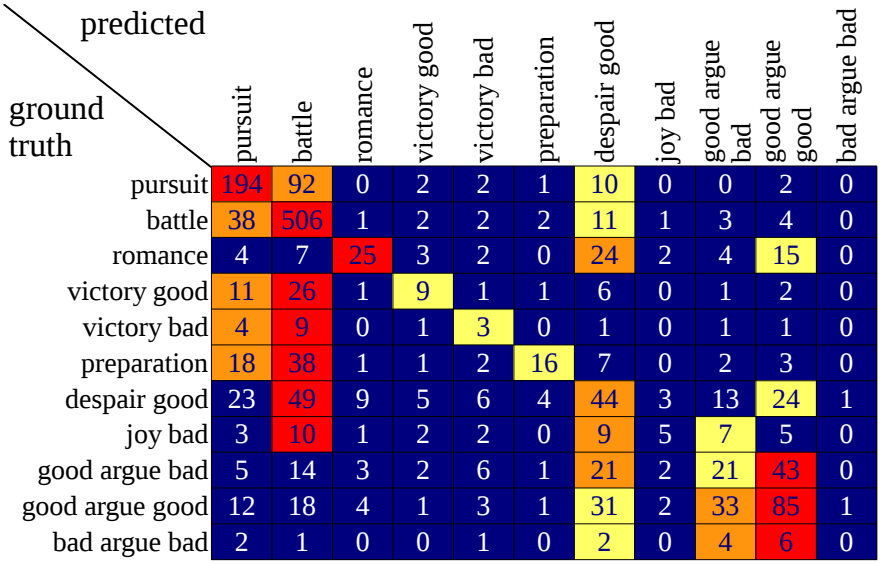


FIGURE 4.6: Confusion matrix for shot classification with SVM and linear score combination for the "leave 4 movies out" setting.

4.5 Experiments

After validating the processing chain on a standard dataset, we report classification and localization performance.

4.5.1 Validation of the classification method

To make sure that our descriptors and classification chain is reliable, we run it on the small Coffee & Cigarettes [Laptev and Pérez, 2007] dataset, and compare the results to the state-of-the-art method of Oneata et al. [Oneata et al., 2013]. For this experiment, we score fixed-size segments and use their non-maximum suppression method NMS-RS-0. We obtain 65.5 % mAP for the "drinking" action and 45.4 % mAP for "smoking", which is close to their performance (63.9 % and 50.5 % respectively).

4.5.2 Shot classification

Table 4.3 shows the classification performance at the shot-level on the two types of splits. The low-level descriptors that are most useful in this context are the dense trajectories

FIGURE 4.7: Sample faces corresponding to shots for which the face classifier (*i.e.* SVM trained on faces) scored much higher than the SIFT classifier (*i.e.* trained on full images). Similar facial expressions can be observed within each beat-category, which suggests that our face classifier learns to recognize human expressions to some extent.

descriptors. Compared to setups like TRECVID MED or Thumos [Over et al., 2014, Oneata et al., 2014], the relative performance of audio descriptors (MFCC) is high, overall the same as for *e.g.* CNN. This is because Hollywood action movies have well controlled soundtracks that almost continuously plays music: the rhythm and tone of the music indicates the theme of the action occurring on screen. Therefore, the MFCC audio descriptors convey high-level information that is relatively easy to detect automatically.

The face descriptor can be seen as a variant of SIFT, restricted to facial regions. The face channel classifier outperforms SIFT in three categories. Upon inspection, we noticed however that only a fraction of shots actually contain exploitable faces (*e.g.* frontal, non-blurred, unoccluded and large enough), which may explain the lower performance for other categories. The performance of the face channel classifier may be attributed to a rudimentary facial expression recognition property: the faces of heroes arguing with other good characters can be distinguished from the grin of the villain in *joy bad*; see Figure 4.7.

The 4 least ambiguous beat-categories (*pursuit*, *battle*, *battle preparation* and *romance*) are detected most reliably. They account for more than half of the annotated shots. The other categories are typically interactions between people, which are defined by identity and speech rather than motion or music. The confusion matrix in Figure 4.6 shows that verbal interactions like "good argue good" and "good argue bad" are often confused.

The "leave-4-movies out" setting obtains significantly better results than "Leave-1-franchise out", meaning that having seen movies from a franchise makes it easier to recognize what is happening in a new movie of the franchise: Rambo does not fight in the same way as Rocky. Finally, the CRF allows to leverage temporal structure using the temporally dense annotations, improving the classification performance by 7 points.

Qualitative results Figure 4.8 shows a few classification examples. The first line is from *Indiana Jones 1*. It is a *pursuit* between Indiana and the villain, and at some point the hero jumps on the bad guy's car, so it becomes a *battle*. Since there are still moving vehicles, the classifier cannot really distinguish the two stages.

In the second example, from *Lethal Weapon 2*, the hero meets his "love interest" in a supermarket. The discussion is energetic and there are many close-ups on the character's faces, which explains why the classifier recognizes *despair good*. Afterwards, the two meet on a beach and the romance starts over (detected properly this time).

In the third line is an extract of *Rocky 3*. This franchise is arguably the one with the most franchise-consistent, and therefore predictable structure. The hero and his love interest are quarreling about him boxing again. After that begins the *battle preparation*, which the classifier and CRF is able to distinguish from the actual *battle*.

The fourth example is from the end of *Rambo 1*, where the hero surrenders after a tense discussion with his mentor, breaking into tears. This is a very unusual ending for an action movie (indeed, it is the only final *victory bad* in our collection). The classifier gives a reasonable result, except that it confuses *good argue good* and *despair good*.

4.5.3 Beat-event localization

Table 4.4 gives results for beat-event localization. We observe that the performance is low for the least frequent actions. Indeed, for 8 out of 11 categories, the performance is below 15% AP. Figure 4.9 displays localization results for different beat-categories. Categories, such as battle and pursuit, are localized reliably. Semantic categories, such as romance, victory good and good argue good are harder to detect.

More advanced low-level features could improve the results on those events. Indeed, recognition of characters, their pose and speech appear necessary.

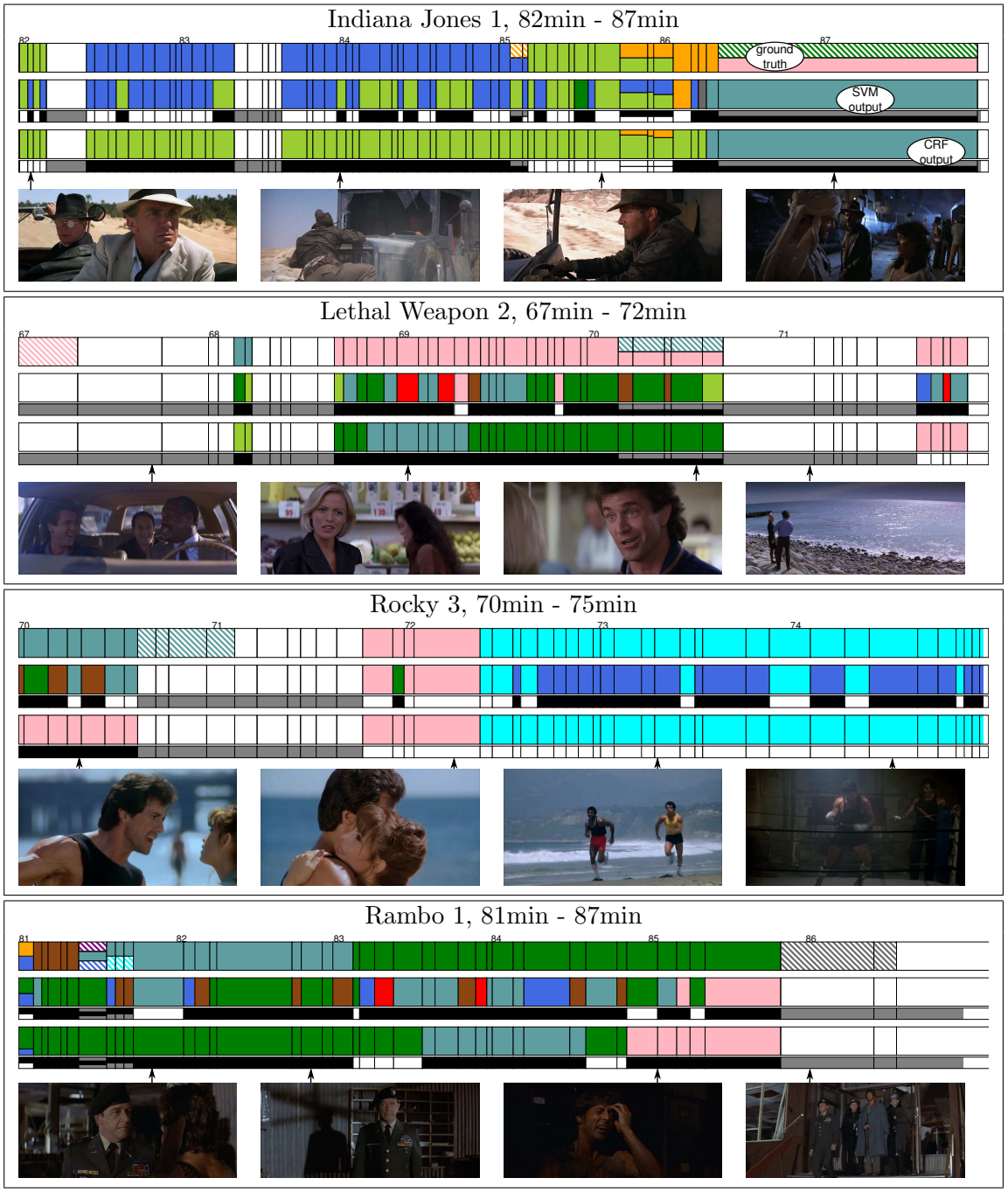


FIGURE 4.8: Classification examples on a few movie extracts, showing each stage of classification and their respective performances, and some example frames. The color codes are the same as in figure 4.3, with hashes = difficult. For each shot, we draw only the shortlist of classes that are taken into account in the scoring (*i.e.* the number of non-difficult ground-truth labels for the shot). The color code for the classifier evaluation is: white = true positive, gray = ignored, black = false positive.

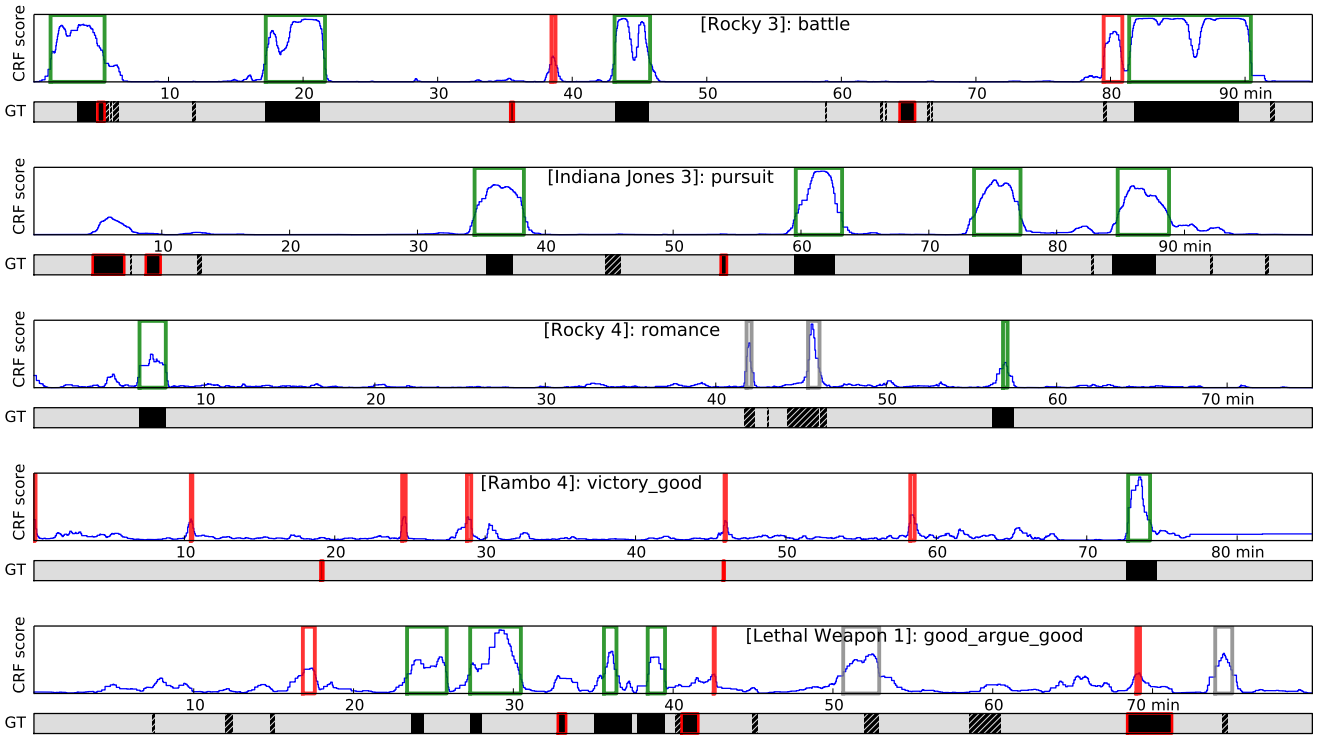


FIGURE 4.9: Example of localization results, for several beat-categories and movies. For each plot, detected beat-events are indicated with bold rectangles (green/gray/red indicate correct/ignored/wrong detections). Ground-truth (GT) annotations are indicated below (beat-events marked as difficult appear hatched), and likewise missed detections are highlighted in red. Most often, occurrences of the beat-events are rather straightforward to localize given the CRF scores.

4.5.4 Domain adaptation

We consider the leave-4-movies-out splitting policy and the classification setting. For a test movie, the three other movies from the same franchise appear in the training set. The training shots from these movies are more relevant to build a classifier for the test movie. For example, a *victory* scene at the end of a boxing match in *Rocky 4* is likely to be more similar to other boxing matches in *Rocky 1* to *Rocky 3* than *victory* scenes in *Indiana Jones*. However, we are in a training regime where the training data is scarce, so we cannot afford to just drop the training examples from other sources.

Here, we explore the setting when the franchise (= domain) of the test movie is known. We call this setting *domain adaptation*. The purpose of these experiments is to investigate the impact of using the semantic consistency within-franchise at training and test time.

Problem formulation. Consider a particular movie franchise. At training time, assume that we have a partitioned set of training examples

$$(x_1, y_1), \ldots, (x_\ell, y_\ell), (x_{\ell+1}, y_{\ell+1}), \ldots, (x_{\ell+m}, y_{\ell+m}), \tag{4.1}$$

partitioned into two sets as follows:

$$\begin{aligned} &\text{within-franchise examples } (x_1, y_1), \ldots, (x_\ell, y_\ell) \\ &\text{other-franchise examples } (x_{\ell+1}, y_{\ell+1}), \ldots, (x_{\ell+m}, y_{\ell+m}) \end{aligned}$$

Sample weights. We would like to train domain-specific classifiers for each movie by putting a larger weight on training examples from the movie franchise, and smaller weight to examples from other movie franchises.

Let Δ be the number of domains in the test set. We treat each category independently, so we can assume there is a binary classification problem. For each domain $\delta \in 1 \ldots \Delta$, we train a separate SVM classifier f_δ^ρ, using the weight ρ for the samples of domain δ.

We define the domain-adapted scoring function as:

$$\widetilde{f}^\rho(x_j) = f_{\delta_j}^\rho(x_j), \tag{4.2}$$

where (x_j, δ_j) is a data example and its domain.

We consider the following domain-adapted SVM:

$$\min_w \|w\|_2^2 + C \left(\frac{\rho}{\ell} \sum_{i=1}^{\ell} L(w^T u_i, y_i) + \frac{1}{m} \sum_{j=\ell+1}^{\ell+m} L(w^T u_j, y_j) \right)$$

where $L(\cdot, \cdot)$ is the usual linear hinge loss.

The hyper-parameters, C the regularization parameter, and ρ the domain-adaptation parameter, are tuned using cross-validation. First, the C parameter is tuned w.r.t the weighted AP metric.

The domain weight ρ is selected using the Average Precision on the training set; the cross-validation scores are aggregated using $\widetilde{f}^\rho$. Again, we select a separate weight for each category.

Averaging scores of domain-specific classifiers. As noted in Tommasi et al. [2013], merging examples from biased datasets is usually not the best way for unsupervised domain adaptation. It is much better to average the scores of classifiers, learned on each dataset separately.

In our setting, we use a weighted sum of scores, with more weight for the target domain. Table 4.5 shows results in the final setting.

	pursuit	battle	romance	victory_good	victory_bad	preparation	despair_good	joy_bad	good_argue_bad	good_argue_good	bad_argue_bad	mAP	accuracy
SIFT+SVM	30.3	48.6	15.1	6.4	0.9	20.6	5.5	2.2	6.6	7.7	0.7	13.1 ± *2.5*	23.8 ± *5.3*
Late-DA	29.0	48.4	13.4	7.9	1.0	31.8	6.2	3.6	8.8	7.1	1.2	14.4 ± *4.0*	26.5 ± *11.6*
CNN+SVM	28.9	37.8	12.5	3.9	1.9	7.0	6.4	3.2	6.1	10.9	4.3	11.2 ± *3.6*	22.0 ± *5.9*
Late-DA	28.2	41.5	8.7	3.5	0.8	16.4	6.0	2.8	9.4	9.1	3.3	11.8 ± *3.8*	24.4 ± *10.4*
DT+SVM	39.1	63.7	22.8	5.5	2.2	32.8	5.9	4.5	7.5	14.5	1.2	18.2 ± *4.3*	29.2 ± *6.1*
Late-DA	41.3	65.4	18.3	7.3	4.3	47.0	6.5	2.9	13.5	17.9	1.3	20.5 ± *5.4*	31.9 ± *8.6*
MFCC+SVM	27.7	54.5	10.1	4.6	1.5	10.6	8.8	2.0	7.5	22.9	4.4	14.1 ± *3.4*	21.9 ± *14.0*
Late-DA	31.4	53.5	11.4	5.4	1.7	18.1	8.0	1.9	8.9	23.8	6.1	15.5 ± *4.8*	28.1 ± *15.6*
(1) SVM + LF	43.9	66.4	26.0	8.3	2.2	37.2	7.9	5.0	10.1	21.3	10.0	21.6 ± *5.8*	30.0 ± *6.9*
(2) $\widetilde{f}^{\rho^*}$ + LF	44.9	66.8	25.8	6.6	4.9	40.7	7.1	7.0	9.3	19.2	8.6	21.9 ± *5.8*	32.0 ± *8.2*
(3) Late-DA + LF	47.4	68.8	22.2	9.8	3.9	50.6	9.1	5.7	15.0	22.0	5.5	23.7 ± *7.0*	**35.3** ± *11.8*
(1) + CRF	62.1	72.1	35.5	23.0	3.7	50.0	7.6	7.2	12.9	18.2	0.6	26.6 ± *7.4*	36.3 ± *10.5*
(2) + CRF	62.3	72.4	34.7	16.2	4.1	50.2	7.7	7.1	10.6	21.0	1.3	26.2 ± *7.1*	38.3 ± *13.3*
(3) + CRF	59.3	71.5	26.2	24.5	12.1	53.5	9.6	8.6	14.4	18.6	9.0	27.9 ± *11.2*	**39.7** ± *17.3*

*

TABLE 4.5: Domain adaptation. Target weight is selected by cross-validation on the training set. Leave 4 movies out. Per-class values correspond to AP.
Legend: ***+SVM** — standard SVM training setup with examples from all domains, **Late-DA** — weighted average of domain-specific scores, **LF** — late fusion.

4.6 Conclusion

Despite the explosion of user-generated video content, people are still watching professionally produced videos most of the time. Therefore, the analysis of this kind of footage will remain an important task. In this context, Action Movie Franchises appears as a challenging benchmark. The annotated classes range from reasonably easy to recognize (*battle*) to very difficult and semantic (*good argue bad*). We also provide baseline results from a method that builds on state-of-the-art descriptors and classifiers.

Therefore, we expect it to be a valuable test case in the coming years. We make the complete annotations and the evaluation scrips publicly available.

Chapter 5

Conclusion

In this manuscript, we have explored several video analysis tasks — summarization, classification, localization, — reviewed existing approaches for these tasks and presented our contributions for these tasks. In this section, we summarize our contributions and discuss possible future directions in the above mentioned areas.

5.1 Summary of contributions

Category-specific video summarization. We proposed a category-specific video summarization approach that relies on a weakly supervised set of videos to learn the importance scoring function. At test time it performs a temporal segmentation of the video and then builds a summary with the most important segments. We introduced a new dataset, called MED-Summaries, containing the annotation of temporal segments in videos and the grades of relevance to one of 10 categories. Experimental evaluation on MED-Summaries showed that the proposed approach constructs video summaries with higher overall importance.

Event Detection in Action Movie Franchises. We introduced a novel Action Movie Franchises dataset for evaluation of beat-event classification and localization in action movie franchises. The dataset contains 20 action movies of 5 franchises with a dense annotation of 11 non-exclusive beat-categories on both shot and event levels. We defined evaluation protocols for classification and localization tasks and two different experimental settings to investigate the impact of intra-franchise information. We proposed an approach for classification of video shots into beat-categories based on a state-of-the-art pipeline for multimodal feature extraction, classification and fusion. The proposed approach for localizing beat-events uses a temporal structure inferred by

a conditional random field (CRF) model learned in a cross-validation way from training data.

Contributions to the TRECVID Multimedia Event Detection submissions. We presented an overview of the Inria TRECVID Multimedia Event Detection system, which solves the task of large scale multimedia event detection in real-world videos. We summarized the contributions to this system made within the work on this thesis. In particular, we presented state-of-the-art image descriptors, based on SIFT, and audio descriptors, based two low-level features: MFCC and ScatNet.

5.2 Future directions

Leveraging context for video summarization. An important aspect of video summarization is modeling the content of the video in relation to context: the subject of the video, the summarization goal, the knowledge of the target audience. Structuring these aspects can improve the quality of the summaries. The structure of the context could be learned through direct supervision, or by exploiting side information gathered from the web, like it is done in recent approaches [Khosla et al., 2013, Kim et al., 2014, Song et al., 2015].

Using more modalities for event recognition Using data from multiple modalities can help better understanding the video. It would be interesting to leverage parallel channels of descriptors, such as visual information, audio information and text subtitles, to construct highly-informative summaries. Furthermore, the events we considered are often human-centric, as in the MED-Summaries and the Action Movie Franchises datasets. Using human-centered dynamic features, for instance using part-based detectors and tracking algorithms [Hua et al., 2014, Gkioxari and Malik, 2015], is likely to lead to improvement for recognition purpose, and would also yield spatial localisation information.

Speed and scalability Current video analysis systems are torn between i) fast approaches that use global bag-of-features type representations, on top of descriptors that can be computed almost real-time; ii) slower approaches that use finer-level representations (temporal, spatial), as in spatial localization for instance, which require a lot of time to be computed. An important problem is to develop faster approaches that capture finer-level information and cleverly combine them with more global and almost

real-time approaches to build high-performance video analysis systems. The TRECVID competitions serie offer an ideal “experimentation field” to design such systems.

Appendix A

TRECVID contributions

Abstract

The Inria TRECVID MED system was developed from 2011 to 2014 for the TRECVID Multimedia Event Detection competitions. Being a member of the corresponding teams, the author of this manuscript deems it necessary to first outline the whole system, and then describe his contributions in more detail in Section A.2.

Publications

M. Douze et al. *The INRIA-LIM-VocR and AXES submissions to Trecvid 2014 Multimedia Event Detection*. TRECVID workshop, Gaithersburg, 2014

R. Aly et al. *The AXES submissions at TrecVid 2013*, TRECVID workshop, Gaithersburg, 2013.

D. Oneata et al. *AXES at TRECVid 2012: KIS, INS, and MED*, TRECVID workshop, Gaithersburg, 2012.

M. Ayari et al. *INRIA@TRECVID'2011: Copy Detection & Multimedia Event Detection*, TRECVID workshop, Gaithersburg, 2011.

A.1 Inria TRECVID MED system

In this section, we describe the 2014 version of the system. The system has been progressively improved since 2011.

A.1.1 Features

The system is based on 3 types of features:

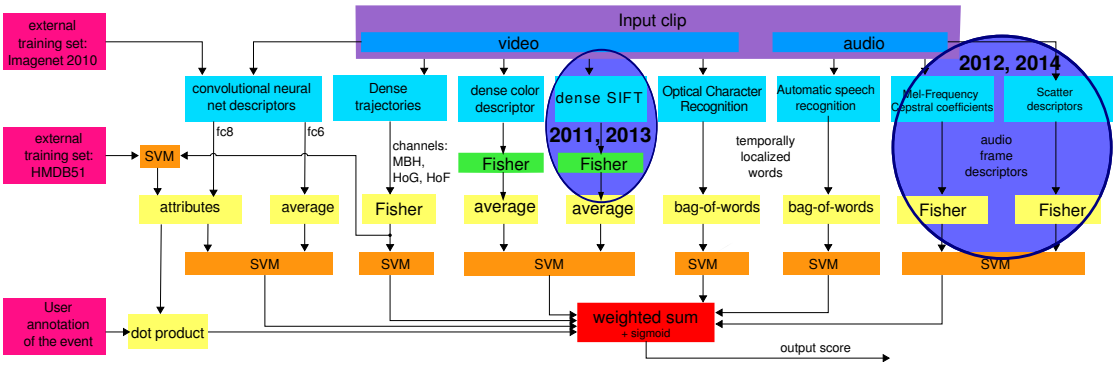


FIGURE A.1: Overview of the whole INRIA-LIM-VocR and AXES system for TRECVID MED 2014 [Douze et al., 2014]. Circles denote the contributions through years 2011-2014, made within the work on this dissertation.

Descriptor	Dimension	Real-time factor
Dense trajectories	434,176	7.1
SIFT	276,479	3.1
Color	72,703	2.6
MFCC	80,895	0.04
Scatter transform	65,663	0.18
CNN	4,096	0.33
Attributes	2,102	(7.43)
OCR	110,000 (sparse)	1.5
ASR	110,000 (sparse)	1.1–2

TABLE A.1: Descriptor dimension and processing time as a factor w.r.t. video duration on one CPU core. The real-time factor in parentheses (for the descriptor Attributes) is derived from other features at a negligible additional cost. Source: [Douze et al., 2014].

1. Local visual and audio descriptors, which are aggregated to global descriptors, one for each type of low-level descriptor, using Fisher Vectors.

2. Mid-level attribute features based on object and action detectors trained on external datasets.

3. Additional high-level features extracted using ASR and OCR features.

Table A.1 lists the features, their dimensions and the computational cost.

For each type of low-level feature, we aggregate the local descriptors into a global signature by means of a Fisher vector (FV) [Sanchez et al., 2013]. The number of Gaussians chosen for the FVs are a trade-off between the accuracy of the representation and computational constraints. Visual frame-based FVs are averaged together to produce a signature for the complete video. The complete descriptions of low-level features can be found in [Douze et al., 2014].

In order to cope with the restricted positive training data, we implemented mid-level representations. These representations rely on detectors trained for a set of object and

action classes that are not directly related to the MED events. The classes are chosen such that they are basic, and a sufficient amount of training data is available to train classifiers for them. For example, the action "stand up" is more basic than the event "townhall meeting". This is inspired by similar representations used for attribute-based and zero-shot image classification [Akata et al., 2013]. The mid-level feature vector of a video clip is built from the confidence scores of the clip for each of the chosen classes. In the case of the CNN features described below, we do not directly use the detection confidences, but rather an internal representation that is used by the convolutional network to detect object classes. The three mid-level representations are detailed in [Douze et al., 2014].

The high-level features temporally localize words in the video. They come from on-screen text transcribed by optical character recognition (OCR) and from speech recognition (ASR). The transcripts are aggregated to sparse feature vectors using a bag-of-words representation based on a 110k-word dictionary consisting mainly of English words. More details in [Douze et al., 2014].

A.1.2 Classification setup

Each of the feature vectors (low-, mid-, or high-level) is used to train a linear SVM classifier with the LIBSVM software package [Chang and Lin, 2011]. To determine the hyper-parameters of the SVM we used different strategies, depending on the number of training examples. In 2014 we used the same classification approach as in 2013 [Aly et al., 2013]: 10-fold cross-validation to estimate the SVM's regularization parameter C and the weighting factor for the positive samples.

A.1.3 TRECVID MED datasets

The evaluation dataset is updated every year in order to prevent overfitting of the submitted systems. The test set labels are not disclosed to the participants before the evaluation is done on the organisers' side.

The TRECVID 2011 MED dataset consists of videos of 10 event categories and an additional *NULL* category that contains videos of none of the 10 categories. For each category there are between 100 and 300 videos in the training set, while there are 9600 video for the *NULL* category. In the test set, there are $32,000$ videos, which have a total duration of $1,000$ hours.

Although the test set of the MED dataset has been growing through years, there is an interest in learning models from little training data. In 2013 and 2014 this corresponds

combined channels	EF	LF with rest
LF only		53.07
Color+SIFT	34.07	53.09
SIFT+trajectory	46.22	52.68
MFCC+ScatNet	18.66	53.05
ASR+OCR	18.57	53.22
CNN+Attributes	39.15	54.17
MFCC+ScatNet, CNN+Attributes		**54.27**
MFCC+ScatNet, CNN+Attributes, Color+SIFT		54.02
MFCC+ScatNet, ASR+OCR		53.22
all possible EFs		53.95
AXES'13 submitted combination		52.58

TABLE A.2: Early and late fusion (EF and LF respectively). Results for the official MED 2011 test set. The "EF" columns report results combining just the 2 features with early fusion (e.g., Color+SIFT: 34.07). The columns "LF with rest" are obtained with a LF on all the channels, where some of them are combined with EF (e.g., EF of Color+SIFT and LF of the other channels: 53.09). See Section A.1.4 for more details.

to a setting *10Ex*, where only 10 training examples are available, which was the main setting in 2014. For more details on 10Ex see [Douze et al., 2014].

In 2011 and 2012 the official performance measure was the Normalized Detection Cost (NDC) [Over et al., 2011], which is defined as a weighted linear combination the False Alarm rate and the Missed Detection rate, with a much higher cost for False Alarms. In 2013 and 2014 the official performance measure was the Average Precision (AP) per category and the Mean Average Precision (mAP) for all categories.

A.1.4 Early fusion and late fusion

We employ two kinds of techniques to combine individual features: early and late fusion. In early fusion, the combined feature vector is a concatenation of the (scaled) individual vectors. The SVM classifier is then trained on this concatenated feature vector. In the case of late fusion, we linearly combine the scores of the SVM classifiers trained on the individual features (with appropriate weights).

Table A.2 presents results obtained with fusion of various channels. The first row of the table gives the results when using late fusion on all channels without doing any early fusion. The next five rows show results of performing early fusion with two channels. Here, we first report the performance of using only this combination of two channels, and then the result when this early fusion combination is late fused with all the remaining channels. Although early fusion generally improves over the individual channel outputs, the early fusion generally does not improve performance significantly when combining it with late fusion of the other channels. One exception to this trend is the

CNN+Attributes combination in the 100Ex case. The following four rows in the table show similar results when using early fusion to combine different pairs of features. All the results in this part of the table improve over the late fusion baseline.

The late fusion weights are estimated with 30-fold cross-validation. The best setting is marked in bold in the table and corresponds to the submitted version.

A.1.5 Final results

Table A.3 shows the evolution of our results through years 2011–2013.

NDC per system	birthday party	changing a vehicle tire	flash mob gathering	unstuck a vehicle	grooming an animal	making a sandwich	parade	parkour	repairing an appliance	sewing project	mean
Best TV MED 2011 result *	44.4	43.9	26.3	38.0	62.2	56.2	44.6	30.8	33.1	57.5	43.7
Inria system 2011	71.7	73.2	43.3	56.5	80.4	85.7	55.1	44.9	50.9	80.3	64.2
Inria system 2012	45.9	45.1	25.8	38.4	53.9	55.1	39.1	22.7	34.5	50.7	41.1
Inria system 2013	43.0	39.2	25.6	34.5	48.4	52.6	33.1	21.6	32.1	49.1	37.9

* [Natarajan et al., 2012b]

TABLE A.3: Evolution of the Inria event classification system through 2011–2013. Performance is reported in the NDC error (lower is better).

AP per system	birthday party	changing a vehicle tire	flash mob gathering	unstuck a vehicle	grooming an animal	making a sandwich	parade	parkour	repairing an appliance	sewing project	mean
Inria system 2012	43.7	34.3	64.7	40.2	25.7	37.6	53.8	62.3	50.3	31.9	44.5
Inria system 2013	48.9	54.7	69.0	48.5	38.8	35.8	60.3	71.9	57.2	40.6	52.6
- trajectories + SIFT	33.5	52.5	64.4	48.4	36.8	25.0	51.2	72.6	43.0	36.6	46.4
- colour	20.0	29.9	48.8	26.0	15.6	17.3	31.0	34.4	38.4	15.8	27.7
- audio	33.3	5.9	21.3	12.9	4.7	11.2	23.2	7.2	43.3	18.4	18.2
- ASR	3.55	16.7	0.4	3.5	0.5	6.7	0.8	0.3	39.2	10.4	8.2
- OCR	10.1	10.0	10.7	1.2	6.3	19.4	9.9	0.9	32.1	7.8	10.8
Inria system 2014	50.9	59.0	65.7	52.0	35.7	45.0	57.3	69.9	61.7	45.7	54.3

TABLE A.4: Evolution of the Inria event classification system through 2011–2013. Performance is reported in terms of the Average Precision (higher is better).

A.2 Contributions of the author to the submissions

A.2.1 2011. SVM classification setup. SIFT Fisher Vectors per frame.

In 2011, I implemented the classification pipeline for the SIFT channel. In this pipeline frame descriptors are used as training examples. The frame labels are inherited from video labels [Ayari et al., 2011].

We extract image features from every 10th frame. For each image SIFT descriptors [Lowe, 2004, Tuytelaars and Mikolajczyk, 2007] are extracted on a dense grid at 5 scales with horizontal and vertical steps of 4 pixels. The dimension of the descriptors is reduced using PCA from 128 to 64 dimensions. The descriptors of an image are, then, aggregated into a Fisher Vector [Perronnin and Dance, 2007]. Here, we use a Fisher Vector based on a Gaussian mixture model [Bishop, 2009, Hastie et al., 2009] with 64 Gaussians, which was a trade-off between computational efficiency and classification performance.

A linear one versus all SVM classifier [Duda et al., 2012] is, then, trained on the Fisher Vectors. We use a subset of 1000 positive and 5000 negative frames for training each event classifier. The positive frames are obtained from approx. 100 videos and the negatives from 5000 videos. The C parameter is selected using 5-fold cross-validation (separately for each event category). We ensure that frames from a video are in the same fold.

To assign a label to a video clip, we score every 10th frame for a given event and, then, use the maximum frame scores as a confidence value for a video clip and event class.

A.2.2 2012 and 2014. MFCC channel.

In 2012, I took part in the implementation of the MFCC channel. In 2014, I was again responsible for the MFCC pipeline. In this pipeline, descriptors for whole videos are used as training examples.

We down-sample the original audio track to 16 kHz with 16 bit resolution and then compute Mel-frequency cepstral coefficients (MFCC) [Rabiner and Schafer, 2007] with a window size of 25 ms and a step-size of 10 ms, keeping the first 12 coefficients of the final cosine transformation and the energy of the signal. We enhance the MFCCs with their first and second order derivatives. The MFCC features are then aggregated into a FV with a vocabulary size of 256.

A.2.3 2013. SIFT channel.

In 2013, I was responsible for running experiments for the SIFT channel pipeline. In this pipeline, descriptors for whole videos are used as training examples.

The frame descriptors are essentially the same as in 2011, except:

- We aggregate frame Fisher Vectors into a single descriptor per video (like in 2012).
- We extract image features for every 60th frame.
- We use K=256 Gaussians.
- We use Spatial Fisher Vectors [Krapac et al., 2011b] to encode the spatial information.

The Fisher Vectors are normalized per frame, then average pooled along the video, and finally normalized again.

A.2.4 2014. ScatNet.

In 2014, in addition to MFCC channel, I implemented the ScatNet channel pipeline.

In this pipeline, we resample the audio track to 44.1 kHz and then compute scattering coefficients [Andén and Mallat, 2011] with a window size of 500 ms and a step size of 185 ms. The ScatNet transform is based on several layers of a modified wavelet decomposition. It is designed to capture longer-range audio structures as compared to the standard MFCC descriptor. We used the ScatNet toolbox [Andén and Mallat, 2013b]. We used first and second order coefficients, with quality factors $Q_1 = 8$ and $Q_2 = 1$, which results in 526 dimensions. The ScatNet features are then aggregated into a FV with a vocabulary size of 128.

Table A.5 shows an intermediate experiment with different window sizes and different number of principal components. We found that using shorter time windows is beneficial.

		PCA dim.			
T, sec.	Orig. dim.	32	64	128	256
0.2	435	11.14	11.84	12.85	13.07
0.5	526	11.66	11.55	11.29	12.92
1.5	718	10.06	10.27	10.47	10.31

TABLE A.5: Performance of the ScatNet descriptor for different sizes of temporal window and number of PCA components

Appendix B

Action Movie Franchises: event definitions.

B.1 Pursuit

Short definition: villains are following heroes, or the opposite (it usually takes some time)

Synonyms: chasing, following; difficult: running, approaching, escaping, crawling

Full definition: During pursuit, one of the parties ("good" or "bad") is following the other, either on foot, or in a vehicle (car, helicopter, etc.). Both the persecutor and the persecuted are aware of the pursuit: the former is trying to catch up and the latter to escape. There is a nonzero distance between parties, so that they mostly interact by shooting. There can be more than 1 character in each of the parties.

Except rare cases, pursuit is fast and dynamic. Another distinct attribute of the pursuit is the tense state of both parties.

Special cases:

- During a car pursuit scene, it often happens, that characters fight in a moving car. We count it as a *battle*, but not as a *pursuit*.
- Running from danger should not be annotated as *pursuit*.

Sample snapshots from *pursuit* events

B.2 Battle

Short definition: active confrontation between heroes and villains

Synonyms: fighting, shooting; difficult: explosion, single shots, torturing

Full definition: During a battle, good and bad characters try to hurt each other. It can be a hand-to-hand fight or an armed conflict. One or both parties can be inside a vehicle such as car, plane, helicopter.

A battle consists of several attacks by each party. Usually when one party attacks, the other one tries to defend: hide, block or escape from being hurt. Sometimes they attack simultaneously.

Special cases:

- In Rocky, there is one main battle per movie + several minor battles; each round is a separate event.
- In Rocky, a boxing battle during training is annotated as both *battle* and difficult *preparation*.
- On the contrary, shooting practice is considered as *preparation* and difficult *battle*.
- Threatening with arms should not be considered as *battle*.

Sample snapshots from *battle* events

B.3 Romance (good)

Short definition: expression of mutual feelings of two good characters

Synonyms: love, difficult: mutual attraction

Full definition: A romance happens between two characters of the same party, mostly between the main hero and his love interest. It is an expression of their mutual feelings to each other and usually implies hugging, kissing, smiling and also flirting. In most cases, the characters stand close to each other and there is an eye contact.

Romance episodes happen when heroes are being separated without their will, or when they rejoin each other after a long separation.

Special cases:

- In Rocky, dialogues between the hero and his love interest are often annotated as difficult *romance*

Sample snapshots from *romance* events

B.4 Victory (good / bad)

Short definition: good / bad characters win a battle or pursuit

Synonyms: winning, happy end, knockout; difficult: knock-down

Full definition: (for *victory good*, victory bad is the opposite) Victories happen in the end or right after a battle or a pursuit. However, not every battle nor pursuit will have a winner. If there is a temporary advantage during the battle, it is not considered as a full victory. The *victory good* event also happens when bad characters lose.

A victory usually implies positive emotions of the characters, although the winners are often exhausted.

Winning a battle means either destroying the major part of the enemy forces or capturing the enemy. Winning a pursuit means either catching the pursued or escaping from the persecutor.

Special cases:

- In Rocky, there is one victory in the end of the battle. Knock-downs are counted as difficult victories.

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *victory good* events

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *victory bad* events

B.5 Preparation

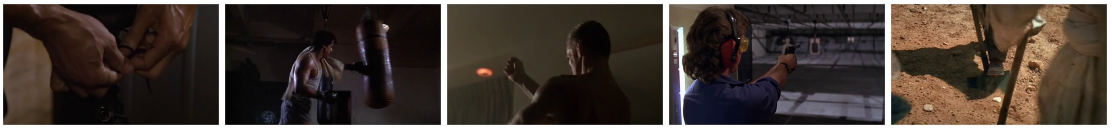
Short definition: preparing to the battle - training, jogging, setting up the armor, etc.

Synonyms: training, drill, jogging; difficult: setting up weapons/equipment, recharging gun, handwork (e.g. bomb installation)

Full definition: Preparation is aimed to increase the chances of winning in the expected battle. For that, characters either improve their physical forces (e.g. jogging, muscle-strengthening), practice required skills (e.g. shooting practice) or imitate the battle with partners.

Except for running, preparation usually takes place in a gym or a similar building. In many cases, characters of the same party prepare together. Often there is a coach that guides the preparation process.

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*



Sample snapshots from *preparation* events

B.6 Despair good

Short definition: desperate behaviour of the heroes, normally not during fight, but connected to the global battle

Synonyms: wail, cry, severe fatigue, exhaustion, "all is lost", depression, shock, fright; difficult: heavy breathing, sad mood

Full definition: Despair or desperation is a state in which all hope is lost or absent [c.f. Wordnet]. Our definition is broader. A despair event contains visual and aural signs of despair: crying, wailing, moaning, etc. In general, people in despair cannot normally communicate with other people. They are not listening others or not saying anything. A strong fright of a hero can be viewed as a short *despair good* event.

In most cases there are 1–3 heroes in despair. A special case - panic in public place - many people are scared and screaming.

Heroes can be suffering because of physical wounds or psychological stress. In both cases heroes express negative emotions. In rare cases, when a hero is seriously wounded, he/she may talk to other person to reduce hurt.

There is a special case when the global battle finishes and the heroes (esp. women) cannot believe in the happy end and start crying. This should not be considered as *despair good* event.

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *despair good* events

B.7 Joy bad

Short definition: villains show dominance or express joyful emotions

Synonyms: laugh, sarcasm, arrogance, exult, gloat over (misfortunes of others); difficult: transient grin, quick smile

Full definition: When villains succeed in their cruel plans, they start to celebrate it. It often happens before the global battle finishes. It usually appears as a close-up on villain's face. The particular expression of the villain varies in different movies. It can be a sarcastic laugh, or angry face, or arrogant look at the heroes, or even happy face.

Sometimes it happens that several villains laugh together. However, if bad characters laugh while joking with good characters, this should not be considered as a positive.

Special cases:

- Fighters in Rocky do not smile as much as villains in other movies. Therefore many of *joy bad* examples are annotated as difficult.

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *joy bad* events

B.8 Good/bad argue good/bad

Short definition: intense discussion with a strong disagreement

Synonyms: argument, debate, quarrel; difficult: objection (protest), argumented disobeidance

Full definition: Argument is an intense discussion with a strong disagreement. Not only each party expresses his/her opinion, but, more importantly, tries to object strongly to the opponent. In a typical tense argument parties raise their voices, may provoke a fight.

In a complete argument we hear both parties arguing. If one of the opponents is mild, tries to find a compromise and to calm down the other, this should not be considered a true argument. This often happens in the end or after the argument.

In some arguments one character is in a dominant position (by means of the weapons, number of people, threatening etc. or due to the hierarchy). In that cases the oppressed party tries to loosen the dominance, while the other tries to keep pressure.

In a civilized debate, the characters do not shout at each other, but rather speak in turn. Insisting tone of voice and disagreement with the opponent distinguish the civilized debate from a simple discussion. It mostly happens for *good argue bad* case.

Special cases:

- Argument good-bad does not include “giving orders” and other 1-side arguments.

- In Rocky, main hero's coach (Mickey) usually criticizes his trainee and therefore their dialogues often resemble debates. If there is no serious debate, it is assigned a difficult *good argue good* label.
- Argument events do not include the introductory speech, but only the intense part.
- In Indiana Jones, discussion of the main hero with friends often looks like a difficult argument.

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *good argue bad* events

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *good argue good* events

Rambo *Rocky* *Die Hard* *Lethal Weapon* *Indiana Jones*

Sample snapshots from *bad argue bad* events

Bibliography

Trecvid multimedia event detection track. http://www.nist.gov/itl/iad/mig/med11.cfm.

Open video archive. http://www.open-video.org/.

Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

Robin Aly, Relja Arandjelovic, Ken Chatfield, Matthijs Douze, Basura Fernando, Zaid Harchaoui, Kevin Mcguiness, Noël O'Connor, Dan Oneata, Omkar Parkhi, Danila Potapov, Jerome Revaud, Cordelia Schmid, J.-L. Schwenninger, David Scott, Tinne Tuytelaars, Jakob Verbeek, Heng Wang, and Andrew Zisserman. The AXES submissions at TrecVid 2013, 2013. TRECVID Workshop, Gaithersburg, United States.

Joakim Andén and Stéphane Mallat. Multiscale scattering for audio classification. In *ISMIR*, 2011.

Joakim Andén and Stéphane Mallat. Deep scattering spectrum. *Transactions on signal processing*, 62:4114–4128, 2013a.

Joakim Andén and Stéphane Mallat. ScatNet (v0.2). http://www.di.ens.fr/data/software/scatnet/, 2013b.

Sylvain Arlot, Alain Celisse, and Zaid Harchaoui. Kernel change-point detection. arXiv:1202.3878, 2012.

Mohamed Ayari, Jonathan Delhumeau, Matthijs Douze, Hervé Jégou, Danila Potapov, Jérôme Revaud, Cordelia Schmid, and Jiangbo Yuan. Inria @ trecvid'2011: Copy detection & multimedia event detection. In *TRECVID*, 2011.

Lei Bao, Shoou-I Yu, Zhen-zhong Lan, Arnold Overwijk, Qin Jin, Brian Langner, Michael Garbus, Susanne Burger, Florian Metze, and Alexander Hauptmann. Informedia @ TRECVID 2011. In *TRECVID Workshop*, 2011.

Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009. ISSN 1935-8237. doi: 10.1561/2200000006.

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2009.

Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *ICCV*, 2013.

Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Liangliang Cao, Shih-Fu Chang, Noel Codella, Courtenay Cotton, Dan Ellis, Leiguang Gong, Matthew Hill, Gang Hua, John Kender, Michele Merler, et al. IBM Research and Columbia University TRECVID-2011 Multimedia Event Detection (MED) system. In *TRECVID Workshop*, 2011.

Liangliang Cao, Yadong Mu, Apostol Natsev, Shih-Fu Chang, Gang Hua, and JohnR. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, 2012.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Vasileios Chasanis, Argyris Kalogeratos, and Aristidis Likas. Movie segmentation into scenes and chapters using locally weighted bag of visual words. In *CIVR*, 2009.

Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.

Hui Cheng, Jingen Liu, Saad Ali, Omar Javed, Qian Yu, Amir Tamrakar, Ajay Divakaran, Harpreet S Sawhney, R Manmatha, James Allan, et al. Sri-sarnoff aurora system at trecvid 2012 multimedia event detection and recounting. In *TRECVID Workshop*, 2012.

Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *Transactions on Multimedia*, 2012.

Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*. 2008.

Franklin C Crow. Summed-area tables for texture mapping. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 207–212, 1984.

Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

Sandra de Avila, Ana Lopes, et al. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1): 56–68, 2011.

Claire-Hélène Demarty, Cédric Penet, Mohammad Soleymani, and Guillaume Gravier. VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. *Multimedia Tools and Applications*, pages 1–26, 2014.

Ajay Divakaran, KadirA. Peker, Regunathan Radhakrishnan, Ziyou Xiong, and Romain Cabasson. Video summarization using Mpeg-7 motion activity and audio descriptors. In *Video Mining*, volume 6. Springer, 2003.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

Matthijs Douze, Dan Oneata, Mattis Paulin, Clément Leray, Nicolas Chesneau, Danila Potapov, Jakob Verbeek, Karteek Alahari, Zaid Harchaoui, Lori Lamel, Jean-Luc Gauvin, Christoph Andreas Schmidt, and Cordelia Schmid. The INRIA-LIM-VocR and AXES submissions to Trecvid 2014 Multimedia Event Detection, 2014. TRECVID Workshop, Gaithersburg, United States.

Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. In *BMVC*, 2006.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010.

Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005.

David A Forsyth and Jean Ponce. A modern approach. *Computer Vision: A Modern Approach*, 2003.

Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *PAMI*, 2013.

Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *IJCV*, 107(3):219–238, May 2014.

G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

Zaıd Harchaoui and Olivier Cappé. Retrospective mutiple change-point estimation with kernels. In *Workshop on Statistical Signal Processing*, pages 768–772. IEEE, 2007.

Zaid Harchaoui, Francis Bach, and Eric Moulines. Kernel change-point analysis. In *NIPS*, 2008.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.

Minh Hoai and Fernando De la Torre. Max-margin early event detectors. In *CVPR*, 2012.

Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011.

Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 24(3):577–584, 2005.

Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *Proc. European Conference on Computer Vision*, 2014.

Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S Davis. Representing videos using mid-level discriminative patches. In *CVPR*, pages 2571–2578, 2013a.

Mihir Jain, Hervé Jégou, Patrick Bouthemy, et al. Better exploiting motion for better action recognition. In *CVPR*, 2013b.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

Steven M Kay. *Fundamentals of Statistical signal processing, Volume 2: Detection theory*. Prentice Hall PTR, 1998.

Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013.

Gunhee Kim, Leonid Sigal, and Eric P Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Learning Tree-structured Descriptor Quantizers for Image Categorization. In *BMVC*, 2011a.

Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, pages 1487–1494, 2011b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

Zhen-zhong Lan, Lei Bao, Shoou-I Yu, Wei Liu, and Alexander G. Hauptmann. Double fusion for multimedia event detection. In *Advances in Multimedia Modeling*, volume 7131 of *Lecture Notes in Computer Science*, pages 173–185. 2012.

Zhen-Zhong Lan, Lu Jiang, Shoou-I Yu, Shourabh Rawat, Yang Cai, Chenqiang Gao, Shicheng Xu, Haoquan Shen, Xuanchong Li, Yipei Wang, et al. CMU-Informedia at TRECVID 2013 multimedia event detection. In *TRECVID Workshop*, volume 1, page 5, 2013.

Ivan Laptev and Patrick Pérez. Retrieving actions in movies. In *ICCV*, 2007.

Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178, 2006.

Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011.

Kang Li, Sangmin Oh, AG Amitha Perera, and Yun Fu. A videography analysis framework for video retrieval and summarization. In *BMVC*, pages 1–12, 2012.

Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010.

Ying Li, Shrikanth Narayanan, and CC Jay Kuo. Content-based movie analysis and indexing based on audiovisual cues. *Circuits and Systems for Video Technology*, 14 (8):1073–1085, 2004.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches, ACL Workshop*, pages 74–81, 2004.

Yang Liu, Feng Zhou, Wei Liu, Fernando De la Torre, and Yan Liu. Unsupervised summarization of rushes videos. In *ACM Multimedia*, 2010.

David Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110, 2004.

Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013.

Yu-Fei Ma, Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. A generic framework of user attention model and its application in video summarization. *Transactions on Multimedia*, 2005.

Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge, 2008.

Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009.

Ayoub Massoudi, Frédéric Lefebvre, C-H Demarty, Lionel Oisel, and Bertrand Chupeau. A video fingerprint based on visual digest and local fingerprints. In *ICIP*, 2006.

Pradeep Natarajan, Prem Natarajan, Vasant Manohar, Shuang Wu, Stavros Tsakalidis, Shiv N Vitaladevuni, Xiaodan Zhuang, Rohit Prasad, Guangnan Ye, Dong Liu, et al. BBN VISER TRECVID 2011 Multimedia Event Detection system. In *TRECVID Workshop*, volume 62, 2011.

Pradeep Natarajan, Prem Natarajan, Shuang Wu, Xiaodan Zhuang, Amelio Vazquez-Reina, Shiv N. Vitaladevuni, Kleovoulus Tsourides, Carl Andersen, Rohit Prasad, Guangnan Ye, Dong Liu, Shih-Fu Chang, Imran Saleemi, Mubarak Shah, Yue Ng, Brand yn White, Larry Davis, Abhinav Gupta, and Ismail Haritaoglu. BBN VISER TRECVID 2012 Multimedia Event Detection and Multimedia Event Recounting Systems. In *TRECVID Workshop*, 2012a.

Pradeep Natarajan, Shuang Wu, Shiv Vitaladevuni, Xiaodan Zhuang, Stavros Tsakalidis, Unsang Park, and Rohit Prasad. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012b.

Pradeep Natarajan, Shuang Wu, Florian Luisier, Xiaodan Zhuang, Manasvi Tickoo, Guangnan Ye, Dong Liu, Shih-Fu Chang, Imran Saleemi, Mubarak Shah, Vlad Morariu, Larry Davis, Abhinav Gupta, Ismail Haritaoglu, Sadiye Guler, and Ashutosh Morde. BBN VISER TRECVID 2013 Multimedia Event Detection and Multimedia Event Recounting Systems. In *TRECVID Workshop*, 2013.

Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Video summarization and scene detection by graph modeling. *Circuits and Systems for Video Technology*, 15(2), 2005.

Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

Dan Oneata, Matthijs Douze, Jérôme Revaud, Schwenninger Jochen, Danila Potapov, Heng Wang, Zaid Harchaoui, Jakob Verbeek, and Cordelia Schmid. Axes at trecvid 2012: Kis, ins, and med. In *TRECVID Workshop*, 2012.

Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

Dan Oneata, Jakob Verbeek, and Cordelia Schmid. The LEAR submission at Thumos 2014. In *ECCV 2014 Workshop on the THUMOS Challenge 2014*, Zurich, Switzerland, September 2014.

Paul Over, Alan F Smeaton, and George Awad. The Trecvid 2008 BBC rushes summarization evaluation. In *2nd ACM TRECVID Video Summarization Workshop*, 2008.

Paul Over, George Awad, Jon Fiscus, Brian Antonishek, Martial Michel, Alan F Smeaton, Wessel Kraaij, Georges Quénot, et al. An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, 2011.

Paul Over, George Awad, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Martial Michel, Alan F. Smeaton, Wessel Kraaij, and Georges Quénot. TRECVID 2012 –

An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVID 2012*, 2012.

Paul Over, George Awad, Jon Fiscus, Greg Sanders, David Joy, , Martial Michel, Alan F Smeaton, Wessel Kraaij, and Georges Quénot. An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, 2013.

Paul Over, Jonathan Fiscus, Greg Sanders, David Joy, Martial Michel, George Awad, Alan Smeaton, Wessel Kraaij, and Georges Quénot. TRECVID 2014 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *ECCV*, 2014.

Danila Potapov, Matthijs Douze, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Beat-event detection in action movie franchises. *arXiv preprint arXiv:1508.03755*, 2015.

Lawrence R Rabiner and Ronald W Schafer. Introduction to digital speech processing. *Foundations and trends in signal processing*, 1(1):1–194, 2007.

Michalis Raptis and Leonid Sigal. Poselet key-framing: A model for human activity recognition. In *CVPR*. IEEE, 2013.

Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):23–38, 1998.

Yong Rui, Thomas S Huang, and Sharad Mehrotra. Exploring video structure beyond the shots. In *Multimedia Computing and Systems*, pages 237–240, 1998.

Yong Rui, Anoop Gupta, and Alex Acero. Automatically extracting highlights for TV baseball programs. In *ACM Multimedia*, 2000.

Jorge Sanchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Compressed fisher vectors for large-scale image classification. *IJCV*, 2013.

M. Schmidt. Ugm toolbox. http://www.cs.ubc.ca/ schmidtm/Software/UGM.html.

Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *ICPR*, volume 3, pages 32–36, 2004.

Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. Discriminative spatial saliency for image classification. In *CVPR*, 2012.

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher Vector Faces in the Wild. In *BMVC*, 2013.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

Blake Snyder. *Save the cat! The last book on screenwriting you'll ever need*. Michael Wiese Productions, 2005.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, pages 5179–5187, 2015.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, pages 787–802. Springer, 2014.

Hari Sundaram, Lexing Xie, and Shih-Fu Chang. A utility framework for the automatic generation of audio-visual skims. In *ACM Multimedia*, 2002.

Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

Tatiana Tommasi, Novi Quadrianto, Barbara Caputo, and Christoph H Lampert. Beyond dataset bias: Multi-task unaligned shared knowledge transfer. In *ACCV 2012*, pages 1–15. Springer, 2013.

Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, pages 776–789, September 2010.

Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):3, 2007.

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2007. ISSN 1572-2740. doi: 10.1561/0600000017.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

Jaco Vermaak, Patrick Pérez, Michel Gangnet, and Andrew Blake. Rapid summarisation and browsing of video sequences. In *BMVC*, pages 1–10, 2002.

Paul Viola and Michael J Jones. Robust real-time face detection. *IJCV*, 2004.

Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 2013.

Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid. A robust and efficient video representation for action recognition. *IJCV*, 2015.

Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010.

Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. Event driven web video summarization by tag localization and key-shot identification. *Transactions on Multimedia*, 14(4):975–985, 2012.

Xi Wang, Yu-Gang Jiang, Zhenhua Chai, Zichen Gu, Xinyu Du, and Dong Wang. Real-time summarization of user-generated videos based on semantic recognition. In *ACM Multimedia*, pages 849–852, 2014.

Daniel Weinland, Rémi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

Lexing Xie, Peng Xu, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. Structure analysis of soccer video with domain knowledge and hidden markov models. *Pattern Recognition Letters*, 25(7), 2004.

SI Yu, Z Xu, D Ding, W Sze, F Vicente, Z Lan, Y Cai, S Rawat, P Schulam, N Markandaiah, et al. Informedia@ TRECVID 2012. In *TRECVID Workshop*, volume 12, 2012.

Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73(2):213–238, 2007.

Bin Zhao and Eric P Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014.