# Visual Concept Learning from Weakly Labeled Web Videos

Adrian Ulges and Damian Borth and Thomas M. Breuel

**Abstract** Concept detection is a core component of video database search, concerned with the automatic recognition of visually diverse categories of objects ("airplane"), locations ("desert"), or activities ("interview"). The task poses a difficult challenge as the amount of accurately labeled data available for supervised training is limited and coverage of concept classes is poor. In order to overcome these problems, we describe the use of videos found on the web as training data for concept detectors, using tagging and folksonomies as annotation sources. This permits us to scale up training to very large data sets and concept vocabularies.
In order to take advantage of user-supplied tags on the web, we need to overcome problems of label weakness; web tags are context-dependent, unreliable and coarse. Our approach to addressing this problem is to automatically identify and filter non-relevant material. We demonstrate on a large database of videos retrieved from the web that this approach – called *relevance filtering* – leads to significant improvements over supervised learning techniques for categorization. In addition, we show how the approach can be combined with active learning to achieve additional performance improvements at moderate annotation cost.

Adrian Ulges
German Research Center for Artificial Intelligence (DFKI), D-67663 Kaiserslautern, Germany, e-mail: adrian.ulges@dfki.de

Damian Borth
University of Kaiserslautern, D-67663 Kaiserslautern, Germany e-mail: d_borth@cs.uni-kl.de

Thomas M. Breuel
University of Kaiserslautern, D-67663 Kaiserslautern, Germany e-mail: tmb@iupr.dfki.de

## 1 Introduction

Recent technological developments like high-speed internet and large-scale storage devices have made it possible for private users to generate, publish, and share large amounts of data. This has led to a break-through of digital video, which is now not only broadcasted by TV stations but is also produced, streamed and stored on a private basis. Web video portals like YouTube, blinkx, or myspace[1] have become essential sources of information and entertainment to millions of users, and it is fair to say that digital video is part of our everyday life, with massive amounts of content being viewed and stored [34, 44].

While video content is fairly simple to produce, finding the desired information becomes a difficult challenge as video databases grow larger and larger. The most comfortable way for users to express their information needs remains a text-based approach on the basis of keywords. This, however, requires an *indexing* that links the video content in a database to semantic concepts (or *tags*) appearing in it, like objects ("airplane"), scene types ("cityscape"), and activities taking place ("interview"). The challenge of creating such an index has been referred to as the *semantic gap* [36], the discrepancy between a video's low-level content on the one hand and the viewer's high-level interpretation on the other.

So far, the only reliable bridge over the semantic gap remains human perception. This means that – to build an accurate textual index for video search – human operators are required to manually label video content with concepts appearing in it. For many large-scale practical applications, however, this approach is simply too time-consuming. As a scalable alternative to complement human labeling, *concept detection* systems have been developed that infer the presence of tags automatically from the content of a video [7, 6, 46, 49]. Though such detectors do not reach a precision comparable to human annotators, they have been demonstrated to be extraordinarily useful in a video search context [38].

While concept detection is considered an approach of high potential for video search and has been realized in several research prototypes [7, 6, 19], it has not been widely applied in practical large-scale settings yet. One reason for this is that the supervised machine learning techniques underlying concept detection require video content labeled with target concepts for training. Currently, this training information is acquired manually, i.e. human operators label data with respect to concept presence. The quality of the resulting training material is high in a sense that the annotated concepts are carefully selected with respect to feasibility and usefulness [27], that clear and restrictive definitions of concepts are predefined, and that fine-grain annotation is done on shot level.

On the downside, the effort associated with such a time-consuming acquisition restricts concept detection in several ways: first, it limits the number of concepts to be learned, such that the size of current detector vocabularies is far from optimal [17]. Second, detectors have been reported to overfit to small training sets and generalize poorly [51]. Third, keeping track of dynamic changes of users' infor-

[1] http://www.youtube.com, http://www.blinkx.com, http://vids.myspace.com

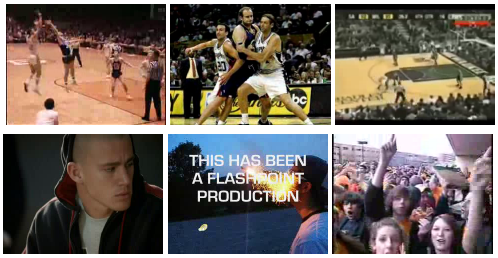



**Fig. 1** Sample frames from YouTube clips tagged with "basketball". While some frames do show basketball (top), other *non-relevant* content is not visually related to the concept (bottom).

mation needs is infeasible as new concepts of interest emerge (such as "President Obama" or "Olympics 2008").

Given this scalability problem, the question arises whether explicit manual annotations – which are precise but difficult to acquire – can be substituted with weaker label information that can be obtained more easily (or is even freely available). One source of such information is *web video*, which is publicly available at a large scale from portals such as YouTube and comes with tags indicating the presence of concepts in a clip. If we could utilize this tag information as class labels in a concept learning framework, systems could automatically harvest training material from the web. This way, detectors could perform a more autonomous learning, scale up to thousands of concepts, and keep track of trends in user interest.

Two key aspects are important: first, the described label information is significantly easier to acquire at a large scale, as the annotations used have already been made by a large community of YouTube users. Second, labels are *weak*, i.e. content annotated with a target concept *may* show the concept but does not necessarily do so. An illustration is given in Fig. 1, which shows representative keyframes from web videos tagged with "basketball". While the concept is present in some frames, others are not visually related to it at all. We will refer to the first kind of frames as *relevant*, while calling the latter *non-relevant*.

It should be noted that non-relevant content can be caused by different reasons: for once, tags are coarse and indicate *that* a concept appears in a video, but not *when* it appears. A second reason is that labels are inherently *unreliable* - for example, the tag "Steven Spielberg" does not necessarily indicate that Steven Spielberg appears in a clip but might just hint to a news report on the Academy Awards.

In the following, we will refer to training content where positive labels are only coarse and unreliable indicators of concept presence as *weakly labeled*. Obviously, training a concept detection system on such data is a difficult challenge: typically, for each target concept a binary classification problem is cast of differentiating concept presence from concept absence, and a statistical model is learned from a set of labeled training samples (here, keyframes). When applying such supervised learning to web video, non-relevant content causes *false positives* in the training set, and it is to be expected (and will be demonstrated later) that concept detection performance degrades with increasing influence of non-relevant content.

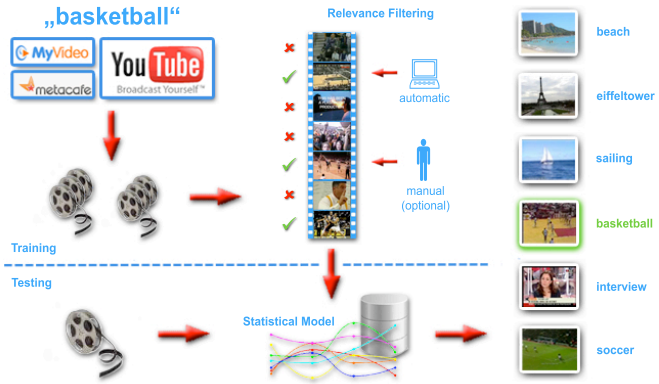



**Fig. 2** Our concept detection system learning from web video. Clips downloaded from online platforms like YouTube are used for concept learning. Since such videos are only weakly labeled, relevance filtering identifies and discards non-relevant content. The resulting statistical model is applied to detect the learned concepts in previously unseen videos.

In this chapter, this setup of training concept detectors on weakly labeled web video data is studied. We will show that the tag information associated with web video is in fact unreliable, and that non-relevant material in the training set can degrade the performance of standard detectors severely (Sect. 3). To overcome this problem, we present a framework for learning from weakly labeled web video data called *relevance filtering* (Sect. 4). This probabilistic approach views the given labels as weak indicators of true latent class labels, which are inferred during concept detector training. This corresponds to a filtering of non-relevant material, which can be applied as a wrapper around well-known supervised learning techniques. Two such extensions are presented, one for a generative approach (kernel density estimation) and one for a discriminative one (support vector machines [SVMs]). It is shown in quantitative experiments (Sect. 5) that relevance filtering can successfully identify non-relevant content and give significant improvements over standard supervised learning.

As outlined so far, relevance filtering works without manual supervision and identifies non-relevant material automatically based on its distribution in feature space. Beyond this, we also demonstrate that the approach can be extended with *active learning*. In this framework, the system requests labels for a few samples from a user (Sect. 6). By selecting the most *informative* query samples, concept models can be improved further at moderate annotation cost.

Overall, our contributions constitute an approach for concept learning from web video. As illustrated in Fig. 2 for the concept “basketball”, concepts are learned by acquiring a raw dataset from web portals like YouTube. Relevance filtering - which can be optionally enriched with thoroughly selected manual annotations – is used for a joint model learning and content filtering. The resulting concept detector can then be applied to previously unseen videos.

## 2 Related Work

In this section, related work in the context of learning from weakly labeled videos is outlined. We will omit a general review of concept detection (for more information on this topic, please refer to the literature [17, 27, 35, 39, 46, 49] or to a recent survey [37]) and focus on the aspect of weak label information instead. This setup is viewed from two different perspectives – the first tackles the machine learning aspects of the problem and discusses *semi-supervised* learning (Sect. 2.1). The second perspective is domain-specific and focuses on learning from noisy image and video content. (Sect. 2.2 and Sect. 2.3).

### *2.1 Semi-supervised Learning*

Semi-supervised learning refers to a class of machine learning techniques designed for dealing with incomplete label information. In this setup, a (usually small) set of training samples $X_L = \{x_1, .., x_l\}$ with class labels $y_1, .., y_l$ is assumed to be given. A second (usually large) set of samples $X_U = \{x_{l+1}, .., X_N\}$ is available as well, but the associated labels are unknown (or *latent*). Semi-supervised learning can be seen as a borderline case between supervised learning (where all training data is labeled, i.e. $X_U = \varnothing$) and unsupervised learning (where no labels are given at all, i.e. $X_L = \varnothing$).

Semi-supervised learning is attractive in application areas where lots of unlabeled training samples can be obtained easily, but the acquisition of label information is associated with considerable effort (as it is the case for concept detection). While supervised methods in such setups learn only from a small set of labeled examples, semi-supervised techniques can exploit further information in form of unlabeled content (which can be viewed as evidence on the overall sample distribution $p(x)$). To leverage this information, a variety of strategies has been proposed (for a survey of the field, please refer to [9, 52]).

One simple semi-supervised learning strategy is to infer the labels of unlabeled samples and treat the resulting labeled samples in a supervised framework. This **self-training** is an iterative wrapper around a base classifier, in which samples are iteratively classified and the training set is automatically expanded with a selection of the newly labeled data (usually the ones for which the classifier is most confident). As an extension, **co-training** [5] has been suggested, where multiple classifiers are trained on different feature subsets of the data and "teach" each other.

Another technique called **Expectation Maximization** (EM) [11] casts semi-supervised learning in a probabilistic setting. Model parameters are fitted by maximizing the data likelihood, whereas a marginalization over latent class labels is done. This leads to a search in parameter space in which alternately label posteriors are inferred, and based on these estimates the system parameters are updated.

An alternative strategy follows from the insight that decision boundaries are usually situated in low-density areas of $p(x)$, and should correspondingly lie far away from data points. If cast in a maximum-margin setting, this approach leads to **trans-**

**ductive SVMs** [20], which estimate unseen labels together with a separating hyperplane. Finally, **graph-based** methods have been proposed, which view samples as nodes in a graph and estimate node labels via label propagation or regularization [9, Ch. 11].

## *2.2 Learning from Web Images*

Web image content (as it can be acquired via text-based image search engines or from portals like Flickr) is a data source similar to web video content in a sense that label information is weak and large parts of the retrieved training data may be junk. For Google Image Search, Fergus et al. [14] have reported a label precision between 18% and 77% for 7 object categories. Schroff et al. [31] have measured an average precision of 39% over 18 categories.

To overcome this label weakness, a variety of approaches have been suggested [4, 31, 40, 50] targeted at a content-based refinement of raw web image sets. Usually, a three-step procedure is applied: first, a raw set of images is acquired from the web. Second, a subset of "good" candidate images for concept presence is selected, which can be done using manual annotation [4] or an analysis of text and meta-data surrounding the image [31, 50]. Finally, a statistical model of concept presence (a support vector machine [31], a region-level annotation model [3], or a mining procedure based on a saliency measure [40]) is trained on the refined image set and used to re-rank all web images. Similar to the work in this chapter, this approach is targeted at a refinement of training sets. However, it does not cover the actual learning of concept models. In contrast to this, we tackle a joint training set refinement and model learning, and our focus is on the performance of the resulting detectors.

Other related work follows an approach more similar to ours and combines training data refinement with model learning. Fergus et al. [14] learn visual models of object categories from Google's image search using a topic model. The key assumption of the approach is that images showing the target object accumulate in a single cluster (or *topic*), which is then used for object recognition. The OPTIMOL system by Li et al. [24] follows an incremental approach instead: a training set is agglomerated while learning an object model in parallel. The approach works in a self-training fashion, starting from an initial highly accurate set of sample images. Iteratively, a topic model is trained and the pool of training data is expanded using a Bayesian decision. The approach has been demonstrated to outperform Fergus' system [14]. Yet, a problem remains in the initialization with *good* training samples, which has been reported to be a crucial factor [26].

In contrast to the incremental OPTIMOL system [24], Wnuk and Soatto [48] follow a filtering approach. A measure of *strangeness* is defined based on a nearest neighbor analysis in feature space, and content with high strangeness values is filtered out. We follow this general idea and extend it to a probabilistic setting called *relevance filtering*, which can be integrated with a variety of supervised learning techniques (Sect. 4).

### *2.3 Learning from Weakly Labeled Videos*

Only few previous contributions have been made with respect to learning from video data with weak labels. Gargi and Yagnik [15] point out that label information in videos may be coarse, which they refer to as the *label resolution problem*. They rely on a feature selection using Adaboost to achieve robustness with respect to non-relevant content. Gu et al. [16] cast concept detection as a multiple instance problem and propose to adapt the kernel function in an SVM framework. Both methods, however, do not model non-relevant content explicitly.

A contribution closer to the one presented here has been made by Wang et al. [47], who study concept detection in a semi-supervised setup (where only a few initial labeled samples are given). A kernel density model is extended such that the contribution of each training sample is weighted by its class posterior, and an iterative fitting algorithm is proposed to match unlabeled content to classes. Performance improvements over supervised learning from a few initial samples are demonstrated.

In previous work, we have already addressed the problem of concept learning from weakly labeled web videos [43] and proposed a model similar to the one by Wang et al. In this chapter, we will extend this idea further and demonstrate that it can be integrated with a variety of supervised learning techniques.

## 3 Concept Learning on Web Video

In a first experiment, we study web video as a data source for concept detector training. First, we present manual annotation results demonstrating that the tag information coming with web videos is only an unreliable indicator of concept presence, such that web video training sets contain significant amounts of non-relevant content (Sect. 3.1). Second, we study how standard concept detection techniques are influenced by this non-relevant content (Sect. 3.2) and show that significant performance loss is to be expected.

### *3.1 The Precision of Web Video Tags*

In a first experiment, we study the precision of web video tags when used as class labels in a concept learning framework. Therefore, keyframes are sampled from YouTube videos and serve as positive training samples (if the video is tagged with the target concept) or as negative ones (if it is not). Since tags are coarse and unreliable, we expect that only a certain fraction of positive training samples is truly relevant. This *relevance fraction* is denoted with $\alpha$ in the following:

$$\alpha := \frac{\text{number of positive training samples showing the concept}}{\text{number of positive training samples}}$$

**Table 1** A manual annotation of training material downloaded from YouTube indicates that the label precision $\alpha$ of web video training sets is low (in most cases below 50%).

| Concept | Raw Query* | Refined Query* | Concept | Raw Query* | Refined Query* |
|---|---|---|---|---|---|
| basketball | 20.5 | 40.6 | helicopter | 14.6 | 38.1 |
| beach | 15.6 | 44.3 | sailing | 16.4 | 26.2 |
| cats | 47.6 | 50.1 | soccer | 25.3 | 43.7 |
| desert | 11.4 | 19.0 | swimming | 23.4 | 60.0 |
| eiffeltower | 21.4 | 39.7 | tank | 14.5 | 24.3 |
| | | | **average** | **21.1** | **38.6** |

* *values indicate fraction of relevant training content* $\alpha$ *(%)*

$\alpha$ can be seen as the *precision* of label information. It is close to 100% if annotations are accurate (as it is usually assumed in concept detector training). For web video, however, we expect $\alpha$ to be significantly lower and also to vary between concepts: while for some concepts high-quality training sets may be obtained, others may be used as tags often but *appear* only infrequently.

To get a deeper insight into the quality of tags as training annotations, we conducted an annotation experiment. Ten test concepts were chosen from the YouTube-22concepts dataset[2] with respect to a good coverage of concepts, including objects ("cats", "eiffeltower"), locations ("beach", "desert"), and sports ("basketball", "golf"). For each concept, $1,000$ clips were downloaded from YouTube using two different queries to the YouTube API (the overall length of the dataset is about 100 hours):

1. **Raw Queries**: The query consists of a single tag describing the concept, like "beach". This may be the case if a concept detection system is given only a vocabulary of tags and crawls YouTube fully automatically for training material.
2. **Refined Queries**: Querying the YouTube API with a single tag must be expected to give very noisy results. For example, the query "beach" does not only return scenes of beaches, but also music videos by the "Beach Boys" and scenes of Daytona Beach. While these may be valid annotations to the video owner, they must be considered non-relevant when it comes to learning a specific concept like beach sceneries. Therefore, two refinements are made. First, the fact is used that videos at YouTube are organized in categories like "Pets&Animals" or "Autos&Vehicles". The download is restricted to a canonical category (like "Travel&Places" for "beach", which excludes music videos). Second, queries are refined according to a brief analysis of the first YouTube results page. For example, the query "beach" is replaced with "walk on the beach", which usually rules out city names.

For each concept, a canonical definition was formulated (which can be found in the Appendix and is publicly available[3]), and over $1,000$ keyframes sampled

[2] http://www.dfki.uni-kl.de/~ulges/youtube-22concepts/

[3] http://www.dfki.uni-kl.de/~ulges/VSM-testconcepts/

from YouTube clips tagged with the concepts were manually assessed according to these definitions. Results of the annotation process are given in Table 1. They indicate that YouTube labels are in fact weak – the downloaded content contains significant fractions (in most cases more than 50%) of non-relevant material. It can also be seen that $\alpha$ is particularly low for raw queries (21.1% on average), whereas a manual refinement leads to better results (38.6%). Finally, the percentage of relevant material varies strongly between concepts: for example, the label precision ranges from 26.2% ("sailing") to 60% ("swimming") for refined queries.

These results correspond to similar observations made previously for the image domain: for datasets based on image search, a precision of 39% have been reported for object category recognition [31]. For Flickr images, Kennedy et al. [21] have observed an accuracy of 50% for the domain of New York sights. These precisions are slightly higher than our results, which can be attributed to the fact that for video the *coarseness* of labels in the time domain poses an additional problem. Yet, it should be noted that this does not necessarily mean that video is a worse source for visual learning than images. Rather, it is to be expected that the preferred training modality depends on the concepts: a wide variety of concepts are action-related or video-specific (for example, think of "soccer" or "interview"). For such concepts, video-based training material will be more appropriate than images.

## *3.2 Concept Learning from Web Video*

In the last section, YouTube datasets have been demonstrated to contain significant amounts of non-relevant content. The next key question is how this influences concept detectors when trained on web video using standard methods. Intuitively, it can be expected that material similar to false positives in the training set will be classified incorrectly, such that detection performance degrades. This is validated in the following experiment.

**Data** The experiment is conducted on the same YouTube data and annotations used in the last section. According to our ground truth labels, we randomly compiled training sets of varying noise ratio $\alpha$ as illustrated for the concept "desert" and $\alpha = 60\%$ in Fig. 3: negative samples – which can be obtained easily from videos not tagged with the concept – are drawn for the background class. Positive samples consist of 60% true positives (which were manually assessed to show the target concept) and 40% non-relevant frames, which were again drawn randomly from YouTube videos *not* tagged with the concept. Further, test sets with known ground truth labels were sampled:

for $\alpha \in \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.7,\ 1.0\}$:

1. *// sample training set*
   - sample 1000 non-relevant frames with label $-1$
   - sample $(1-\alpha)\cdot 500$ non-relevant frames with label 1 ("false positives")
   - sample $\alpha \cdot 500$ relevant frames with label 1 ("true positives")

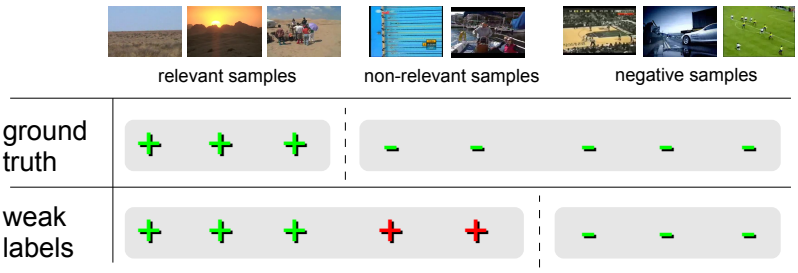



**Fig. 3** Sampling a random training set for the concept "desert" and $\alpha = 60\%$. Non-desert content models the background class (right). Positive samples are mixed of 60% desert frames and 40% non-desert frames. The latter are incorrectly labeled as relevant. This weakly labeled setup (top) is compared with learning from correct labels (bottom).

2. *// sample test set*
   - sample 500 relevant frames with label 1
   - sample 1500 non-relevant frames with label $-1$

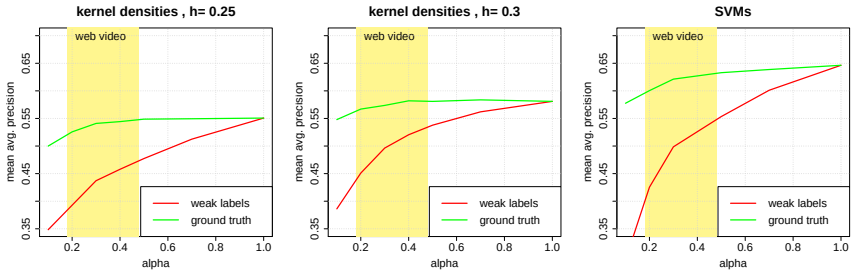
To avoid overfitting, it was made sure that no material from the same video clip was assigned to training and testing at the same time. Also, it should be noted that only the training set is weakly labeled, while the test set uses ground truth to assure a precise evaluation.

**Features** Frames are represented by bag-of-visual-words features [33], which have previously been demonstrated to give a good performance in a variety of recognition tasks including concept detection [45] or object category recognition [13]. For each frame, a feature is extracted by regularly sampling about $3,600$ SIFT patches [25] at several scales. These are matched with a $2,000$-dimensional visual codebook learned previously on a large dataset of 81 concepts. A dimensionality reduction is applied to the resulting visual word histograms using PLSA [18], obtaining a 64-dimensional feature vector per frame. This dimensionality reduction is done for efficiency purposes and has previously been validated to give comparable results to the high-dimensional visual word histograms.

**Models** Tests were run for two standard supervised learning approaches: a generative model (kernel densities) [12, Ch. 4] and a discriminative one (SVMs) [30]. Given training samples $x_1,..,x_n$ with labels $y_1,..,y_n \in \{-1,1\}$, the **kernel density** approach models class-conditional densities of concept presence and absence:

$$\begin{aligned} p^1(x) &= \frac{1}{Z} \sum_{i:y_i=1} K_h(x;x_i), \\ p^0(x) &= \frac{1}{Z'} \sum_{i:y_i=-1} K_h(x;x_i). \end{aligned} \tag{1}$$

A test frame $x$ is scored using Bayes' rule (the class prior – which does not influence the ranking of test items – is assumed to be uniform):




(a) (b) (c)

**Fig. 4** Comparing concept detection when trained on ground truth labels (green) and on weak labels (red). The mean average precision over all 10 test concepts is plotted against the label precision $\alpha$. The two results on the left represent the kernel density system for bandwidths 0.25 (a) and 0.3 (b), the result on the right is for SVMs (c).

$$P(y=1|x) = \frac{p^1(x)}{p^1(x)+p^0(x)}$$

As a kernel function, the well-known Epanechnikov kernel with Euclidean distance function is used:

$$K_h(x;x') = \frac{3}{4} \cdot \left(1 - \frac{||x-x'||^2}{h^2}\right) \cdot 1_{(||x-x'|| \leq h)}$$

This choice is made mostly for efficiency reasons: as the Epanechnikov kernel has only local support, it can be evaluated efficiently when combined with methods for fast nearest neighbor search such as kd-trees [28]. The kernel bandwidth $h$ is a free parameter of the system. It has been reported previously to have a strong influence on the resulting kernel densities [42], with high values of $h$ leading to a smoother density in feature space. For the experiment presented here, several choices of the kernel bandwidth $h \in \{0.225,\ 0.25,\ 0.275,\ 0.3,\ 0.325\}$ were tested, and results are reported for a representative low value ($h = 0.25$) and a high one ($h = 0.3$).

As a discriminative approach, **Support Vector Machines** (SVMs) [30] were tested, which are a popular choice for concept detection [45, 49, 51]. An RBF kernel is used, whereas the smoothness $\sigma$ and the cost $C$ are evaluated in a grid search cross-validation (for more information on these parameters, please refer to [8]). For efficiency reasons, no complete search was done for each run, but the values $C = 5$ and $\sigma = 25$ are used, which were validated to give stable good results. SVM scores were mapped to class posterior estimates using the LIBSVM implementation [8].

**Results** Both systems – kernel densities and SVMs – were tested in two setups (as illustrated in Fig. 3):

- **weak labels**: this setup corresponds to the practical situation of concept learning from weakly labeled web content. Only a fraction $\alpha$ of positive training samples is truly related to the concept.

- **ground truth**: this is a control run with an oracle providing ground truth labels (which are not available in practice). The run indicates how well concept learning would work if non-relevant content content was filtered out.

As a performance measure, average precision is used, i.e. the area under the recall-precision curve over the ranked list of test frames. By averaging over all 10 test concepts, we obtain the *mean average precision* (MAP), which is a standard choice for concept detector evaluation [22]. Quantitative results are given in Fig. 4 (all values were obtained by averaging over 5 runs). Both systems were tested for varying fractions of relevant content $\alpha$, and the system performance on the test data is plotted against $\alpha$.

We now study how concept detection behaves when varying the noise level in the training set. A first observation is that the influence of non-relevant material on the oracle-based control run (green) is negligible, such that performance remains almost constant when varying $\alpha$. This is intuitively correct, since non-relevant samples (which become more frequent with lower values of $\alpha$) are assigned their correct negative labels. A lower performance for low relevance fractions $\alpha \approx 10\%$ can be attributed to a lower absolute number of positive training samples.

When comparing the ground truth runs with systems trained on weakly labeled data, we can see that in the absence of noise ($\alpha = 1$) both systems give the same performance (which is trivial, as no false positives exist and training labels are identical otherwise). However, when decreasing $\alpha$ the performance of the weakly supervised system degrades significantly: for example, for training sets with 70% non-relevant material ($\alpha = 0.3$) and a bandwidth of 0.25, the kernel density estimation trained on weakly labeled data gives a performance of 43.7%, while training on the correct labels gives 54.1%. The more noise in the training data, the stronger the gap between the weakly supervised run and the control run becomes. This observation can be made for the generative model (Fig. 4(b) and 4(a)) as well as the discriminative one (Fig. 4(c)).

We can now match these results with the annotations in Table 1, which indicate that the label precision of web video data is typically in the range of 20% (for raw queries) to 50% (for refined queries). This range is highlighted in yellow in all plots. If we focus on this area, we can see that performance degradation due to weak labels is significant, ranging from 4% to 19%.

## 4 Relevance Filtering

Our results in the last experiment indicate that concept learning on web video could be improved significantly if we were able to filter out non-relevant content in the training set. In this section, we follow this strategy and present a framework in which the statistical models underlying concept detection are adapted such that non-relevant content is automatically identified and filtered during training. The approach is based on a formulation of concept learning as a *weakly supervised learning* problem, in which the given labels (here: YouTube tags) are only weak

**Table 2** An overview of the basic concepts and notation used in Sect. 4. Concept detection is viewed as a *weakly supervised learning problem*, in which given labels are only weak indicators of true, latent ones.

| | |
|---|---|
| $x$ | feature vector representing a test keyframe |
| $y \in \{-1,1\}$ | absence/presence of target concept in $x$ |
| $P(y=1\|x)$ | keyframe score (to be estimated) |
| $x_1,..,x_n$ | feature vectors representing training frames |
| $\tilde{y}_1,...,\tilde{y}_n \in \{-1,1\}$ | weak labels of concept presence in training frames (observed) |
| $y_1,.,y_n \in \{-1,1\}$ | actual absence/presence of target concept in training frames (unknown) |
| $\beta_j := P(y_i=1\|x_i,\tilde{y}_i)$ | *relevance score*: the probability of a training frame being relevant (unknown) |
| $\alpha := P(y_i=1\|\tilde{y}_i=1)$ | *relevance prior*: assumed fraction of truly relevant training frames among potentially relevant ones |

indicators of true class labels. These true class labels are inferred during concept learning.

This approach will be referred to as *relevance filtering* in the following. Its core assumption is that relevant content forms clusters in feature space, while non-relevant material comes as outliers that can be identified and relabeled. The approach can be combined with a variety of well-known supervised learning techniques. Two such combinations are presented for the models used in the last experiment (namely, kernel density estimation and Support Vector Machines).

## *4.1 Basic Concepts*

In the following, a video is represented by keyframes, such that concept detection is effectively conducted on keyframe level. Each frame is associated with a feature vector $x \in \mathbb{R}^d$. The presence of the target concept is denoted with a label $y$, such that $y=1$ indicates concept presence and $y=-1$ concept absence. The goal of concept detection is to estimate the concept *score* $P(y=1|x)$. Training data is also represented by keyframes (or associated features) $x_1,..,x_n \in \mathbb{R}^d$. For each training frame $x_i$, a weak indicator of concept presence is given that tells us whether the concept *may* appear in the frame (in practice, this is a tag given to the corresponding web video clip). This information is denoted by a *weak label* $\tilde{y}_i \in \{-1,1\}$. The *actual* presence of the target concept, however, is latent. It is denoted with $y_i \in \{-1,1\}$. Concept detection is now cast as a binary classification problem (see Table 2 for an overview of the notation used):

**Definition 1.** ***Weakly Labeled Binary Classification Problem***
Given training data in form of samples $x_1,..,x_n \in \mathbb{R}^d$ with labels $\tilde{y}_1,..,\tilde{y}_n \in \{-1,1\}$, learn a scoring function $\phi : \mathbb{R}^d \to [0,1]$ such that $\phi(x) \approx P(y=1|x)$. Thereby, training labels are assumed to be *weak* indicators of *true* labels $y_1,..,y_n$ such that:

1. If the weak label is negative ($\tilde{y}_i = -1$), the true label is negative as well ($y_i = -1$).
2. If the weak label is positive ($\tilde{y}_i = 1$), the sample *may* belong to the positive class, but does not necessarily do so, i.e. the true label $y_i$ is unknown.
3. A prior for weakly labeled samples being truly positive is assumed to be given, which is denoted with $\alpha := P(y_i = 1 | \tilde{y}_i = 1)$.

In this setup, true latent class labels are separated from given ones. They can thus be estimated during learning, such that the model $\phi$ is effectively trained on the estimated true class labels instead of the weak labels. It should also be noted that – while we model false positives (i.e. it is possible that $\tilde{y}_i = 1$ and $y_i = -1$) – false negatives ($\tilde{y}_i = -1$ and $y_i = 1$) are not taken into account. Strictly speaking, this is not true (for example, there might be videos showing "basketball" that the user has simply forgotten to label). According to our observations made on web video, however, the percentage of such false negatives is negligible compared to the one of false positives.

Let us compare the weakly labeled classification problem with other learning setups. First, when compared with standard *supervised learning*, two key differences are that only weak indicators of the true class labels are given, and that an additional assumption is made (in form of $\alpha$) on how much of the weakly labeled material does in fact show the target concept. Particularly, the supervised setting can be seen as a special case of the weakly supervised one, where $\alpha$ equals 100%.

Compared with the *semi-supervised learning* setup, the above definition can be seen as a degenerate special case. This is because weakly labeled samples $\{x_i : \tilde{y}_i = 1\}$ can be viewed as *unlabeled* (their true label $y_i$ is not known). This leads to an extremely *imbalanced* problem: while semi-supervised learning usually assumes a few initial labels of either class to be given, in our setup we are confronted with many samples from class $-1$ (simply because content not labeled with a concept can be obtained easily) but no sample of class 1 (since indicators of concept presence are weak). This renders a straightforward application of many semi-supervised algorithms impossible, since these would require an initialization with a few reliable samples of both classes.

Finally, the weakly labeled learning setup strongly resembles several approaches for visual learning from noisy image sources like Google's image search [14, 24, 48]. The work in this chapter follows a strategy similar to these approaches (particularly to the one by Wnuk and Soatto [48], who also propose a distribution-based filtering of training sets). Yet, several differences remain. First (and obviously), the web video domain addressed here differs from images delivered by web search engines. Second (and more importantly), we do not cover a single statistical model, but view relevance filtering as a wrapper than can be applied around a variety of supervised learning techniques. For both a generative and a discriminative base model, relevance filtering extensions will be presented in the following.

### *4.2 The Generative Case: Kernel Density Estimation*

In this section, relevance modelling is used as a wrapper around a generative model for concept detection, namely *kernel density estimation* [12, Ch. 4]. Thereby, the relevance of training content is modeled as a latent random variable that is inferred during the learning procedure.

**Class-conditional Densities and Scoring** Class-conditional densities of relevant and non-relevant content are modeled by the following weighted kernel densities $p^1_\beta$ and $p^0_\beta$:

$$\begin{aligned} p^1_\beta(x) &= \frac{1}{Z} \cdot \sum_{i=1}^{n} \beta_i \cdot K_h(x;x_i), \\ p^0_\beta(x) &= \frac{1}{Z'} \cdot \sum_{i=1}^{n} (1-\beta_i) \cdot K_h(x;x_i), \end{aligned} \tag{2}$$

where $Z = \sum_i \beta_i$ and $Z' = n - Z$ are normalization constants. Compared to the fully supervised setup from Equation (1), the key difference is that $p^1$ and $p^0$ are now parameterized by a vector $\beta = (\beta_1, .., \beta_n)$. This vector consists of *relevance scores* $\beta_i := P(y_i|\tilde{y}_i, x_i)$, which means that for $p^1_\beta$ each training sample is weighted by its probability of being relevant (correspondingly, for the distribution of non-relevant content $p^0_\beta$ this weight is $1-\beta_i$). Consequently, if a training sample is likely to be relevant, it has a strong influence on the distribution of relevant samples $p^1_\beta$ but low influence on $p^0_\beta$. In this way, the uncertainty of label information is taken into account (a similar model has been used in a semi-supervised setup before [47]).

Note that if we set the relevance scores according to the weak labels:

$$\beta_i = \begin{cases} 1, & \tilde{y}_i = 1 \\ 0, & \tilde{y}_i = -1 \end{cases}$$

the system degenerates to the standard supervised case (Equation (1)) in which all positively labeled samples are assumed to be relevant.

**Training** To compute the class-conditional densities $p^1_\beta$ and $p^0_\beta$, the vector of relevance scores $\beta$ must be inferred in system training. The input consists of features $x_1, .., x_n$, weak labels $\tilde{y}_1, .., \tilde{y}_n$, and the relevance prior $\alpha$. For each training frame $x_i$, three situations may occur:

1. $\tilde{y}_i = -1$ (*negative*): if $x_i$ is *not* labeled with the concept, it is assumed to be non-relevant, i.e. $\beta_i = 0$.
2. $\tilde{y}_i = y_i = 1$ ( *true positive*): $x_i$ is labeled with the concept and is in fact relevant. Accordingly, $\beta_i$ should be high.
3. $\tilde{y}_i = 1, y_i = -1$ (*false positive*): $x_i$ is labeled with the concept but is not relevant. Such noise samples may occur, since labels $\tilde{y}_i$ are only weak indicators of concept presence. For them, $\beta_i$ should be low.

Let us assume that $m$ training samples are weakly labeled with the concept, and that training samples are sorted such that $\tilde{y}_1 = .. = \tilde{y}_m = 1$ and $\tilde{y}_{m+1} = .. = \tilde{y}_n = -1$. While we know that $\beta_{m+1} = .. = \beta_n = 0$, the relevance scores $\beta_1, .., \beta_m$ need to be estimated, i.e. training must divide potentially relevant frames into actually relevant ones and non-relevant ones. Therefore, the parameter vector $\beta$ is restricted to the non-zero entries $\beta = (\beta_1, .., \beta_m)$.

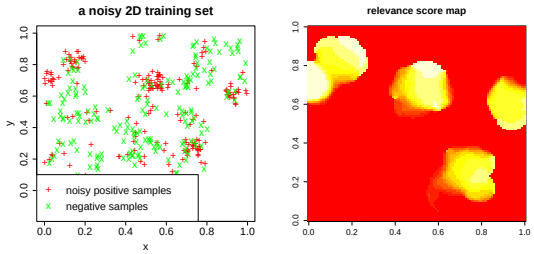
Our strategy to estimate $\beta$ is based on a simple fixpoint iteration in parameter space. First, relevance scores are initialized with the relevance prior: $\beta^0 = (\alpha, .., \alpha)$. Then, the parameter vector $\beta^k$ is iteratively updated to a new version $\beta^{k+1}$ by plugging the current parameter estimate $\beta^k$ into the class-conditional densities $p^1_{\beta^k}$ and $p^0_{\beta^k}$ (Equation (2)). From these densities, new estimates of relevance scores can be obtained using Bayes' rule:

$$\begin{aligned}
\beta_i^{k+1} &:= P(y_i = 1 | x_i, \tilde{y}_i = 1) \\
&= \frac{p(y_i = 1, x_i | \tilde{y}_i = 1)}{p(y_i = 1, x_i | \tilde{y}_i = 1) + p(y_i = -1, x_i | \tilde{y}_i = 1)} \\
&\approx \frac{P(y_i = 1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = 1)}{P(y_i = 1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = 1) + P(y_i = -1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = -1)} \\
&\approx \frac{\alpha \cdot p^1_{\beta^k}(x_i)}{\alpha \cdot p^1_{\beta^k}(x_i) + (1 - \alpha) \cdot p^0_{\beta^k}(x_i)}
\end{aligned}$$

This process is repeated for a fixed number of iterations. Intuitively, the algorithm identifies regions in feature space where positively labeled samples concentrate and assigns high relevance scores to them. Outliers similar to negative content are given low relevance scores. The approach resembles the well-known Expectation Maximization (EM) scheme [11], which maximizes the data likelihood by alternating so-called "E" steps (in which posteriors for latent variables are estimated) and "M" steps (in which system parameters are updated according to this knowledge by maximizing the expected log-likelihood of training data). If we compare the EM scheme to the fixpoint iteration used here, the relevance scores $\beta_i$ resemble posteriors for latent variables in the EM scenario (namely the true labels $y_i$). However, since the parameters of the class-conditional densities are equal to the relevance scores $\beta_i$ and the framework is non-parametric otherwise, no "M" step is required.

The approach is also similar to the training procedure used by Wang et al. [47], but the system is constrained in a different way: while Wang et al. addressed a semi-supervised setup – where initial reliable training samples for all classes are available – we cannot rely on such information in our weakly supervised setup. Instead, we constrain the system with a certain prior of the label precision $\alpha$. Note that if we choose this *relevance prior* to be $\alpha = 1$, it follows that $\beta_1 = \beta_2 = .. = \beta_m = 1$, such that the model degenerates to the supervised case (Equation (1)).

**A Sample Problem** In the following, an illustration of relevance filtering for kernel densities is given in a small experiment. A two-dimensional weakly labeled dataset is generated such that samples from the positive class contain a certain




(a) (b)

| error rate (%) | | | |
|---|---|---|---|
| | $\hat{\alpha}$ | | |
| $\alpha$ | 0.2 | 0.6 | 1.0 |
| 0.2 | 27.7 | 32.2 | 35.5 |
| 0.6 | 13.0 | 12.1 | 15.8 |
| 1.0 | 11.0 | 10.9 | 4.1 |

(c)

**Fig. 5** (a) A 2D sample training set. Positive samples (red) concentrate in 5 peaks, but contain 40% outliers. (b) A learned relevance map shows that relevant content is identified at the correct five peaks. (c) Classification error rates on synthetic sample sets, whereas the fraction of relevant content $\alpha$ and its estimate $\hat{\alpha}$ are varied. A choice of $\hat{\alpha} \approx \alpha$ gives the best classification results.

amount of incorrectly labeled false positives. For two classes (representing concept presence and absence), random prototypes are drawn from $[0,1]^2$. Samples are drawn from the surrounding of these prototypes according to kernel densities $p^1$ and $p^0$ with bandwidth $h = 0.05$, obtaining a training set with 200 noisy positive samples (which are again compiled of positive and negative content):

$$x_1,..,x_{200} \sim \alpha \cdot p^1 + (1-\alpha) \cdot p^0.$$

The fraction of relevant samples is varied such that $\alpha \in \{0.2,\ 0.6,\ 1.0\}$, i.e. we use one clean training set without false positives ($\alpha = 1.0$), one with moderate noise ($\alpha = 0.6$) and one with lots of noisy samples ($\alpha = 0.2$). For each training set, negative training samples are drawn from $p^0(x)$ and added. Finally, a test set of equal size is sampled from the same distribution as the training set. This experiment is repeated 100 times, whereas for each run the relevance filtering framework is tested with a relevance prior of $\hat{\alpha} \in \{0.2, 0.6, 1.0\}$. Note that the true relevance fraction is unknown in practice, which is why we distinguish between the true value $\alpha$ and the relevance prior we expect (which is denoted with $\hat{\alpha}$ in the following).

A typical dataset used in this experiment is illustrated in Fig. 5(a). It can be seen that positive samples (red) concentrate near five prototypes of class 1, but many red outliers (false positives) occur. The result of relevance filtering is also illustrated: a *relevance map* plots the relevance score $\beta$ over feature space (Fig. 5(b)). It can be seen that high relevance scores are assigned to samples accumulating near the five prototypes, while outliers close to negative samples are assigned low relevance scores. Classification results when applying the kernel density model with relevance filtering are reported in Table 5(c). Two observations can be made: first – and not surprisingly – the overall error rate of classification increases with the amount of noise material in the training set. The second observation is that the actual noise level and the optimal choice of the relevance prior are correlated, i.e. the lowest error rate is achieved for $\hat{\alpha} \approx \alpha$. For example, for the clean training set ($\alpha = 1$) the

supervised system ($\hat{\alpha} = 1$) performs best, while for $\alpha = 0.2$ the best performance is achieved for $\hat{\alpha} = 0.2$. Generally, this result indicates that relevance filtering can improve kernel density classification on weakly labeled training sets.

**Table 3 Weakly Supervised Discriminative Training**: Samples are iteratively refined in a self-training fashion by learning a discriminative classifier, scoring training content, and relabeling the samples most likely to be false positives.

1. for i=1,.,n: set $\beta_i = \begin{cases} 1, & \tilde{y}_i = 1 \\ 0, & \tilde{y}_i = -1 \end{cases}$
2. randomly split $X = \{x_1, .., .x_n\}$ into five folds $X_1, .., X_5$
3. until $\frac{1}{p}\sum_{i=1}^{p} \beta_i \leq \alpha$:
   - for $k = 1, .., 5$:
     - train a classifier on $X \setminus X_k$
     - apply the classifier to $X_k$, obtaining *scores* $\sigma$
     - for the $N_f$ samples $x_i \in X_k$ with $\beta_i = 1$ and lowest scores $\sigma(x_i)$: set $\beta_i = 0$

## *4.3 The Discriminative Case: Support Vector Machines*

While in the last section a generative technique was adapted for weakly labeled concept detection, a similar extension will be presented for discriminative models in the following. The approach can be applied as a wrapper around a variety of discriminative base classifiers. The only requirement on the base model is that it delivers a posterior-like *score* $\sigma$. As a sample classifier, SVMs are used (which can be considered a standard choice for concept detection [45, 49, 51]).

The basic idea of relevance filtering for discriminative methods is similar to a semi-supervised self-training but works in a filtering fashion instead of an incremental one: iteratively, the base classifier is trained and used to identify false positives in the training set. Samples that the classifier identifies as most likely to be false positives are relabeled (i.e., their relevance scores $\beta_i$ are set from 1 to 0), and training is repeated. This way, the weakly labeled positive samples are iteratively filtered and refined. The whole process is repeated until the estimated relevance prior $\frac{1}{p}\sum_i \beta_i$ (which constantly decreases due to relabeling) reaches the expected relevance prior $\alpha$. The whole training procedure is outlined in Table 3 (note that filtering is done in a cross-validation fashion to avoid overfitting).

Let us compare the approach with the generative relevance filtering from the last section. Generally, both techniques follow the same idea, namely to estimate the relevance of training content using the distribution in feature space and a relevance prior. However, two key differences can be identified. First, while the generative approach relied entirely on the distribution of content in feature space, the discriminative technique involves a classification method, such that the quality of filtering

results is inherently bound to the classifier used. Second, the discriminative relevance filtering approach is not probabilistic: the scores $\sigma$ used for filtering may be interpretable as relevance posteriors but do not necessarily have to be. Also, no soft assignment is used (as for kernel densities), but a complete relabeling of samples from the positive to the negative class takes place.

## 5 Experiments with Automatic Relevance Filtering

In the last section, relevance filtering has been proposed as a strategy to overcome label unreliability in concept detection training sets, and based on this idea extensions of two standard techniques (kernel densities and SVMs) have been presented. In practice, however, such an automatic filtering – which is entirely based on the distribution of content in feature space – is not 100% accurate. Therefore, we need to investigate how well relevant content be separated from non-relevant one, and whether the performance of concept detection can be improved this way. In the following, it is demonstrated that the filtering of non-relevant content is possible (though far from perfect), and performance improvements of up to 9% compared with an equivalent supervised system are validated when false positives are drawn from an overall "world" distribution (Sect. 5.1). After this, the relevance filtering framework is trained on raw web video content downloaded from YouTube (where non-relevant material is correlated with the concept), and it is shown that relevance filtering still gives performance improvements in the range of $2-5\%$ over the supervised case (Sect. 5.2).

### *5.1 Controlled Setup*

The purpose of this experiment is to study relevance filtering in a controlled scenario with known relevance fraction $\alpha$. The setup is almost identical to the one used in Sect. 3: the same randomly sampled training sets and test sets are used, results are averaged over 5 runs, the feature representation remains the same (visual words, followed by a dimensionality reduction using PLSA), and the same statistical models are tested (namely kernel density estimation and Support Vector Machines). The only difference is that – besides the control runs used in Sect. 3.2 – additional results for relevance filtering extensions are presented. The following approaches are tested:

1. **ground truth**: the control run from Sect. 3.2 trained on ground truth labels.
2. **weak labels**: supervised learning from Sect. 3.2 trained on weak labels.
3. **relevance filtering – kernel densities**: the relevance filtering extension of the generative kernel density approach from Sect. 4.2. The number of training iterations is set to 100. The relevance prior is set to the correct fraction of relevant

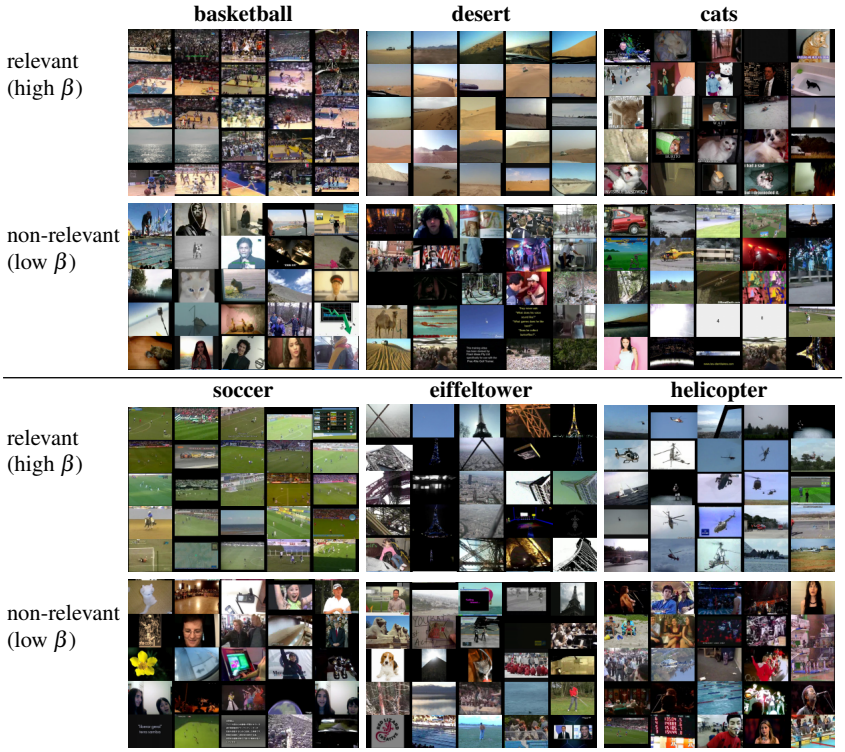
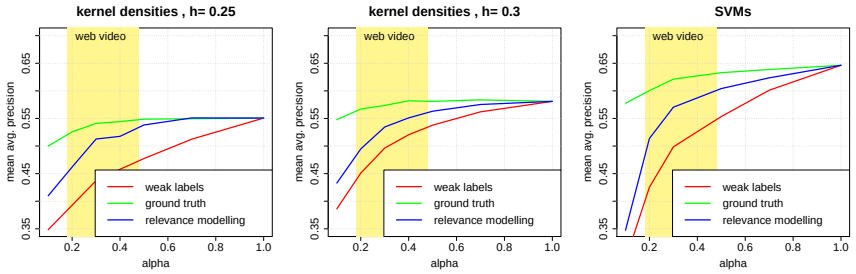



**Fig. 6** Results of relevance filtering (using the generative approach): for six concept, the frames are displayed that the relevance filtering approach learns to be most relevant (top) and least relevant (bottom). Relevance filtering works in general, and non-relevant content – though labeled with the concept – can be identified. However, the quality of filtering seems strongly related to concept difficulty: for example, compare "cats" (top right) with "soccer" (bottom left).

material, i.e. $\hat{\alpha} := \alpha$ (the behavior when varying this parameter will be studied later). The run appears in Figs. 7(a) and 7(b).

4. **relevance filtering – SVMs**: the relevance filtering extension of the discriminative approach from Sect. 4.3 using SVMs as base classifiers. Ten false positives are filtered in each training iteration. The same smoothness parameter $\sigma = 25$ and cost parameter $C = 5$ are used as in Sect. 3. The run appears in Fig. 7(c). Again, we set $\hat{\alpha} := \alpha$.

We first visualize the effects of relevance filtering in Fig. 6 to find out what content is actually identified as non-relevant by the system. Positive training frames are ranked by their score $\beta_i$, and the images with highest scores and lowest scores are displayed in Fig. 6 (a training set with $\alpha = 0.3$ was used, a bandwidth of 0.275, and a relevance prior of 0.3). At the top, we see the content identified to be most relevant, i.e. the highest scores $\beta$ were assigned. Below this, material is illustrated that was labeled with the concept but was identified to be non-relevant by our sys-




(a) (b) (c)

**Fig. 7** Results of relevance filtering for kernel densities (a,b) and SVMs (c). Performance is plotted against the relevance fraction $\alpha$. It can be seen that relevance filtering (blue) – though not achieving the performance of a hypothetical perfect relevance filter (green) – gives significant improvements over its standard supervised equivalent (red).

tem. Obviously, the content identified as relevant is in fact very likely to be visually related to the concept, and non-relevant material – though labeled with the target concept – tends to be identified successfully.

Quantitative results of the experiment are illustrated in Figs. 7(a) and 7(b) (for the generative model) and in Fig. 7(c) (for the discriminative one). Performance is plotted against the label precision $\alpha$ in a similar fashion as in Sect. 3. In contrast to earlier results, however, relevance filtering is included.

One first observation is that for $\alpha = 1$ all methods perform equally well (which can be observed both for kernel densities and SVMs). This is trivial as the systems are all trained on the same labels – no false positives exist and no filtering takes place. However, with decreasing $\alpha$ – i.e. with increasing non-relevant content in the training set – differences can be observed. It can be seen that relevance filtering (though not reaching the performance of the oracle-based control run) significantly outperforms standard supervised learning. For example, for a bandwidth $h = 0.25$ and a relevance fraction of $\alpha = 0.3$, relevance filtering gives an improvement from 44% to 51%. For a higher bandwidth of 0.3 this improvement is lower, which can be explained by the fact that the supervised baseline is more competitive due to a stronger smoothing.

When comparing the results for kernel densities with the ones for the SVM approach, similar observations can be made for the discriminative model: due to inherent errors of filtering, relevance filtering does not reach the performance of the oracle-based control run, but it significantly outperforms its standard supervised counterparts.

Finally, the experiment also indicates for which label precisions relevance filtering is the most promising. If the training set is extremely noisy ($\alpha \leq 10\%$), a fully automatic relevance filtering becomes difficult. This can be observed in Fig. 7(c), where for the leftmost point ($\alpha = 10\%$) the improvement by relevance filtering is only weak. On the other hand, for high values of $\alpha$ the supervised baseline is already quite competitive. For moderate values of $0.2 \leq \alpha \leq 0.5$, the benefits of relevance

**Fig. 8** Non-relevant content from web videos labeled with "Eiffel Tower", which indicate that noise content in real-world training sets is correlated with the target concept. This renders a fully automatic relevance filtering based only on the distribution in feature space a difficult challenge.

filtering are most prominent. According to this result, relevance filtering is of particular interest for web content, which comes with noise ratios in the same range. Here, performance improvements in the range of $3-9\%$ are achieved.

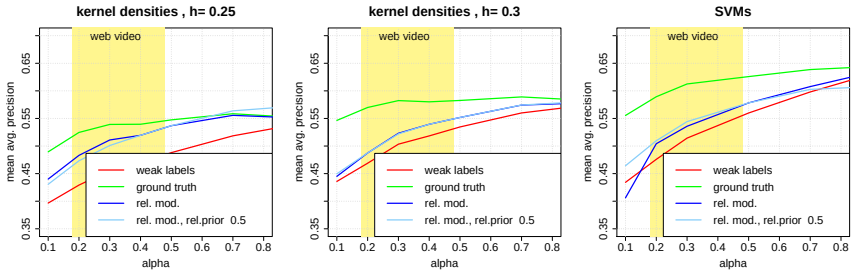
## *5.2 Raw Web Video Content*

The purpose of the last experiment was to give a proof-of-concept for relevance filtering in a controlled setup, where the ratio of relevant material is known. In this case, relevance filtering was demonstrated to outperform supervised models significantly.

In the following, we test relevance filtering on training sets of real-world web video content. In contrast to the controlled setup studied in the last section, there are two key differences. First, the fraction of relevant content is not known a priori when downloading raw material from YouTube. A simple workaround for this is to set $\hat{\alpha}$ to a "reasonable" value like 0.5, which will be demonstrated to give comparable results to using the true relevance prior. The second issue is related to the non-relevant samples themselves: while the proposed approach assumes such false positives to be drawn from an overall "world" distribution, non-relevant content in practice depends strongly on the concept. For example, non-relevant material in "basketball" videos tends to show scenes of a cheering crowd, while non-relevant material for the concept "eiffeltower" contains many urban scenes of Paris (a few typical false positives from "Eiffel Tower" videos are displayed in Fig. 8). Note that – since relevance filtering is entirely based on the fact that relevant content forms peaks in feature space – non-relevant material forming similar peaks (for example "shots of Paris") may be difficult to separate from truly relevant content.

We use a similar setup as in previous experiments, i.e. training sets of known noise ratios are randomly compiled (the same sample numbers are used as described in the last section). The key difference is that false positives – which were previously sampled from videos *not* labeled with the concepts – are now drawn from clips tagged with the concept (but are still non-relevant according to manual annotation). Correspondingly, negative test samples now consist of 500 frames from videos tagged with the concept and $1,000$ frames from other videos.

Figure 9 also tackles the question how to estimate the relevance prior $\alpha$. It suggests a very simple solution, namely to set it to a "reasonable" choice such as 0.5




(a) (b) (c)

**Fig. 9** Comparing relevance filtering on raw web video training sets when using the correct relevance prior ($\hat{\alpha} = \alpha$, dark blue) with a default choice of $\hat{\alpha} = 0.5$ (light blue). It can be seen that a simple choice of $\hat{\alpha} = 0.5$ leads to comparable results.

(which corresponds to a typical value for web-based training sets as shown in Sect. 3). Figure 9 compares relevance filtering when using the true prior $\hat{\alpha} = \alpha$ and when using $\hat{\alpha} = 0.5$. It can be seen that by simply setting $\hat{\alpha} = 0.5$ (i.e. by filtering half of all positive training samples) a stable performance can be obtained that is comparable to the true relevance fraction, at least for the range of $\alpha = 0.2 - 0.5$ typical for web video. For the SVM approach and very noisy training sets ($\alpha < 0.2$ in Fig. 9(c)), this choice even outperforms a more aggressive filtering as for $\hat{\alpha} = \alpha$.

## 6 Relevance Filtering with Active Learning

In previous experiments, it was shown that automatic relevance filtering improves concept detection performance by up to 9% for real-world web video material when run in an controlled setup. When trained on raw web video content as downloaded from YouTube, it was shown again that automatic relevance filtering gives performance improvements. These improvements, however, are lower than for the controlled setup, which can be explained by the fact that non-relevant content tends to be concept-dependent in practice (for example, noise material tagged with “Eiffel Tower” tends to show urban scenes of Paris, as illustrated in Fig. 8). Such content forms clusters in feature space similar to relevant material and cannot be separated easily.

This raises the question whether a better filtering could be achieved using a little manual supervision, i.e. by requiring human operators to provide a few selected labels. Here, active learning techniques – where the system selects informative examples for the user to annotate [32] – is an interesting extension to the current relevance filtering framework.

The approach has already proven to be successful in the large-scale concept detection evaluation TRECVID [2], where training examples for a concept of interest are accumulated from a completely unknown video database. This setup (which

usually starts from few reliable initial labels [1, 2, 10]) differs from the one studied in this chapter, as we focus on a *refinement* of large and only partly non-relevant training sets. Despite this difference, however, such an extension fits quite elegantly into the proposed learning framework: whenever a user annotates a training sample $x_i$, the relevance score $\beta_i$ is adapted and fixed to the given label, and learning is re-iterated.

In this section, an active learning extension to relevance filtering is presented. Also, we compare several active learning sample selection strategies in an experiment and show that the approach – if integrated with the kernel density version of relevance filtering – leads to significant improvements of concept learning from web video at moderate annotation effort.

## *6.1 Basic Concepts*

In Sect. 4.2, a generative approach using kernel density estimation has been extended such that relevance scores $\beta_i$ capture the uncertainty of the given label information. To reduce this uncertainty, we propose an iterative manual refinement of selected samples and successive retraining. Such a relevance feedback mechanism can be placed as a wrapper around automatic relevance filtering. The procedure is illustrated in Table 4: iteratively, relevance filtering training is applied, obtaining relevance scores $\beta$. Based on these scores, the most informative weakly labeled sample is selected for manual annotation (note that – as according to Definition 1 only positive labels are unreliable – we focus on the positive samples). After manual labeling of the selected sample $s^*$, we can fix $\beta_{s^*}^j$ to either 0 or 1 depending on the received label. This new information is used for retraining relevance filtering, providing new relevance scores $\beta_i^{j+1}$ for the next iteration of sample selection. With increasing iterations of such *active learning*, the procedure separates relevant content from non-relevant one more reliably.

**Table 4 Active Learning Extension**: Wrapped around relevance filtering, active learning selects informative samples for refinement by a human operator. Once the label is given, its relevance score is set to either 0 or 1, the system is re-trained, and the remaining relevance scores are adapted.

1. for $j = 1,.,m$ do:
   - train relevance filtering and get relevance scores $\beta^j = \{\beta_i^j\}$
   - select sample $s^*$ according to an *active learning* criterion $Q$:

$$s^* := \underset{i:\tilde{y}_i=1}{\arg\max} \; Q(\beta_i^j)$$

   - annotate manually, obtaining the true label $y_{s^*}$
   - fix the relevance score $\beta_{s^*}^{j+1,..,m} = \begin{cases} 1, & y_{s^*} = 1 \\ 0, & y_{s^*} = -1 \end{cases}$

### 6.2 Active Learning Methods

Different sample selection strategies for active learning have been proposed in the literature (see [1, 10, 41] for work in the video retrieval domain and [32] for a more complete survey). We test a few of the most popular ones in combination with relevance learning. These strategies select samples based on their a class posterior (which in our case corresponds to the *relevance score* $\beta$).

1. **random sampling**: samples are selected randomly (serves as a baseline).
2. **most relevant sampling**: samples are selected which are most likely to be relevant and are therefore associated with a maximum relevance score $\beta$. This approach was first introduces in information retrieval [29] but has also been proven to be a good option in a concept detection setup [1, 2]:

$$Q_{REL}(\beta) := \beta$$

3. **uncertainty sampling**: samples are selected for which the relevance filtering method is most inconfident, i.e. $\beta \approx 0.5$ [23]:

$$Q_{UNC}(\beta) := 1 - |\beta - 0.5|$$

### 6.3 Experiments with Active Relevance Filtering

In the following, we run an experiment on raw web video similar to Sect. 5.2 and apply the active learning extension to relevance filtering with the goal to improve concept detection performance further.

The dataset used is equal to the one in Sect. 5.2. The kernel bandwidth is fixed to $h = 0.275$. The key difference to Sect. 5.2 is that the relevance fraction is fixed at $\alpha = 0.2$, which poses a difficult challenge to automatic relevance filtering as the majority of content is non-relevant.

Results averaged over 5 runs are illustrated in Fig. 10. In contrast to the previous experiments, performance is plotted against the number of manually annotated samples ($\alpha = 0.2$). Also, the results from previous experiments in Sect. 5.2 can be found in the plot as horizontal lines: *no relevance filtering*, *automatic relevance filtering*, and *ground truth* training.

Now, we study how concept detection performance increases if we iteratively replace weak labels – potentially associated with false positives – with true labels provided by human annotations. As seen in Fig. 10 the performance of the different active learning methods lies within a corridor bounded by automatic relevance filtering (bottom) and the ground truth run (top). Starting with no refined samples, the performance equals the one of automatic relevance filtering. With more manual annotations, performance increases and finally converges to the ground truth case.

When comparing the different active learning strategies, it can be seen that they both outperform random sampling significantly, and that the best performance is

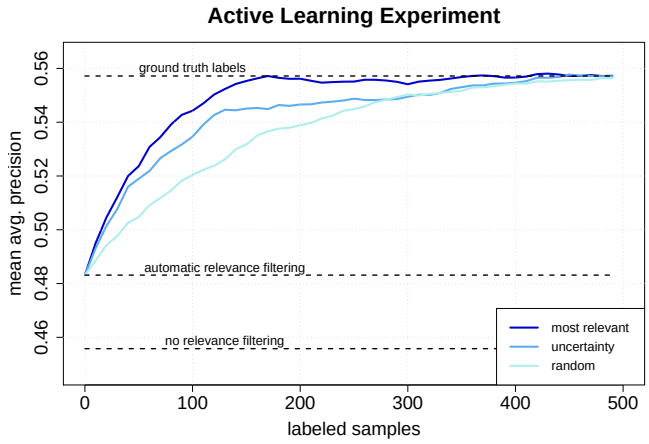



**Fig. 10** Results of active learning for relevance filtering with generative kernel densities. The label precision $\alpha$ is fixed at 0.2. Performance is plotted against the number of manually annotated training samples. It can be seen that – if using a proper sample selection – it is sufficient to annotate only $30-40$ weakly positive training samples to achieve a significant performance improvement.

achieved by *most relevant sampling*, which mines the dataset for truly relevant samples. This can be explained by the fact that relevance filtering relies strongly on correct relevance weights. Consider the example of "eiffeltower" and assume that the illustrated false positives in Fig. 8 all belong to the same cluster. If the system misleadingly assigns high relevance score to this cluster, one of the samples will be selected early for manual refinement. The false positives will be identified, corrected, and further iterations of relevance filtering will propagate this new information among the cluster in giving neighboring samples lower relevance scores.

Overall, it can be seen that with active learning we can improve performance at moderate annotation cost. For example, with as few as 40 annotations, a performance increase of about 4% is achieved. When continuing with annotation, we can see that concept detection performance converges to the ground truth case at $100-150$ iterations (which corresponds to only $25-30\%$ of the overall dataset). Concluding, relevance filtering – if combined with appropriate active learning strategies – can improve concept learning on the difficult domain of raw web video content.

## 7 Conclusions

In this chapter, the visual learning of concept detectors from weakly labeled videos has been addressed. Such data offers a scalable alternative to the conventional manual acquisition of training data, as label information can be acquired without manual

overhead. On the other hand, class information becomes unreliable, and labels are only weak indicators of concept presence. We have demonstrated for the domain of web video that typical training sets include significant amounts of non-relevant noise material, with a resulting performance degradation of up to 20%.

To overcome this problem without additional manual overhead, a framework called *relevance filtering* has been proposed. A binary classification problem is cast for each target concept, whereas true (latent) concept presence is inferred during system training. Based on this idea, two relevance filtering extensions of well-known supervised techniques were presented, one for a generative approach (kernel densities) and one for a discriminative one (Support Vector Machines). In experiments on real-world web video material, it was demonstrated that relevant content can be separated from non-relevant one in general, and that the performance of concept detection can be improved by up to 9% in a controlled setup. When trained on raw web video content as downloaded from YouTube, relevance filtering still improves over the supervised case though by a lower margin, which is because non-relevant content shows a tendency to be concept-dependent. To handle such conditions, an extension to relevance filtering was introduced, where minimal manual supervision adapts the scores of the weakly labeled training samples. In particular, by utilizing active learning methods for sample selection, the system could improve by up to 8% by refining only $25-30\%$ of weak positive labels from the training set.

As the approach in this chapter focuses entirely on a *visual* learning of concepts, one interesting extension of the framework would be the additional use of tag information coming with web video clips for relevance filtering. While we currently make only very limited use of such meta information, we envision our future system to use tag information directly in the refinement process. Particularly, *deep tagging* – where users provide detailed tags at certain time stamps in a video – might be an interesting clue to overcome the *label coarseness* problem.

**Acknowledgements** This work was supported by the German Research Foundation (DFG), project MOONVID (BR 2517/1-1).

## Appendix

This section gives a description of the concepts used in our experiments. We have defined canonical definitions of each concept and performed a manual annotation of web-based material. Table 5 provides the definitions as well as information on how video data was downloaded from YouTube.

**Table 5** Meta Information regarding the 10 test concepts used in the experiments

| Concept | Description | YT Query* | YT Category* |
|---|---|---|---|
| basketball | Scenes showing people playing basketball. Includes streetball if recognizable as such. | basketball<br>basketball nba<br>basketball dunking<br>basketball best moves<br>basketball dunks | sports |
| beach | Scenes showing a beach. Water does not have to be visible (if anything else qualifies the scene as showing a beach). Shots from a distance qualify as well, but only if the coastline is clearly a beach. | walk on the beach<br>beach sunbath<br>beach hawaii<br>beach panorama<br>beach malibu day | travel&places |
| cats | Scenes showing one or multiple cats. Closeups qualify as well as full body shots. | cats<br>cats funny<br>cats pets animals<br>cats playing<br>cats eating | pets&animals |
| desert | Scenes showing desert landscape. Panoramic shots involving significant amounts of sky are allowed (as long as some desert landscape is visible at the bottom). Things like plants, rocks, canyons, cars, etc. are allowed, but the landscape should show desert. | desert egypt<br>driving through desert<br>desert panorama<br>desert sahara<br>desert trip | travel&places |
| eiffeltower | Scenes showing the Eiffel Tower. Views from top of the tower qualify only if you see a part of tower (like parts of the steel construction). Night shots qualify. Closeups showing only parts of the steel construction qualify (if the tower can be identified) as well as panoramic shots from the distance. Shots with people in the foreground and the tower in the background count as well. | tour eiffel,<br>eiffel tower,<br>eiff.t. paris france,<br>eiffelturm paris | travel&places |
| helicopter | Scenes showing a helicopter (airborne or on the ground). Views from inside the helicopter are allowed if they can be identified as such. Only instruments or the pilot are not sufficient. Shots of toy helicopters qualify as well. | helicopter,<br>helicoptero,<br>helicopter flying,<br>helicopter landing | autos&vehicles |
| sailing | Scenes showing sailing ships/boats on the water/in the harbor. Panoramic views from onside a boat qualify if you see a part of the boat (like sails). Catamarans qualify as sailing ships, but surf boards or tankers do not (generally, everything with a sail qualifies). | sailing,<br>sailing trip,<br>sailing boat,<br>sailing holiday,<br>sailing mediterranean | travel&places |
| soccer | Shots showing a soccer match. Actions only distantly related to soccer do not qualify (like people doing soccer tricks in the street). Close-ups of players are allowed as well as global shots (if clearly identifiable as soccer). Soccer fields without action qualify as well. Shots of a cheering crowd do not qualify. | soccer bundesliga,<br>soccer goals,<br>soccer match,<br>soccer game outdoor,<br>fussball spiel | sports |
| swimming | Scenes showing somebody swimming. A swimming pool counts too (even if nobody is swimming inside it). Also includes swimming objects (fish, bottles). | swimming,<br>swimming pool -clean,<br>swimming technique,<br>sw. competition,<br>swimming olympics,<br>sw. championship | sports |
| tank | Scenes showing a tank, i.e. a heavily armored vehicle. Any scene qualifies if a part of the tank is visible such that the tank is identifiable. Other sorts of military ground vehicles qualify. | tanques,<br>tank,<br>tank battle,<br>panzer,<br>tank fire -flashpoint | autos&vehicles |

* *Values used for YouTube API calls*

## References


1. S. Ayache and G. Quenot. Evaluation of Active Learning Strategies for Video Indexing. *Signal Processing: Image Communication*, 22(7-8):692–704, 2007.
2. S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In *Proc. Europ. Conf. on Information Retrieval*, pages 187–198, March 2008.
3. K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching Words and Pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.
4. T. Berg and D. Forsyth. Animals on the Web. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1463–1470, June 2006.
5. A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. In *Proc. Ann. Conf. on Computational Learning Theory*, pages 92–100, July 1998.
6. C. Snoek et al. The MediaMill TRECVID 2007 Semantic Video Search Engine. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.
7. M. Campbell, A. Haubold, M. Liu, A. Natsev, J. Smith, J. Tesic, L. Xie, R. Yan, and J. Yang. IBM Research TRECVID-2007 Video Retrieval System. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.
8. C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
9. O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-supervised Learning*. MIT Press, Cambridge, MA, 2006.
10. M. Chen, M. Christel, A. Hauptmann, and H. Wactlar. Putting Active Learning into Multimedia Applications: Dynamic Definition and Refinement of Concept Classifiers. In *Proc. Int. Conf. on Multimedia*, pages 902–911, November 2005.
11. A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
12. R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
13. M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, October 2008.
14. R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. *Computer Vision*, 2:1816–1823, 2005.
15. U. Gargi and J. Yagnik. Solving the Label Resolution Problem in Supervised Video Content Classification. In *Proc. Int. Conf. on Multimedia Retrieval*, pages 276–282, October 2008.
16. Z. Gu, T. Mei, X.-S. Hua, J. Tang, and X. Wu. Multi-layer Multi-instance Kernel for Video Concept Detection. In *Proc. Int. Conf. on Multimedia*, pages 349–352, September 2007.
17. A. Hauptmann, R. Yan, and W. Lin. How many High-Level Concepts will Fill the Semantic Gap in News Video Retrieval? In *Proc. Int. Conf. Image and Video Retrieval*, pages 627–634, July 2007.
18. T. Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.
19. J. Yuan et al. THU and ICRC at TRECVID 2007. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.
20. T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *Int. Conf. Machine Learning*, pages 200–209, June 1999.
21. L. Kennedy, S.-F. Chang, and I. Kozintsev. To Search or to Label? Predicting the Performance of Search-based Automatic Image Classifiers. In *Int. Workshop Multimedia Information Retrieval*, pages 249–258, October 2006.
22. W. Kraaij and P. Over. TRECVID-2007 High-Level Feature Task: Overview. In *Proc. TRECVID Workshop (slides available from: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html)*, November 2007.
23. D. Lewis and W. Gale. A Sequential Algorithm for Training Text Classifiers. In *Proc. Int. Conf. Research and Development in Information Retrieval*, pages 3–12, July 1994.
24. L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic Object Picture collecTion via Incremental MOdel Learning. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 57–64, June 2007.

25. D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Int. Conf. Computer Vision*, pages 1150–1157, September 1999.
26. N. Morsillo, C. Pal, and R. Nelson. Semi-supervised Visual Scene and Object Analysis from Web Images and Text. In *Scene Understanding Symposium*, February 2008.
27. M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-Scale Concept Ontology for Multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.
28. R. Paredes and A. Perez-Cortes. Local Representations and a Direct Voting Scheme for Face Recognition. In *Proc. Workshop on Pattern Rec. and Inf. Systems*, pages 71–79, July 2001.
29. G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.
30. B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
31. F. Schroff, A. Criminisi, and A. Zisserman. Harvesting Image Databases from the Web. In *Proc. Int. Conf. Computer Vision*, pages 1–8, October 2007.
32. B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
33. J. Sivic and A. Zisserman. Video Google: Efficient Visual Search of Videos. In *Toward Category-Level Object Recognition*, pages 127–144. Springer-Verlag New York, Inc., 2006.
34. A. Smeaton. Techniques Used and Open Challenges to the Analysis, Indexing and Retrieval of Digital Video. *Inf. Syst.*, 32(4):545–559, 2007.
35. A. Smeaton, P. Over, and W. Kraaij. Evaluation Campaigns and TRECVid. In *Int. Workshop Multimedia Information Retrieval*, pages 321–330, October 2006.
36. A. Smeulders, M. Worring, S. Santini, and A. Gupta R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
37. C. Snoek and M. Worring. Concept-based Video Retrieval. *Foundations and Trends in Information Retrieval*, 4(2):215–322, 2009.
38. C. Snoek, M. Worring, O. de Rooij, K. van de Sande, R. Yan, and A. Hauptmann. VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems. *IEEE MultiMedia*, 15(1):86–91, 2008.
39. C. Snoek, M. Worring, B. Huurnink, J. van Gemert, K. van de Sande, D. Koelma, and O. de Rooij. MediaMill: Video Search using a Thesaurus of 500 Machine Learned Concepts. In *1st Int. Conf. Sem. Dig. Media Techn. (Posters and Demos)*, 2006.
40. Y. Sun, S. Shimada, Y. Taniguchi, and A. Kojima. A Novel Region-based Approach to Visual Concept Modeling using Web Images. In *Int. Conf. Multimedia*, pages 635–638, October 2008.
41. S. Tong and E. Chang. Support Vector Machine Active Learning for Image Retrieval. In *Proc. Int. Conf. on Multimedia*, pages 107–118, September 2001.
42. B. Turlach. Bandwidth Selection in Kernel Density Estimation: A Review. In *CORE and Institut de Statistique*, pages 23–49, 1993.
43. A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Identifying Relevant Frames in Weakly Labeled Videos for Training Concept Detectors. In *Proc. Int. Conf. Image and Video Retrieval*, pages 9–16, July 2008.
44. YouTube Serves up 100 Million Videos a Day Online. in USA Today (Garnnett Company, Inc.); available from http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm (retrieved: Sep'08), July 2006.
45. K. van de Sande, T. Gevers, and C. Snoek. A Comparison of Color Features for Visual Concept Classification. In *Proc. Int. Conf. Image and Video Retrieval*, pages 141–150, July 2008.
46. D. Wang, X. Liu, L. Luo, J. Li, and B. Zhang. Video Diver: Generic Video Indexing with Diverse Features. In *Proc. Int. Workshop Multimedia Information Retrieval*, pages 61–70, September 2007.
47. M. Wang, X.-S. Hua, Y. Song, X. Yuan, S. Li, and H.-J. Zhang. Automatic Video Annotation by Semi-supervised Learning with Kernel Density Estimation. In *Proc. Int. Conf. on Multimedia*, pages 967–976, October 2006.

48. K. Wnuk and S. Soatto. Filtering Internet Image Search Results Towards Keyword Based Category Recognition. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
49. A. Yanagawa, S.-F. Chang, L. Kennedy, and W. Hsu. Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts. Technical report, Columbia University, 2007.
50. K. Yanai and K. Barnard. Probabilistic Web Image Gathering. In *Int. Workshop on Multimedia Inf. Retrieval*, pages 57–64, November 2005.
51. J. Yang and A. Hauptmann. (Un)Reliability of Video Concept Detection. In *Proc. Int. Conf. Image and Video Retrieval*, pages 85–94, July 2008.
52. X. Zhu. Semi-supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.