VideoCut: Removing Irrelevant Frames by Discovering the Object of Interest

David Liu[1], Gang Hua[2], and Tsuhan Chen[1]

Dept. of ECE, Carnegie Mellon University[1] and Microsoft Live Labs[2]
dliu@cmu.edu, ganghua@microsoft.com, tsuhan@cmu.edu

Abstract. We propose a novel method for removing irrelevant frames from a video given user-provided frame-level labeling for a very small number of frames. We first hypothesize a number of candidate areas which possibly contain the object of interest, and then figure out which area(s) truly contain the object of interest. Our method enjoys several favorable properties. First, compared to approaches where a single descriptor is used to describe a whole frame, each area's feature descriptor has the chance of genuinely describing the object of interest, hence it is less affected by background clutter. Second, by considering the temporal continuity of a video instead of treating the frames as independent, we can hypothesize the location of the candidate areas more accurately. Third, by infusing prior knowledge into the topic-motion model, we can precisely follow the trajectory of the object of interest. This allows us to largely reduce the number of candidate areas and hence reduce the chance of overfitting the data during learning. We demonstrate the effectiveness of the method by comparing it to several other semi-supervised learning approaches on challenging video clips.

1 Introduction

The endless streams of videos on the Internet often contain irrelevant data. Our goal is to cut video clips shorter and retain the frames that are relevant to the user input. We assume the user has an "*object of interest*" (OOI) in mind, which can, for example, be a car, a book, or the scene of a forest. The system will infer which frames contain the OOI. This application can be used, e.g., for shortening surveillance videos or TV programs.

We consider the case where the system is provided with very limited information. Specifically, the user will label at least one frame as relevant and another frame as irrelevant. These labels are at the frame-level instead of at the pixel-level. Although pixel-level labeling (such as using a bounding box or segmentation mask to specify the location of the OOI) can provide more information, we intend to explore the possibility of letting the user provide coarser and less tedious labeling.

We formulate the task as a self-training multiple instance learning problem. For each frame, we postulate a number of candidate areas, and use a multiple instance learning algorithm to simultaneously find out whether the OOI exists

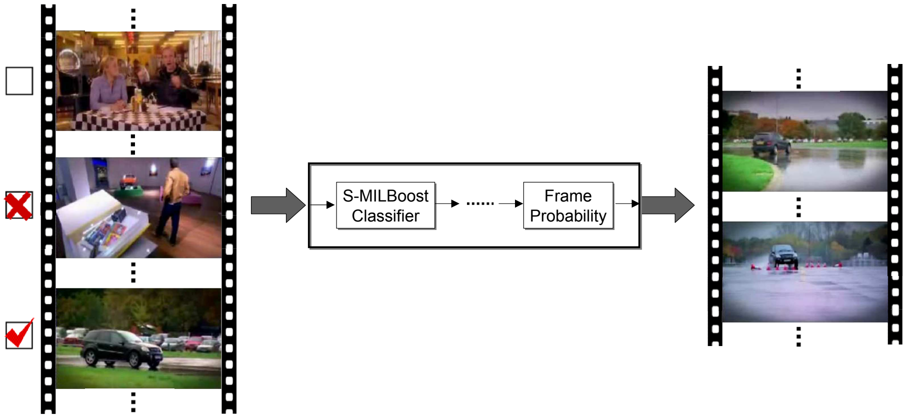


Fig. 1. Frames are unlabeled (top left), labeled as irrelevant (middle left) or relevant (bottom left). The system will find out what the object of interest is (in this case, the black vehicle) and remove frames that don't contain the vehicle.

in the frame, and if it does, where it is located. The reason that we go one step beyond our goal (that is, trying to locate the OOI) is because we are able to exploit the temporal smoothness property of video objects to consolidate their locations. That is to say, objects tend to move in a continuous manner from frame to frame.

We use sporadically labeled frames to train a multiple instance learning algorithm called MILBoost [22]. It was originally applied to a face detection problem. In their work, images are manually labeled by drawing a rectangle around the head of a person. In our system, we only have frame-level labels, i.e., no rectangles are available.

Our semi-supervised framework can be distinguished from prior work in several aspects. Our work does not require pixel-level labeled data. In[17], learning requires both pixel-level labeled data and frame-level labeled data. An object detector is initially trained on the pixel-level labeled data, and the learned model is used to estimate labels for the frame-level labeled data. As illustrated in Fig. 1, we "**discover**" the OOI since no bounding box is given, which also distinguishes our work with the video object retrieval work in [20][19], where the OOI is explicitly labeled at the pixel-level.

Image retrieval systems often allow users to provide positive and negative feedback, hence the task of image retrieval can also be cast under the self-training [14] or multiple instance learning [22] framework. Nonetheless, our system exploits temporal information of videos in a novel way, which distinguishes itself from the image retrieval literature. In [16], activities in a video are condensed into a shorter period by simultaneously showing multiple activities. It does not intend to **discover** the frames that contain the user-desired OOI from limited user input.

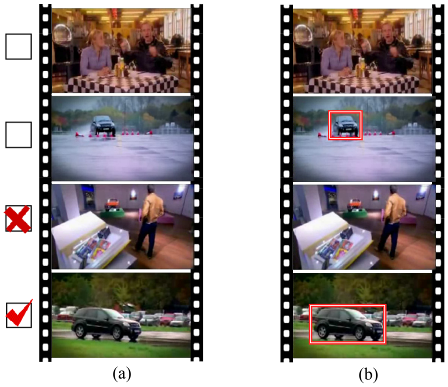


Fig. 2. (a) Labeling at the frame level assumed in this work. Frames can be unlabeled, or labeled as positive or negative. (b) The bounding box type of labeling provides more explicit information regarding the object of interest, but is also more tedious in the labeling process.

Our method is based on the bag-of-words representation, which is part-based. Different than other part-based methods such as the one-shot learning framework [7], we leverage motion consistency to improve recognition, while the one-shot learning framework did not utilize that. We leverage the unsupervised topic-motion model in [11] and extend it to a semi-supervised setting by incorporating additional prior models during learning. The problem solved, the application targeted, as well as the fundamental approach adopted in our paper, are significantly different from [11].

Our contribution can hence be summarized as follows: **1)** A novel application that summarizes videos based on the implicitly specified OOI. **2)** A novel system that uses weakly labeled data for object discovery in video. **3)** A novel method that takes advantage of the temporal smoothness property during semi-supervised learning.

The paper is organized as follows. In section 2 we define the type of user labeling information that is available to the system. In section 3 we introduce a baseline method, where features at the frame-level are used for semi-supervised learning. In section 4 we explain in detail the proposed method. In section 5 we will compare the proposed method with the baseline method and several variants of the proposed method. Finally, we conclude in section 6.

2 Frame-level labels

The amount of user label information as well as its format has a major impact on system design. The amount of user label information can range from all frames being labeled to none. For those frames being labeled, the labeling can be as detailed as providing bounding boxes for each frame (which we call pixel-level labeling), or as coarse as "this frame does (or does not) contain the OOI" (which we call frame-level labeling).

In this paper, we consider the more challenging task of having as input only frame-level labeling; see Fig. 2 for a comparison. This kind of 'weak labeling' is very different from traditional object detection; see for example [18], where the characteristics of the OOI are learned from plenty of pixel-level labeled data. This is also different from the recent video retrieval work in [20][19]. Traditional object detection not only involves a lot of human labor for labeling the images by putting bounding boxes on the OOI, but also has the difficulty of scaling to multiple categories of objects. Since the OOI in a sequence can be of any category, it is very difficult to train a comprehensive object detector that covers all types of objects.

3 Semi-supervised learning at frame-level

Our first attempt to achieve the goal of VideoCut is to use semi-supervised learning at the frame-level. Each frame is represented as a histogram of *visual words*, or *textons* [9]. To generate visual words, we use the Maximally Stable Extremal Regions (MSER) operator [8] to find salient patches [1]. MSERs are the parts of an image where local contrast is high. Other operators could also be used; see [2] for a collection. Features are extracted from these MSERs by Scale Invariant Feature Transform (SIFT) [12]. In this work we extract MSERs and SIFT descriptors from grayscale images. Patches and features extracted from color images [21] can also be used instead. The SIFT features from a video are vector quantized using K-Means Clustering. The resulting $J = 50$ cluster centers form the dictionary of visual words, $\{w_1, ..., w_J\}$. Each MSER can then be represented by its closest visual word.

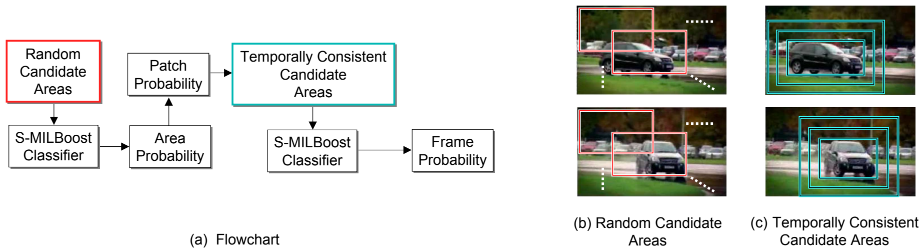
The histograms of the labeled frames along with their labels are fed to the system to train a classifier. The classifier is then applied to the unlabeled frames. Frames with high confidence scores are assigned pseudo-labels. The pseudo-labeled data is combined with the original labeled data and the classifier is trained again. The classifier we use is Discrete AdaBoost [4]. We will use this method as a baseline method in the experiments. This kind of self-training [14] procedure has been used extensively in different domains [10][17] and achieved top results in the NIPS competition [4].

4 Semi-supervised learning at sub-frame level

There are two issues with the frame-level learning framework in Sec. 3.

1. The OOI can be small and the visual words from the whole frame are usually dominated by background clutter. Hence the full-frame histogram representation is not a truthful representation of the OOI.

[1] The word 'region' should not be confused with the 'candidate areas' to be introduced later. Each candidate area contains a set of MSER patches.



(a) Flowchart

(b) Random Candidate Areas

(c) Temporally Consistent Candidate Areas

Fig. 3. Semi-supervised learning at sub-frame level using temporally consistent candidate areas.

2. Objects in video often follow a smooth trajectory, which we call the *temporal smoothness property*. With frame-level learning, the temporal smoothness property cannot be readily exploited.

We address these issues by learning at a sub-frame level. Fig. 3(a) shows the proposed system flowchart. In each frame, we propose a number of *Random Candidate Areas* that potentially contain the OOI (illustrated in Fig. 3(b)). This will be detailed in section 4.1. The candidate areas are passed to a self-training version of MILBoost (S-MILBoost) and assigned an *Area Probability*, a score that tells us how likely this candidate area truly belongs to the OOI. This will be detailed in section 4.2. After each candidate area receives a score, we assign each image patch (MSER) a *Patch Probability*, which is defined as the largest *Area Probability* among the candidate areas that cover that image patch. Given the *Patch Probability*, in section 4.3 we will explain how to obtain the *Temporally Consistent Candidate Areas*. Basically, this is achieved by fitting a model which simultaneously *discovers* the OOI and *tracks* it across frames. The *Temporally Consistent Candidate Areas* are illustrated in Fig. 3(c); using them, we train S-MILBoost once again. As we will show in the experiments, this new S-MILBoost classifier will be more reliable than the previous one trained with the *Random Candidate Areas*. Finally, the S-MILBoost classifier gives us the *Frame Probability*, which tells us how likely each frame contains the OOI. Using the *Frame Probability*, we can determine the irrelevant frames and perform VideoCut.

Notice how the two issues mentioned earlier are resolved by using this proposed flowchart. First, the candidate areas are smaller than the whole frame and hence include less background clutter, which address the first issue mentioned above. Second, the candidate areas in one frame can be temporally correlated with the candidate areas in the next frame by performing 'weak' object tracking (illustrated in Fig. 3(c)), which addresses the second issue. We emphasize that this 'weak' tracking is different from traditional object tracking, as we will explain later.

In the experiments section we will compare our proposed flowchart with some other methods, which replace or omit some parts of the modules in Fig. 3(a). In the following subsections we will explain the details and merits of each component in Fig. 3(a).

4.1 Random candidate areas

Since the user labeling does not tell us where the OOI is located (neither in the labeled nor in the unlabeled frames), we need to set the candidate areas based on prior knowledge, if any. At the beginning, we use candidate areas with fixed size and uniform spacing and call them the random candidate areas. Each candidate area is represented as a histogram of visual words, as shown in Fig. 4. After we have a rough guess (using the techniques in the next two subsections), we will refine the candidate areas by placing them more densely around the estimated location of the OOI. We call these later candidate areas as temporally consistent candidate areas. See Fig. 3(b)(c) for illustrations.

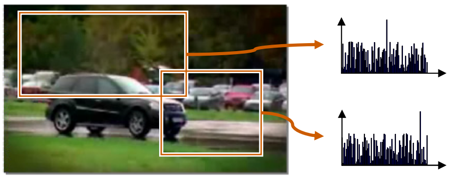


Fig. 4. Candidate areas, each represented by a histogram over visual words. In the experiments, we use a variety of different densities and spacings of candidate areas.

4.2 Self-training MILBoost

Using a similar self-training procedure as in Sec. 3, we first use the labeled frames to train a multiple instance learning [22] classifier. As a result, each candidate area of the labeled frames is assigned an area probability, which is the probability that an area contains the OOI. The classifier is then self-trained with the unlabeled frames and pseudo-labels included. As a result, the area probabilities of candidate areas in unlabeled frames are obtained as well.

Different than in Sec. 3, we have multiple histograms per frame, instead of a single one, therefore we use a multiple instance learning classifier, MILBoost [22]. First let us define some notations. We denote the histogram over visual words of a candidate area as $x_{k,a}$, where k indices over frames and a indices over the candidate areas inside frame k. Let $t_k \in \{0,1\}$ denote the label or pseudo-label of frame k. Each frame has a *frame probability* p_k, and each candidate area has an *area probability* $p_{k,a}$. The *frame probability* is the probability that a frame contains the OOI, and the *area probability* is the probability that the area contains the OOI. Since a frame is labeled as positive as long as it contains the OOI, it is natural to model the relationship between p_k and $p_{k,a}$ using the Noisy-OR model [15], $p_k = 1 - \prod_{a \in k}(1 - p_{k,a})$. The likelihood is given by $L(C) = \prod_k {p_k}^{t_k}(1-p_k)^{(1-t_k)}$.

As implied by its name, MILBoost produces a strong classifer $C(x_{k,a})$ in the form of a weighted sum of weak classifiers: $C(x_{k,a}) = \sum_u \lambda_u c_u(x_{k,a})$, $c_u(x_{k,a}) \in$

$\{-1, +1\}$. The strong classifier score $C(x_{k,a})$ translates into the area probability, $p_{k,a}$, by the logistic sigmoid function $p_{k,a} = 1/(1 + exp(-C(x_{k,a})))$. Using the AnyBoost [13] method, the boosting weight $\varpi_{k,a}$ of each candidate area is the derivative of the log-likelihood, easily to be shown as $\frac{t_k - p_k}{p_k} p_{k,a}$. In round u of boosting, one first solves the optimization problem $c_u(.) = \arg\max_{c'(.)} \sum_{k,a} c'(x_{k,a})\varpi_{k,a}$. A line search is then performed to seek for the optimal parameter λ_u, i.e., $\lambda_u = \arg\max_{\lambda} L(C + \lambda c_u)$.

In summary, S-MILBoost produces a classifier that assigns each frame a frame probability, and each candidate area an area probability. Notice that the S-MILBoost classifier is always used in a learning mode, during which the area and frame probabilities are estimated.

4.3 Temporally consistent candidate areas

The accuracy of the frame probabilities depends heavily on the placing of the candidate areas; as an extreme example, if the OOI appears in a frame but none of the candidate areas cover it, then there would be no chance we could have correctly estimated the frame probability. This suggests a refinement of the placing scheme of candidate areas based on extra information. Notice that, we haven't yet exploited the temporal smoothness property of videos.

We would like to use the temporal smoothness property to refine the placing of the candidate areas. The temporal smoothness property is typically exploited through tracking the object. However, tracking requires manual initialization of the object location and size, information which is not available to us.

The topic-motion model [11] simultaneously estimates the appearance and location of the OOI. However, it was used in an unsupervised setting where one has no prior knowledge about the label (object vs. background) of each image patch. In our case, the area probabilities estimated by S-MILBoost provides information that we could use as prior knowledge.

The topic-motion model was designed for the case where at most one OOI appears in each frame. But this is not a problem for our system, because as long as one of the possibly many OOIs is discovered, the frame probability will be high. In other words, we don't need to identify every OOI to decide if a frame is relevant or irrelevant. Also notice that discovering the OOI is not our ultimate goal.

Denote frame k as d_k, where k indices over all frames. Each patch in d_k is associated with a visual word w, a position $\mathbf{r}$, and a hidden variable $z \in \{z_+, z_-\}$. Define $p(z_+|d_k)$ as the probability of a patch being originated from the OOI in frame k, and likewise $p(z_-|d_k)$ for the background. We define a spatial distribution $p(\mathbf{r}|z_+, d_k)$ that models the location of the patches originated from the OOI. We assume $p(\mathbf{r}|z_+, d_k)$ follows a Gaussian distribution, but other distributions (such as a mixture of Gaussians) could be used as well. Likewise, $p(\mathbf{r}|z_-, d_k)$ models the location of patches originated from background and we assume it follows a uniform distribution. The third distribution is $p(w|z_+)$, which models the appearance of the OOI. It is the normalized histogram over visual words corresponding to patches originated from the OOI. Likewise, $p(w|z_-)$ models

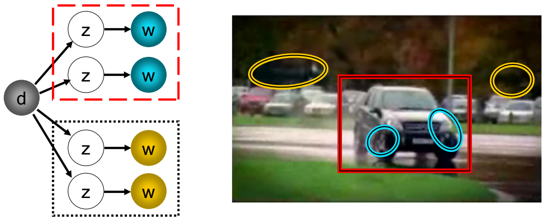


Fig. 5. Graphical model representation. Dashed lines are not the typical plate representation.

the appearance of the background. We assume that the joint distribution of word w, position $\mathbf{r}$, and hidden label z of a patch in frame d_k is modeled as $p(z, \mathbf{r}, w|d_k) \equiv p(z|d_k)p(\mathbf{r}|z, d_k)p(w|z)$.

Define the state $\mathbf{s}(k)$ as the unknown position and velocity of the OOI in frame d_k. We assume a constant velocity motion model and the state evolves according to $\mathbf{s}(k+1) = \mathbf{F}\mathbf{s}(k) + \boldsymbol{\xi}(k)$, where $\mathbf{F}$ is the state matrix and the process noise sequence $\boldsymbol{\xi}(k)$ is white Gaussian. Suppose at time k there are a number of m_k patches. If a patch is originated from the OOI, then its position can be expressed as $\mathbf{r}_i(k) = \mathbf{H}\mathbf{s}(k) + \boldsymbol{\zeta}_i(k)$, where $\mathbf{H}$ is the output matrix and the observation noise sequence $\boldsymbol{\zeta}_i(k)$ is white Gaussian; otherwise, the position is modeled as a uniform spatial distribution. The state estimate can be written as $\hat{\mathbf{s}}(k) = \sum_{i=1}^{m_k} \hat{\mathbf{s}}_i(k)\beta_i(k)$, where $\hat{\mathbf{s}}_i(k) = \hat{\mathbf{s}}(k^-) + \mathbf{W}(k)\boldsymbol{\epsilon}_i(k)$ is the updated state estimate conditioned on the event that $\mathbf{r}_i(k)$ is originated from the OOI, where $\boldsymbol{\epsilon}_i(k) = \mathbf{r}_i(k) - \hat{\mathbf{r}}(k^-)$ is the innovation, $\hat{\mathbf{r}}(k^-)$ is the observation prediction, $\hat{\mathbf{s}}(k^-)$ is the state prediction, and $\mathbf{W}(k)$ is the Kalman Filter gain [3]. The state estimation equations are essentially the same as in the PDA filter [3]. The association probability $\beta_i(k)$ is defined as $\beta_i(k) \propto N(\boldsymbol{\epsilon}_i(k)|0, \boldsymbol{\Upsilon}(k))p(z_i(k)|w_j, \mathbf{r}_i(k), d_k)$, where the first term contains motion information, the second term contains appearance and location information, and $\boldsymbol{\Upsilon}(k)$ is the innovation covariance.

Parameter estimation The distributions $P(w|z)$, $P(z|d)$, and $P(\mathbf{r}|z, d)$ are estimated using the Expectation-Maximization (EM) algorithm [6], which maximizes the log-likelihood $\mathcal{R} = \sum_k \sum_j \sum_i n_{kji} \log p(d_k, w_j, \mathbf{r}_i(k))$, where $n_{kji} \equiv n(d_k, w_j, \mathbf{r}_i(k))$ is a count of how many times a patch in d_k at position $\mathbf{r}_i(k)$ has appearance w_j. The EM algorithm consists of two steps. The E-step computes the posterior probabilities for the hidden variables:

$$p(z_l|d_k, w_j, \mathbf{r}_i(k)) = \frac{p(z_l|d_k)p(w_j|z_l)p(\mathbf{r}_i(k)|z_l, d_k)}{\sum_R p(z_l|d_k)p(w_j|z_l)p(\mathbf{r}_i(k)|z_l, d_k)} \quad (1)$$

The M-step maximizes the expected complete data likelihood. We adopt a Bayesian approach to estimating the probabilities, using m-probability-estimation [5]. First, notice that the *area probability*, $p_{k,a}$, computed from S-MILBoost contains prior knowledge about the OOI. This prior knowledge should be incorporated into the detection of temporally consistent candidate areas. This is a significant improvement over the algorithm in [11], which was completely unsupervised.

Noticing that each patch can belong to multiple candidate areas, we define the *patch probability* as the largest *area probability* among the candidate areas that cover an image patch. The *patch probability* is written as $p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k))$, with the subscript "MIL" emphasizing that this probability is estimated from the outcome of S-MILBoost. A simplified graphical model is illustrated in Fig. 5, where the variable $\mathbf{r}$ is omitted to simplify illustration. Dashed lines indicate groups of image patches having the same value of p_{MIL}. More specifically, dashed lines in red correspond to the red box (candidate area) in the picture, and blue (yellow) nodes in the graphical model correspond to blue (yellow) ellipses in the picture. We then obtain:

$$p(z_l|d_k) = \frac{\sum_{j,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{j,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))}{\sum_{l,j,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{l,j,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))} \quad (2)$$

$$p(w_j|z_l) = \frac{\sum_{k,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{k,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))}{\sum_{j,k,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{j,k,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))} \quad (3)$$

$$p(\mathbf{r}_i(k)|z_+, d_k) = \mathcal{N}(\mathbf{r}_i(k)|\hat{\mathbf{r}}(k), \boldsymbol{\Sigma}_{d_k}) \quad (4)$$

where $z_l \in \{z_+, z_-\}$ is the value taken by $z_i(k)$ and $\hat{\mathbf{r}}(k) = \mathbf{H}\hat{\mathbf{s}}(\mathbf{k})$ is the position estimate. The covariance $\boldsymbol{\Sigma}_{d_k}$ in the Normal distribution in Eq.(4) is the weighted covariance matrix of the observations $\mathbf{r}_i(k)$. The weighted covariance matrix is the covariance matrix with a weighted mass for each data point, with weights equal to the association probabilities $\beta_i(k)$. As a result, if the association probabilities have high uncertainty, the spatial distribution $p(\mathbf{r}|z_+, d)$ will be flatter; if low uncertainty, it will be sharper around the position of the OOI.

Finally, we propose a number of temporally consistent candidate areas that have $\hat{\mathbf{r}}(k)$ as center and with various sizes, as shown in Fig. 3(c). We use a 1.2 scale ratio between two areas, with the smallest one equal to the variance specified by $\boldsymbol{\Sigma}_{d_k}$ in Eq.(4), and with no more than 5 areas in total. Using various sizes is to increase system robustness in case of inaccurate size estimates.

5 Experiments

We use 15 video clips from YouTube.com and TRECVID [1]. Sample frames are shown in Fig. 6. Most of the clips are commercial advertisements with a well defined OOI and range from 20 to 356 seconds in length. We sample each video at two frames per second. In total, there are 3128 frames of size 320 × 240. The frames have visible compression artifacts.

The video frames are ground-truthed as positive or negative according to whether they contain the OOI; e.g., in a PEPSI commercial, we assume the PEPSI logo is the OOI. Each video clip is run twenty runs, where in each run we randomly select N_p frames from the positive frames and N_n frames from

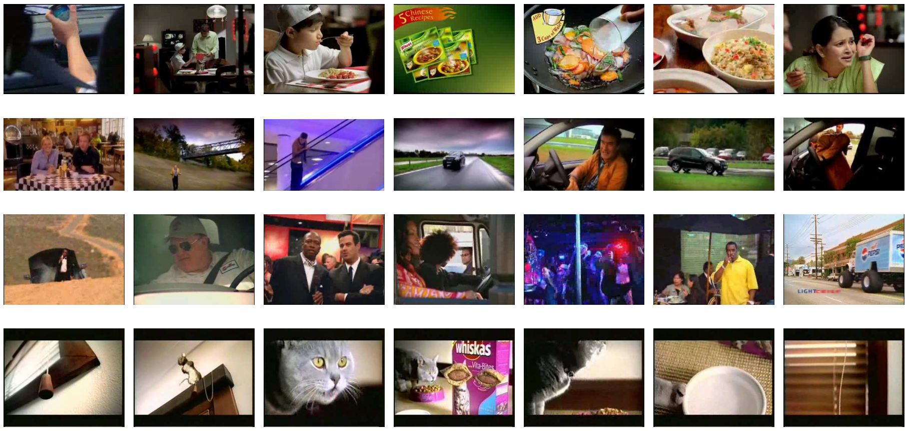


Fig. 6. Sample frames. Name of video clip, from top to bottom: Knorr, Benz, Pepsi, Whiskas.

the negative frames as labeled data, where N_p and N_n are one or three. The rest of the frames are treated as unlabeled data. Results are averaged over the twenty runs. Notice that the labeled frames are labeled at the frame-level but not pixel-level.

Table 1 shows the average precision (area under precision-recall curve) of different methods. In the following, we will introduce the different comparative methods listed in Table 1 while we discuss the results. In general, we have the following observations:

Method 1: Supervised learning using only labeled data is consistently outperformed by the semi-supervised variants. When the number of labeled frames is low, its performance is close to by chance.

Method 2: Semi-supervised learning at frame level performs only marginally better than supervised learning when the number of labeled frames is as low as $(1+, 1-)$, but improves significantly as the number of labeled frames increases.

Method 3: Semi-supervised learning at sub-frame level with random areas consistently outperforms semi-supervised learning at the frame level. This justifies our claim in Sec. 4 that frame-level learning can be hindered when background clutter dominates the appearance features. Using sub-frames (candidate areas) helps the learning process to focus on the features originated from the OOI. The candidate areas consist of rectangles of size 160 × 120 with equal spacing between each other. In addition, a rectangle of size 320 × 240 covering the whole frame is used in here, in Method 4, and in the proposed method, in order to take care of large objects and inaccurate size estimates. After training S-MILBoost, we did not refine the placing of candidate areas, as we do in Method 4 and in the proposed method.

We experimented with different numbers of rectangles by changing the spacing between them and obtained different performances as shown in Fig. 7. There

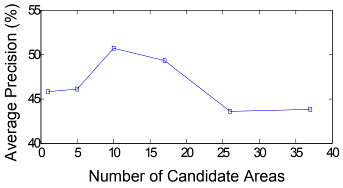


Fig. 7. Increasing the number of areas does not lead to increase in performance.

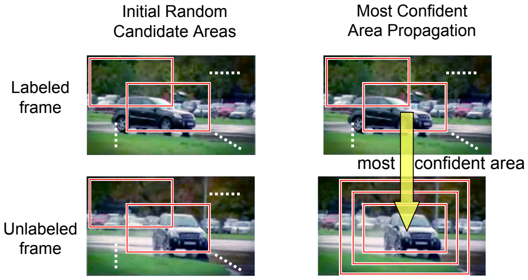

Initial Random Candidate Areas
Most Confident Area Propagation
Labeled frame
Unlabeled frame
most confident area

Fig. 8. Illustration of Method 4.

is a sweet spot at the number of 10 areas, which shows that the more candidate areas does not necessarily yield better performance. Even though increasing the number of areas will increase the chance that one of the candidate areas faithfully represents the OOI, the chance of overfitting also increases, hence the drop in performance. We also experimented with placing the areas more concentrated around the center of the frame but obtained similar results.

Method 4: Most confident area propagation: This method is the closest to the proposed method. Instead of using 'weak' tracking, we assume the OOI is stationary within a shot. As illustrated in Fig. 8, each unlabeled frame obtains its 'base' candidate area by replicating, from the nearest labeled frame, the size and position of the most confident area. Nearness can be defined as the visual similarity between frames or as the time difference between frames. We found the latter to work better. The base area is then resized and replicated within the frame using a 1.2 scale ratio between two areas, with the smallest one equal to the size of the base area, and no more than 5 areas in total. Since videos often contain multiple scene transitions or shots, we only allow the replication to happen within a shot and not across shots. If there are no labeled frames within a shot, we place random candidate areas in that shot.

In summary, the proposed method outperforms all the other methods (Table 1). Together with Fig. 7, this justifies our earlier expectation that properly placed candidate areas are crucial to the performance; using a huge number of candidate areas overfits the data and lowers the performance. The temporally consistent candidate areas reduce the need for a large number of uninformative candidate areas. Finally, in Fig. 9, we display some frames that are inferred by the proposed method.

6 Conclusion and future work

We have presented an approach for removing irrelevant frames in a video by discovering the object of interest. Through extensive experiments, we have shown

Sequence	Label	By Chance	Method 1 Supervised	Semi-Supervised			
				Method 2 Frame Level	Sub-Frame Level		
					Method 3	Method 4	Proposed
Benz	1+,1-	32.6	26.0	28.9	31.7	29.3	38.7
	3+,3-	32.5	29.1	52.9	54.6	48.8	58.3
Pepsi	1+,1-	34.1	32.6	34.3	41.9	39.1	42.3
	3+,3-	33.7	39.4	53.9	57.7	50.6	63.2
Whiskas	1+,1-	43.7	49.0	54.2	62.6	64.1	65.3
	3+,3-	43.5	53.9	71.2	77.0	78.1	73.8
SkittlesFunny	1+,1-	3.9	2.8	5.2	10.7	10.7	6.5
	3+,3-	2.0	4.1	11.3	21.2	22.5	22.7
CleanClear	1+,1-	21.2	14.4	15.5	45.4	41.8	36.1
	3+,3-	19.4	21.1	41.9	51.4	57.6	62.2
CatFood	1+,1-	39.0	40.5	41.7	62.7	65.1	66.9
	3+,3-	38.2	58.2	76.0	91.4	91.4	91.4
E-Aji	1+,1-	27.1	26.5	27.0	31.4	29.9	36.0
	3+,3-	25.5	23.8	32.6	42.7	34.7	36.2
CaramelNut	1+,1-	25.9	39.3	53.7	67.6	58.9	58.9
	3+,3-	24.1	58.4	67.5	67.6	70.2	70.2
Knorr	1+,1-	20.7	20.4	32.2	44.2	62.1	59.4
	3+,3-	18.5	20.2	48.9	57.2	69.4	67.7
Kellogs	1+,1-	18.4	19.8	20.0	26.4	30.3	30.3
	3+,3-	14.7	18.6	22.1	25.3	36.4	38.0
FlightSimul	1+,1-	10.8	15.4	43.5	42.6	53.5	59.6
	3+,3-	10.5	18.7	50.7	44.4	40.8	62.1
SpaceShuttle	1+,1-	4.8	2.8	2.8	3.5	3.7	4.2
	3+,3-	4.2	3.6	12.7	27.7	27.3	25.1
WeightAero	1+,1-	11.6	8.5	38.1	27.8	33.9	44.9
	3+,3-	11.2	46.9	56.3	40.9	48.5	56.1
WindTunnel	1+,1-	24.1	14.7	15.0	36.1	33.8	35.2
	3+,3-	23.8	41.6	47.2	56.9	56.7	56.1
Horizon	1+,1-	11.2	15.8	18.4	22.6	28.3	34.1
	3+,3-	10.5	18.0	41.3	44.1	48.9	54.6
Average	1+,1-	**21.9**	**21.9**	**28.7**	**37.1**	**39.0**	**41.2**
	3+,3-	**20.8**	**30.4**	**45.8**	**50.7**	**52.1**	**55.8**

Table 1. Comparing the average precision (%). The number of labeled frames are one positive (1+) and one negative (1−) in the upper row, and three positives and three negatives in the lower row for each video sequence.

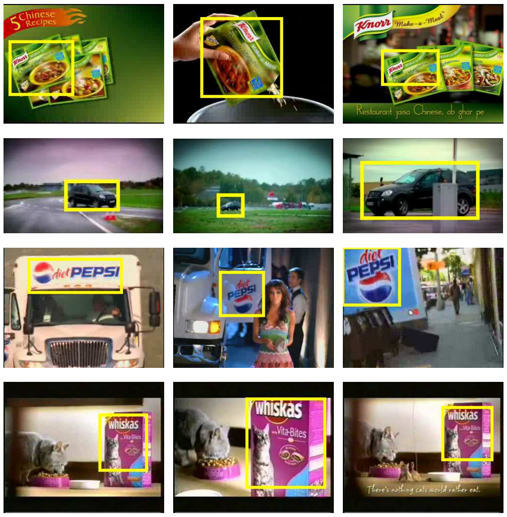

5 Chinese Recipes
Knorr
Make-a-Meal
Restaurant jaisa Chinese, ab ghar pe
diet PEPSI
whiskas
Vita-Bites
There's nothing cats would rather eat.

Fig. 9. Sample frames that are inferred as positive. A yellow box shows the candidate area with highest area probability. Name of video clip, from top to bottom: Knorr, Benz, Pepsi, Whiskas.

that this is not easily achieved by directly applying supervised or semi-supervised learning methods in the literature developed for still images.

On a higher level, our method can be considered as a tracking system but without manual track initialization; the system finds out itself what the "best track" is, with the objective of agreeing with the user's labeling on which frames contain the object of interest.

References

1. *http://www-nlpir.nist.gov/projects/trecvid/.*
2. *http://www.robots.ox.ac.uk/∼vgg/research/affine/.*
3. Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association.* Academic Press, 1988.
4. K. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. *Intl. Conf. Knowledge Discovery and Data Mining*, 2002.
5. B. Cestnik. Estimating probabilities: A crucial task in machine learning. *Proc. European Conf. Artificial Intelligence*, pages 147–149, 1990.
6. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society*, 39:1–38, 1977.
7. L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. PAMI*, 28 (4):594–611, 2006.
8. J.Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, 2002.
9. B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.

10. Y. Li, H. Li, C. Guan, and Z. Chin. A self-training semi-supervised support vector machine algorithm and its applications in brain computer interface. *IEEE Intl. Conf. Acoustics, Speech, and Signal Processing*, 2007.
11. D. Liu and T. Chen. A topic-motion model for unsupervised video object discovery. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
12. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Computer Vision*, 60:91–110, 2004.
13. L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1999.
14. K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. *Intl. Conf. Information and Knowledge Management*, 2000.
15. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc, 1988.
16. Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. *IEEE Intl. Conf. Computer Vision*, 2007.
17. C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. *IEEE Workshop on Applications of Computer Vision*, 2005.
18. H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Intl. J. Computer Vision*, 56:151–177, 2004.
19. J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *Intl. Journal of Computer Vision*, 67:189–210, 2006.
20. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE Intl. Conf. Computer Vision*, 2003.
21. J. van de Weijer and C. Schmid. Coloring local feature extraction. *Proc. European Conf. Computer Vision*, 2006.
22. P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2005.