Weakly Supervised Object Localization and Segmentation in Videos

Mrigank Rochan[a,*], Shafin Rahman[a], Neil D. B. Bruce[a], Yang Wang[a]

[a]*Department of Computer Science, University of Manitoba, Winnipeg, MB R3T 2N2, Canada*

Abstract

We consider the problem of localizing and segmenting objects in weakly labeled video. A video is weakly labeled if it is associated with a tag (e.g. YouTube videos with tags) describing the main object present in the video. It is weakly labeled because the tag only indicates the presence/absence of the object, but does not give the detailed spatial/temporal location of the object in the video. Given a weakly labeled video, our method can automatically localize the object in each frame and segment it from the background. Our method is fully automatic and does not require any user-input. In principle, it can be applied to a video of any object class. We evaluate our proposed method on a dataset with more than 100 video shots. Our experimental results show that our method outperforms other baseline approaches.

Keywords: weakly supervised, object localization

1. Introduction

Due to the popularity of online video sharing websites (e.g. YouTube), an ever-increasing amount of video content is becoming available nowadays. These online videos prove to be both a valuable resource and a grand challenge for computer vision. Internet videos are often weakly labeled. For example, many

*Corresponding author

Email addresses: mrochan@cs.umanitoba.ca (Mrigank Rochan), shafin12@cs.umanitoba.ca (Shafin Rahman), bruce@cs.umanitoba.ca (Neil D. B. Bruce), ywang@cs.umanitoba.ca (Yang Wang)

YouTube videos have some tags associated with them. Those tags are generated by users and provide some information about the contents (e.g. objects) of the video. However, these tags only provide the presence/absence of objects in the video, but they do not provide detailed spatial and temporal information about where the objects are. For instance, if a YouTube video is tagged with "dog", we know there is probably a dog somewhere in the video. But this does not indicate the location of the dog in the video. In this paper, we consider the problem of localizing objects and generating pixel-level object segmentation from weakly labeled videos. This will enable us to accurately localize the object in the video.

Our work is motivated by previous work on learning localized concepts [1, 2, 3, 4, 5, 6] in videos. In this paper, we propose a simple and effective method to localize and segment the object corresponding to the tag associated with the video. Figure 1 illustrates the goal of this work. Given a video with a tag, say "car", we would like to localize and segment out the pixels in the video corresponding to the "car". In other words, we try to answer the question "where is the object" in the video? A reliable solution to this problem will provide better video retrieval and browsing experience for users. It will also help us to solve a wide range of tasks in video understanding.

There has been a lot of work on object detection (e.g. [7]) and segmentation (e.g. [8]) in the computer vision literature. The strategy proposed in these prior efforts typically relies on machine learning approaches to train a detection or segmentation model for each object category. They usually require a large amount of labeled training data. The final models are often limited to a handful of object classes that are present in the training data. For example, the detection and segmentation tasks in the PASCAL challenge [9] only deal with 20 fixed object categories. The Microsoft COCO dataset [10] contains 80 object categories. Although this is an improvement to the PASCAL dataset, the number of object categories is still small. The main difference in our work is that we do not require labeled training data. In principle, our method can be applied to videos of any object category.

In this paper, we introduce a method that combines object appearance mod-

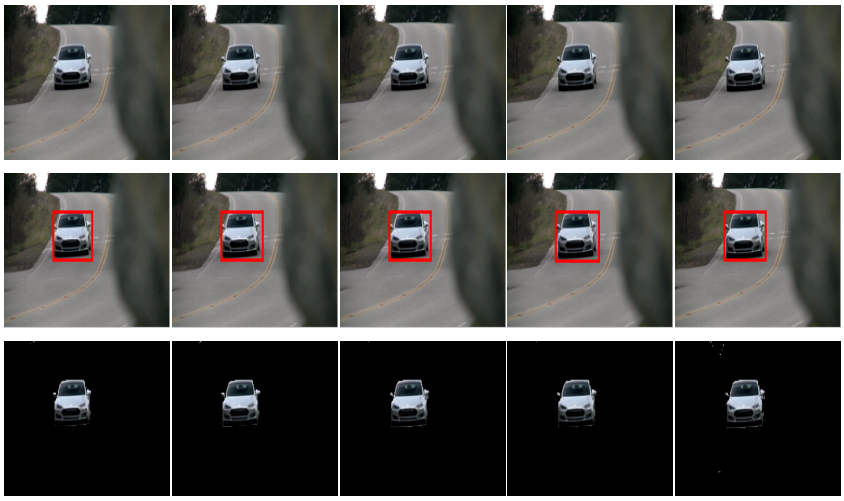


Figure 1: An illustration of our work. Given an input video with an object tag, e.g. "car" (1st row), our proposed method can automatically localize (2nd row) and segment (3rd row) the object of interest in each frame.

eling and temporal consistency among frames in a principled framework. For a given video with an object tag, at first we construct a video specific object appearance model, and then we enforce temporal consistency between two consecutive frames to make the object localization algorithm more efficient.

Preliminary versions of this work have appeared in Rochan et. al [11] and Rochan and Wang [12]. In [11], we proposed an averaging-based method to learn the video-specific object appearance model in order to localize the object of interest in weakly labeled videos. In [12], we introduced a temporal consistency constraint between consecutive frames to improve the performance of the framework in [11]. In this paper, in addition to applying the averaging-based appearance model method, we also propose an alternative way (i.e. SVM-based) to build the appearance model of the object of interest. Moreover, we also conduct experiments with state-of-the-art CNN features, whereas in [11] and [12] we have only used the normalized color histogram features.

The rest of the paper is organized as follows: Section 2 reviews previous work in spatio-temporal segmentation of videos. Section 3 provides detailed descrip-

tion of our proposed approach. We present experimental results in Section 4 and conclude in the last section.

2. Previous Work

Semantic video segmentation is a well studied problem in computer vision. It has been tackled with various levels of supervision. For example, some of the proposed approaches are unsupervised, e.g. region tracking [13], hierarchical graph models [14, 15], multiple hypothesis tracking [16] and spatio-temporal based segmentation [17, 18]. Unsupervised methods can perform low-level segmentation but cannot provide semantic labels for the segments. Supervised methods have also been studied for semantic video segmentation [19]. The major drawback of supervised video segmentation is that it requires lots of labeled video data for training. To address this issue, semi-supervised video segmentation [20, 21] methods are proposed. These methods address the limitation of supervised methods to some extent, but getting sufficient pixel-level labeled data is still nontrivial. In order to further reduce the need of labeled data, weakly supervised semantic segmentation techniques are proposed [5, 1]. Our proposed method is inspired by this line of research. In general, these methods use the weak label (e.g. semantic tags) associated with the videos and thus do not require pixel-level label information. Since it is much easier to tag a video than labeling each pixel within it, the need for human annotation can be greatly reduced.

Our work is related to a line of research on fully automatic and semi-supervised video segmentation. Perazzi et al. [22] perform segmentation in videos using multiple object proposals. The problem of video segmentation is formulated as an energy minimization over a fully connected graph defined on the object proposals. Note that this method requires some manually annotated foreground proposals. The method proposed by Zhang et al. [23] is also related to ours. This method extracts the object regions in videos. It then uses a Directed Acyclic Graph (DAG) based approach to detect and segment the

object of interest in every frame of a video.

Our work is motivated by recent work that uses object annotation for various tasks in video understanding, including human activity recognition[24], event detection [25], and object segmentation [1, 26].

Recent work on video co-localization [3, 27] is very relevant to our work. They tackle the problem of co-localization in videos by proposing candidate regions in each frame and then select the correct one from each video. In [3], the authors leverage temporal information by proposing candidate tubes, but their learning algorithm still suffers from poor performance. However, Tang et al. [27] consider the temporal information directly in their model. Our temporal consistency formulation is very similar to this method.

One major advantage of our technique is that it can be easily used for object annotation in videos, which has been of increasing interest among computer vision researchers. For example, Tang et al. [5] presented an algorithm for annotating spatio-temporal segments using video-level tags provided in Internet videos. Our work is closely related to this line of research, since our goal is also to build an effective approach for object annotation in Internet videos.

Our proposed method is also inspired by some work on tracking humans [28, 29] or animals [30] by learning video-specific appearance models. For example, the human kinematic tracking system in [29] first detects stylized human poses in a video, then builds an appearance for human limbs specifically tuned to the person in this particular video. It then applies the appearance model to all frames in the video. At a high level, our proposed approach operates on a similar idea.

Our work is also related to a line of research on weakly supervised learning (in particular, multiple-instance learning) in computer vision. For example, Maron et al. [31] applied multiple-instance learning for scene classification. Galleguillos et. al [32] proposed MIL-based method to recognize and localize objects in images. Recently, multiple-instance learning has been adopted in many computer vision applications, e.g. object detection [7], image annotation [6], etc.

3. Our Approach

The type of input processed by our method is a video with an object tag, e.g. "cow". In our work, we focus on videos that are relatively simple. In particular, we make the following two assumptions about the videos: 1) the tag corresponds to the main object in the video; 2) there is only one instance of the tagged object in the video. More concretely, if a video is tagged with "cow", there should be a cow somewhere in the video. We assume the cow is the dominant object in the video, i.e. it is not too small. We also assume there is only one cow in the video. Previous work (e.g. [5]) in this area makes similar assumptions.

Based on these assumptions, our proposed approach involves four major steps:

1) Generating object proposals: Given a video with an object tag, the first step of our approach is to generate a collection of *object proposals* (also called *hypotheses*) on each frame in the video. Each object proposal is a bounding box that is likely to contain an object. The method we use for generating object proposals is generic and is not tuned for any specific object classes.

2) Building object appearance model: Many of the object proposals obtained from the previous step might not correspond to the object of interest. In the second step, we use some simple heuristics to choose a few bounding boxes from the collection of all object proposals. The hope is that these selected bounding boxes are likely to correspond to the object of interest. We then build an appearance model for the object based on the selected bounding boxes. Note that the appearance model is built for a specific video. If the video contains a "black cow", our appearance model will try to detect this "black cow", instead of other generic cows.

3) Object localization: We localize the object by selecting one bounding box in each frame of a video. We could use the learned appearance model from the previous step to re-score the object proposals from the first step. After re-scoring, a bounding box will have a high score only if it is likely to contain

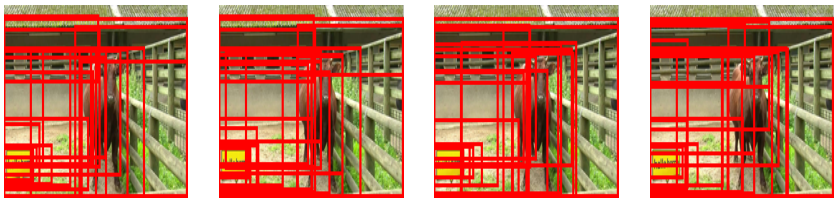


Figure 2: Examples of generating object proposals on frames within a video. Given a frame, the Edge Boxes algorithm [34] is applied. It returns a collection of bounding boxes in an image that are likely to be *any* object. For each bounding box, the algorithm also assigns a score indicating how likely it is to be an object.

an object instance specific to this video, e.g. a "black cow". However, this strategy alone may not be efficient enough to localize the object correctly. In this paper, we assume that the object of interest in a video does not undergo drastic change in their properties such as size, position and appearance between two consecutive frames. Previous work (e.g. [23, 27]) has also made similar assumptions. Therefore, we enforce these constraints by incorporating a temporal consistency model between adjacent video frames. We model the object localization problem as performing the maximum a posteriori (MAP) inference in an undirected chain graphical model. Each node in the graphical model corresponds to a frame and object proposals within a frame are the possible states of the node. An edge in the model enforces temporal consistency between two consecutive frames. The object in the video is localized by finding the optimal labeling of nodes in the graphical model.

4) Segmenting objects: After localizing an object in each frame, the GrabCut [33] algorithm is applied on the selected bounding box to segment the object from the background.

We describe the details of each step in the following.

3.1. Generating Object Proposals

Given an input video, the first step of our approach is to generate a set of candidate object bounding boxes on each frame. For certain object categories (e.g. people, car, etc.), one might be able to use state-of-the-art object de-

tectors, e.g. [7]. But the limitation of this approach is that there are only a handful of object categories (e.g. 20 object categories in the PASCAL object detection challenge) for which we have reasonably reliable detectors. Since we are interested in segmenting objects in a video regardless of the object class, we choose not to use object detectors.

Instead, we use the Edge Boxes algorithm [34] to generate object bounding box proposals. This algorithm relies on one simple observation: the number of contours that are wholly enclosed by a bounding box is indicative of the presence of the object within the bounding box. The object proposals are detected using the edge maps. For a given bounding box, the algorithm also defines an objectness scoring function which measures the likelihood of this bounding box containing an object. Since this algorithm is not restricted to any particular object classes and therefore can be potentially used for any object categories. We choose to use this algorithm to generate the object proposals.

Given an input video, we apply the Edge Boxes algorithm [34] to generate 10 object proposals (i.e. bounding boxes) for every frame within the video. This gives us a collection of candidate bounding boxes which are likely to contain an object. Figure 2 shows some examples of applying the Edge Boxes algorithm on frames within a video.

3.2. Building Object Appearance Model

Given a video, the Edge Boxes algorithm approach (see Section 3.1) gives us a collection bounding boxes. Those bounding boxes correspond to image windows that are likely to contain *any* object. However, since this algorithm is a generic for any object class, it is not specifically tuned for any specific object categories. Figure 3 shows some examples of bounding boxes with high objectness scores, but that do not correspond to the object of interest (aeroplane) in the video. The next step of our approach is to select a few bounding boxes from all the generated object proposals. Ideally, the bounding boxes being selected will correspond to the object of interest in the video.

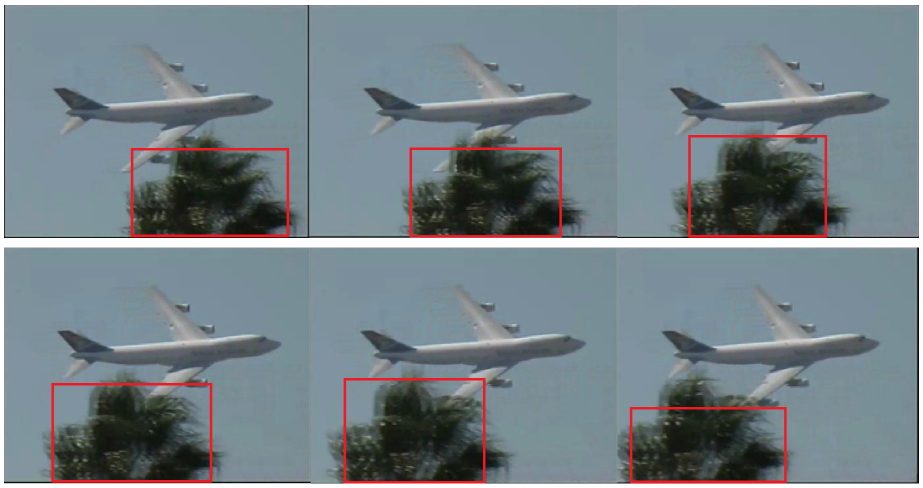


Figure 3: Example of high scoring bounding boxes on an image that do not correspond to the object of interest (aeroplane).

Our bounding box selection strategy is based on the following two observations. First, if a video is tagged with an object, say "cow", the image windows corresponding to the "cow" in the video tend to have high objectness scores. The reason is that people are less likely to tag an object if it is not salient (e.g. too small) in the video. Second, we assume there is only one instance of the object of interest in the video. I.e. if a video is tagged as "cow", we only consider segmenting one "cow" in the video. In this case, the object of interest tends not to change appearance across different frames in the video. For example, if we know a "cow" is black in one frame, we know that it must be black in other frames as well. If we can somehow build an appearance model for this specific "black cow", we can use this appearance model to find "cow" bounding boxes in other frames.

Note that since our goal is to build an appearance model for the object of interest, our bounding box selection strategy does not necessarily have to retrieve all the true positive examples. As long as most of the bounding boxes being selected are positive examples of this object, we will be able to build a good appearance model for this object. In other words, we would like our

bounding box selection to have a precision, but can tolerate a low recall.

In our work, we use a simple yet effective strategy. We observe that if a video is tagged as "cow", most of the bounding boxes with the highest objectness scores tend to correspond to this object. This suggests that we can simply sort the bounding boxes in a video according to their objectness scores. Then we select K bounding boxes with the highest objectness scores. We empirically find the number of frames within a video to be a good choice for K and use this value in all of our experiments.

Based on the selected K bounding boxes, we build a video-specific appearance model for the object. We first extract the visual feature from each bounding box. In our experiments, we have used both the normalized color histogram and the CNN-based features implemented in Caffe [35]. We define two methods for building the appearance model. (1) *Averaging*: in this method, we simply take the average of the feature vectors extracted from all selected bounding boxes. Let A be the appearance model obtained by this method and $\mathbf{x}$ be the feature vector of an object proposal. We can use $(-||A - \mathbf{x}||_2)$ as a measure of how likely it is that $\mathbf{x}$ is the object in this video. (2) *SVM-based*: in the second method, we learn a model of the object of interest from the object proposals extracted from the video frames. We consider the selected K bounding boxes as positive examples of the object present within the video. We then choose a set of negative examples by randomly selecting object proposals from videos that do not correspond to the object of interest. Given this set of positive and negative examples, we train a linear SVM (with either color histogram or CNN features) to learn the video specific object appearance model. Let $\mathbf{x}$ be the feature vector (normalized color histogram or CNN features) of an object proposal in a video $\mathbf{v}$, the video specific object appearance model is represented by parameter vector $\mathbf{w}_v$. The dot product $\mathbf{w}_v^\top \mathbf{x}$ indicates the likelihood of $\mathbf{x}$ being the specific object in the video $\mathbf{v}$.

3.3. Object Localization

We have a set of bounding boxes for every frame within a video. In this section, our goal is to localize the object in the video by selecting one bounding box for each frame. We could use the learned appearance model from Section 3.2 to localize the object of interest within a given video. I.e., we can use the learned appearance model to re-score the bounding boxes with the frames of that video. A bounding box will have a high score only if it is likely to contain an object instance specific to this video, e.g. a "black cow". However, this strategy alone may not be sufficient to localize the object correctly. We know that within a video it is very unlikely that objects will undergo drastic change in their properties such as size, position and appearance between two consecutive frames of a video. This prior is often used in tracking [27, 36, 37, 38, 39, 40, 41, 42] objects in videos. Therefore, we enforce a temporal consistency model between consecutive video frames.

We model the object localization problem within a video using an undirected chain graph. Each node in the graph represents a frame within a video. The value assigned to a node indicates which object proposal is chosen for this frame. Since we have 10 object proposals for each frame, each node can take its value from $\{1, 2, ..., 10\}$. The nodes of two adjacent frames are connected by an edge indicating the temporal consistency constraint between these two frames. Let $X_1, X_2, ..., X_k$ be the frames in a video with k frames, and $P_1, P_2, ..., P_k$ be the corresponding object proposals selected for each frame. We use the following optimization problem to solve the object localization:

$$\max_{P_1,P_2,...,P_k} \sum_i \phi(P_i, X_i) + \sum_{i,i+1} \psi(P_i, P_{i+1}) \tag{1}$$

This optimization problem in Eq. 1 involves unary potential functions $\phi(\cdot)$ defined on nodes and pairwise potential functions $\psi(\cdot)$ defined on edges in the graph. In the following, we describe these potential functions in detail.

3.3.1. Unary Potentials

The unary potential $\phi(\cdot)$ measures the likelihood that an object proposal belongs to the object class, i.e. it captures the compatibility between an object proposal and the appearance model of the object. We use two different ways to define the unary potential. Firstly, we define the unary potential for each frame as follows:

$$\phi(P_i, X_i) = \exp\Big(-\|A - f_h(P_i, X_i)\|_2\Big) \tag{2}$$

where A is the appearance model (obtained by averaging) of the object of interest within the video (see Sec. 3.2) and $f_h(P_i, X_i)$ is the feature vector (color histogram or CNN feature) of the image patch corresponding to the bounding box P_i in the frame X_i.

Secondly, we also use the video specific object appearance model learned using SVM to define the unary potential for each frame. In this case, the unary potential is computed as follows:

$$\phi(P_i, X_i) = \Big(\mathbf{w}_v^\top \cdot f_h(P_i, X_i)\Big) \tag{3}$$

where $\mathbf{w}_v$ is the learned video object specific appearance model and $f_h(P_i, X_i)$ is the feature vector from the image patch corresponding to the bounding box P_i in the frame X_i.

The unary potential in Eq. 2 and Eq. 3 will encourage each frame to choose a bounding box whose appearance (i.e. color histogram or CNN feature vector) is consistent with the video specific appearance model of the object. We conduct experiments with both the definitions of unary potential with both color histograms and CNN features.

3.3.2. Pairwise Potentials

The pairwise potential is a term which encourages the temporal consistency between the bounding boxes selected in two adjacent frames. It ensures that the bounding boxes selected between adjacent frames do not undergo drastic changes in their properties such as size and position.

Following [27], we define the temporal consistency $C_{temporal}(P_i, P_j)$ between two bounding boxes P_i and P_j) of adjacent video frames as follows:

$$C_{temporal}(P_i, P_j) = \alpha\left(\|f_c(P_i) - f_c(P_j)\|_2^2 + \|f_a(P_i) - f_a(P_j)\|_2^2\right) \quad (4)$$

where $f_c(P_i)$ denotes the coordinates of the center of the bounding box P_i, and $f_a(P_i)$ denotes the area of this bounding box. We normalize $f_c(P_i)$ by the height and width of the frame, and $f_a(P_i)$ by the maximum area between the two object proposals.

Using the above temporal consistency definition, we compute the pairwise potential between two bounding boxes of adjacent video frames as follows:

$$\psi_v(P_i, P_j) = \exp\left(-\left(C_{temporal}(P_i, P_j)\right)^2\right) \quad (5)$$

The parameter α in Eq. 4 control the relative influence of the pairwise potential in the model.

The pairwise potential is very intuitive because if two object bounding boxes of adjacent frames contain the same object then they should not be far apart and their area should not vary either. In summary, the pairwise potential encourages the algorithm to select bounding boxes that are consistent in terms of positions and sizes between adjacent video frames.

3.3.3. Decoding

Given the model defined above, the inference problem we need to solve is to jointly choose the values of $P_1, P_2, ..., P_k$ to maximize Eq. 1. Figure 4 illustrates this inference problem. Each column in Fig. 4 corresponds to a frame. In each column, the rows indicate the object proposals in that frame. The inference problem can be interpreted as finding the optimal path from the start to end in Fig. 4. It can be efficiently solved by dynamic programming.

3.4. Segmenting Object of Interest

Finally, we apply GrabCut [33] to segment out the object in each frame. GrabCut is an efficient algorithm for foreground segmentation in images. The

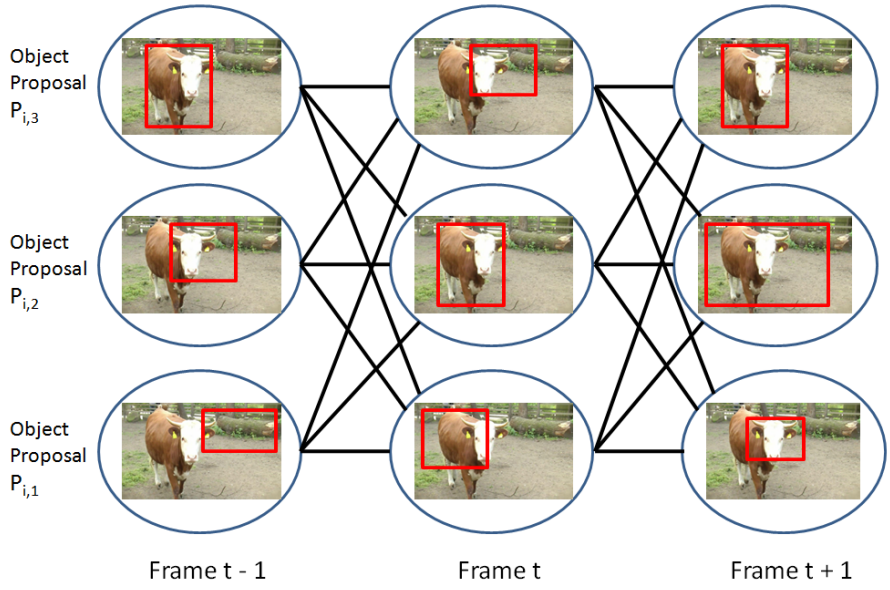
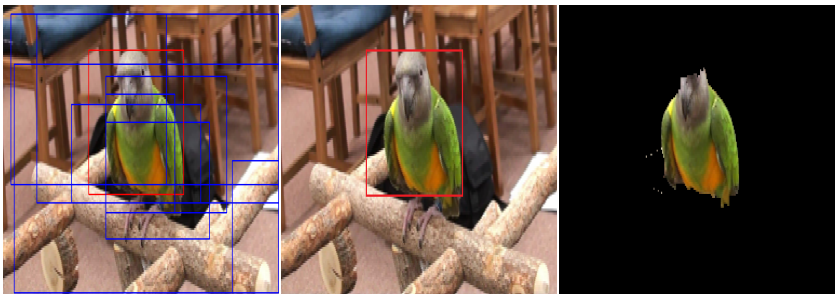


Figure 4: For the given consecutive frames of a video, the inference problem for object localization can be represented as finding the optimal path in a graph. Each frame in the graph represents the node and their object proposals (blue circle) represent the possible state that node can take. The edges between the object proposals of two frames indicate the pairwise consistency constraint between the bounding boxes of two adjacent frames . Our goal is to find the best configuration of object bounding boxes among the frames of the video. This is equivalent to finding the optimal path in the graph.

standard GrabCut is not fully automatic. It requires the user input in the form of marking a rectangle around the foreground object. In contrast, our approach does not require user interaction. We simply consider the one bounding box selected by our localization algorithm within each frame as the user input. Figure 5 illustrates the pipeline of our approach.

4. Experiments

In this section, we first describe the dataset and evaluation metrics (Sec. 4.1). We then present our experimental results in Sec. 4.2.



(a) (b) (c)

Figure 5: An illustration of our approach. (a) A frame in the video with selected bounding boxes (see Sec. 3.2). An appearance model is built based on the selected bounding boxes from all frames of this video. (b) After applying the appearance model on this frame, we obtain a single bounding box that is most likely to contain the object of interest (bird) in this frame. (c) The GrabCut algorithm is applied to segment the object in this frame. The standard GrabCut algorithm requires users to draw a rectangle around the foreground object as the part of the input. In our case, we use the bounding box obtained from (b) as the user input. So our method is fully automatic and does not require any user interactions.

4.1. Dataset and Setup

We evaluate our proposed approach using a subset of the dataset in Tang et al. [5]. This dataset consists of video shots collected for 10 different object classes, including aeroplane, bird, boat, car, cat, cow, dog, horse, motorbike, and train. Each frame of the video shot is annotated with the segmentation of the object of interest in the video. Table 1 shows the summary of this dataset. We use 144 video shots with a total of 24,723 frames in our experiments.

We define a quantitative measurement in order to evaluate our approach. Our quantitative measurement is inspired by the measurement used in the PASCAL challenge [9]. Given a video frame, let P_b be the foreground pixels returned by our method and P_{gt} be the ground-truth foreground pixels provided by the annotation in the dataset. We measure the quality of P_b by the ratio of $|P_b \cap P_{gt}|$ and $|P_b \cup P_{gt}|$:

$$r = |P_b \cap P_{gt}| \,/\, |P_b \cup P_{gt}| \tag{6}$$

Class	Number of Shots	Number of Frames
Aeroplane	9	1423
Bird	6	1206
Boat	17	2779
Car	8	601
Cat	13	3870
Cow	20	2978
Dog	27	3803
Horse	17	3990
Motorbike	10	827
Train	18	3270
Total	144	24723

Table 1: Summary of the dataset used in the experiments.

If this ratio r is greater than 50%, we consider the segmentation on this frame to be correct. We evaluate the performance of our algorithm by computing the percentage of frames that are correctly segmented.

We extract 10 object proposals (or bounding boxes) from each frame of a video shot. We use normalized color-histograms and state-of-the-art 4096 dimensional fine-tuned CNN features [35] as our feature representations for an object proposal. We randomly choose one video shot from every object class for setting the free parameter α (see Section 3.3) in our experiments.

4.2. Results

In order to measure the performance of our proposed approach, we perform several experiments.

We first consider using the averaging-based appearance model based on color histogram (see Sec. 3.2). We compare our method with several baseline approaches. The first baseline simply chooses the bounding box with the highest objectness score (from Edge Boxes algorithm [34]) for each frame within a video.

method	aero	bird	car	cow	mbike	boat	cat	dog	horse	train	avg.
top proposal only	52.5	**46.3**	42.5	33.3	5.0	24.8	**17.4**	34.1	21.0	10.9	28.8
appearance only	54.6	37.8	49.7	42.3	9.3	25.5	16.1	34.8	21.6	12.3	30.4
ours	**57.5**	38.1	**50.3**	**44.4**	**10.5**	**27.1**	**17.4**	**36.0**	**22.9**	**12.4**	**31.7**

Table 2: Quantitative results using the averaging-based appearance model on color histogram features. For each object class, we compare segmentation accuracy across the sequence of video frames. A frame is considered to be correctly segmented if the ratio of intersection over union defined in Eq. 6 is greater than 50%. We compare four different methods: (1st row) bounding box with highest objectness score selected on each frame; (2nd row) video specific appearance model generated 3.2 using normalized color-histogram feature from top-scored bounding boxes 3.2; (3rd row) incorporating temporal consistency between two consecutive frames with the color histogram based video specific object appearance model.

method	aero	bird	car	cow	mbike	boat	cat	dog	horse	train	avg.
top proposal only	52.5	**46.3**	42.5	33.3	5.0	24.8	**17.4**	34.1	21.0	10.9	28.8
appearance only	58.1	41.6	42.7	34.9	7.5	26.5	11.7	34.2	22.9	**11.5**	29.2
ours	**58.9**	42.4	**46.7**	**37.1**	**7.7**	**27.1**	12.5	**35.9**	**23.0**	11.3	**30.3**

Table 3: Quantitative results using the averaging-based appearance model on CNN features.

We call this baseline "top proposal only". The second baseline applies the video specific object appearance model (averaging based on color histogram) to re-score the object proposals on each frame, then selects the proposal with the highest score. Note that this baseline does not consider the temporal consistency information between the object proposals selected from adjacent frames of a video. We call this baseline "appearance only". Table 2 shows the performance of three methods: 1) using first baseline method, i.e. "top proposal only"; 2) using second baseline method, i.e. "appearance only"; 3) using our method that combines video specific object appearance with temporal consistency. Our approach achieves the best performance on most of the object classes.

Table 3 shows the performance of different methods using the averaging-based appearance model based on CNN features. Similar to Table 2, we compare the performance of three methods: 1) using "top proposal only"; 2) using the averaging-based appearance model based on CNN feature, i.e. "appearance only"; 3) using our approach that combines video specific object appearance model (CNN feature based) with temporal consistency information. Our final

method	aero	bird	car	cow	mbike	boat	cat	dog	horse	train	avg.
appearance only	52.9	36.5	41.6	30.0	**10.1**	15.2	10.2	26.3	21.5	9.8	25.4
ours	**60.3**	**38.6**	**56.3**	**36.1**	9.9	**16.3**	**14.0**	**30.3**	**25.1**	**11.3**	**29.8**

Table 4: Quantitative results using the SVM-based appearance model based on color histogram. We learn a video specific appearance model using a linear SVM without the bias term. We select the object proposal with highest objectness score on each frame of a given video as positive example and select a set of negative examples by randomly choosing object proposals from videos of different object class. We compare performance of two methods: (1st row) using only the learned video specific appearance model; (2nd row) incorporating temporal consistency between two consecutive frames with the video specific appearance model.

method	aero	bird	car	cow	mbike	boat	cat	dog	horse	train	avg.
appearance only	**60.8**	53.6	56.3	41.1	**11.8**	34.2	**19.5**	34.7	30.2	**11.7**	35.4
ours	**60.8**	**54.6**	**57.4**	**42.1**	11.7	**34.7**	19.2	**35.8**	**30.4**	11.4	**35.8**

Table 5: Quantitative results on using SVM-based appearance model based on CNN features. Similar to Table 4, we compare the performance of two methods: (1st row) using appearance model only; (2nd row) incorporating temporal consistency to the framework.

method again outperforms the other baseline methods.

We further investigate and evaluate the performance of video specific object appearance model learned using SVM 3.2. Table 4 shows the performance of two methods: 1) using video specific object appearance model learned with normalized color histogram feature from object proposals , i.e. "appearance only"; 2) using temporal information with the SVM-based appearance model. Similar to Table 4, we also compare the two methods when CNN feature is used to learn the video specific object appearance model (see Table 5). In both the cases, our final approach outperform the other baseline method. Note that, in contrast to Table 2 and Table 3, we obtain better performance with CNN feature rather than color histogram for SVM-based methods. The main reason is that SVM learns a better appearance model using the high-dimensional discriminative CNN feature representation than the low dimensional color histogram feature. These results are also in agreement with many CNN feature representation based visual recognition algorithms where the representation has proved to be one of the state-of-the-art.

method	aero	bird	car	cow	mbike	boat	cat	dog	horse	train	avg.
color-hist	53.6	35.0	29.6	31.4	7.0	16.6	8.9	19.5	20.4	8.9	23.1
CNN	**60.7**	**52.5**	**54.7**	**40.5**	**12.1**	**33.4**	**20.6**	**37.1**	**26.3**	**11.0**	**34.9**

Table 6: Quantitative results on using SVM-based appearance model learned using negative training examples drawn from all other videos. We use both color histogram (1st row) and CNN (2nd row) features to learn this variant of SVM-based appearance model.

Tables 2–5 show that our final approach (video specific object appearance model with temporal consistency) outperforms the baseline methods on most of the object categories. Firstly, from various results, we observe that building a video specific object appearance model (averaging-based or SVM-based) is an effective strategy to tackle the localization problem in weakly labeled video. Secondly, we show that incorporating temporal consistency information to the framework further improves the performance. Qualitative results of our approach on these 10 object classes are shown in Fig. 8 and Fig. 9.

We also perform an additional experiment (see Table 6) in which we learn the SVM-based appearance model using negative training examples drawn from all other videos. This is essentially an "unsupervised" variant of our SVM-based "appearance only" method in Tables 4 and 5. Note that the "appearance only" method in Tables 4 and 5 requires weak supervision because we use the video tag to select negative examples (i.e. videos that do not correspond to the object of interest) for training the SVM. While in this "unsupervised" variant, we consider all the other videos (including those that might contain the object of interest) as negative examples. Comparing the results in Table 6 with Tables 4 and 5, we can see the "unsupervised" variant does not perform as well as the weakly supervised variant. This is not surprising since the "unsupervised" variant has less supervisions.

Figure 6 shows two examples demonstrating the benefit of having the pairwise potential in the model. Without the pairwise potential (1st row and 3rd in Fig. 6), the selected bounding boxes between adjacent frames of a video can vary dramatically in terms of size and position. The pairwise potential alleviates this problem and enforces the consistency across the selected bounding boxes

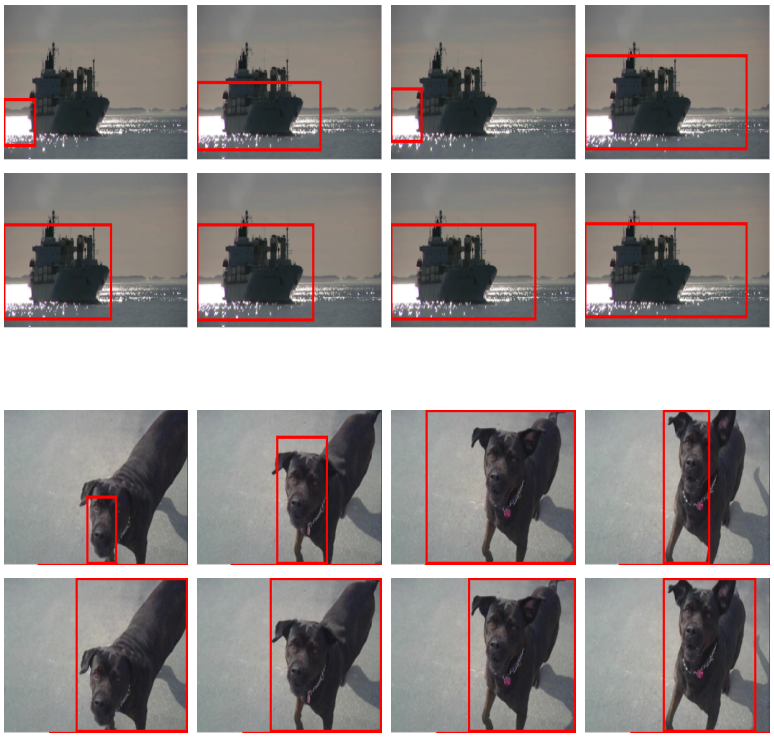


Figure 6: Examples illustrating the benefit of enforcing consistency between adjacent frames of videos. (1st and 3rd row) Without the pairwise potential, the selected bounding boxes can be dramatically different. (2nd and 4th row) With the pairwise potential, the bounding boxes are more consistent across all frames.

between consecutive frames of a video (2nd row and 4th row in Fig. 6).

4.3. Failure Cases

In Fig. 7, we show some representative failure cases of our approach. The failures are often caused by occlusion, multiple instances of the object of interest and the object of interest being too small in the scene.

5. Conclusion

We have introduced an efficient approach for localizing and segmenting the object of interest in weakly labeled videos. Our approach is fully automatic and does not require any user interaction. Our approach is based on two main

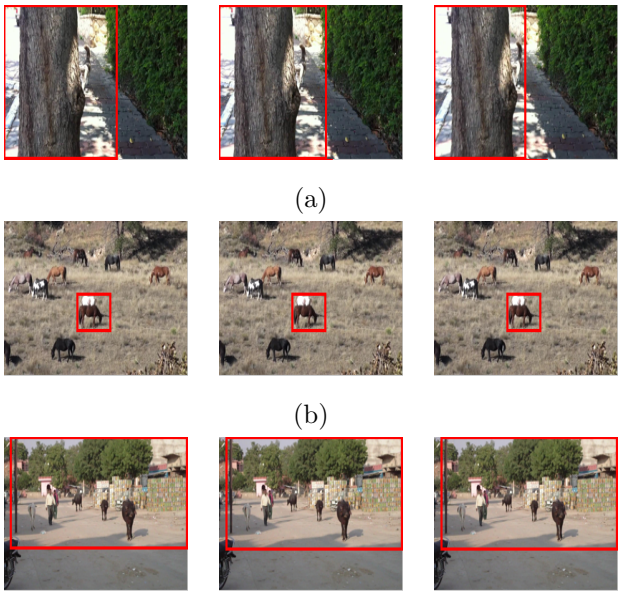


Figure 7: Some typical failure cases of our approach: (a) occlusion; (b) multiple instances of the object of interest; (c) object of interest is too small in the scene.

observations. First, the main object in a video tends to be salient (i.e. object-like). Second, the object appearance does not change across different frames in a video. We introduced a chain structured graphical model formulation to tackle this problem. We then use dynamic programming to select best bounding box (i.e. object of interest location) within each frame of a given video. We demonstrate the effectiveness of our approach by comparing with several other baseline methods.

There are many possible directions for future work. First of all, we would like to extend our approach to handle multiple object instances in a video. Secondly, for some object categories (e.g. people, car), reliable object detectors do exist. We would like to incorporate those object detectors in our framework. Thirdly, we like to use our proposed method as a starting point towards the grand goal of understanding contents of online videos (e.g. YouTube).

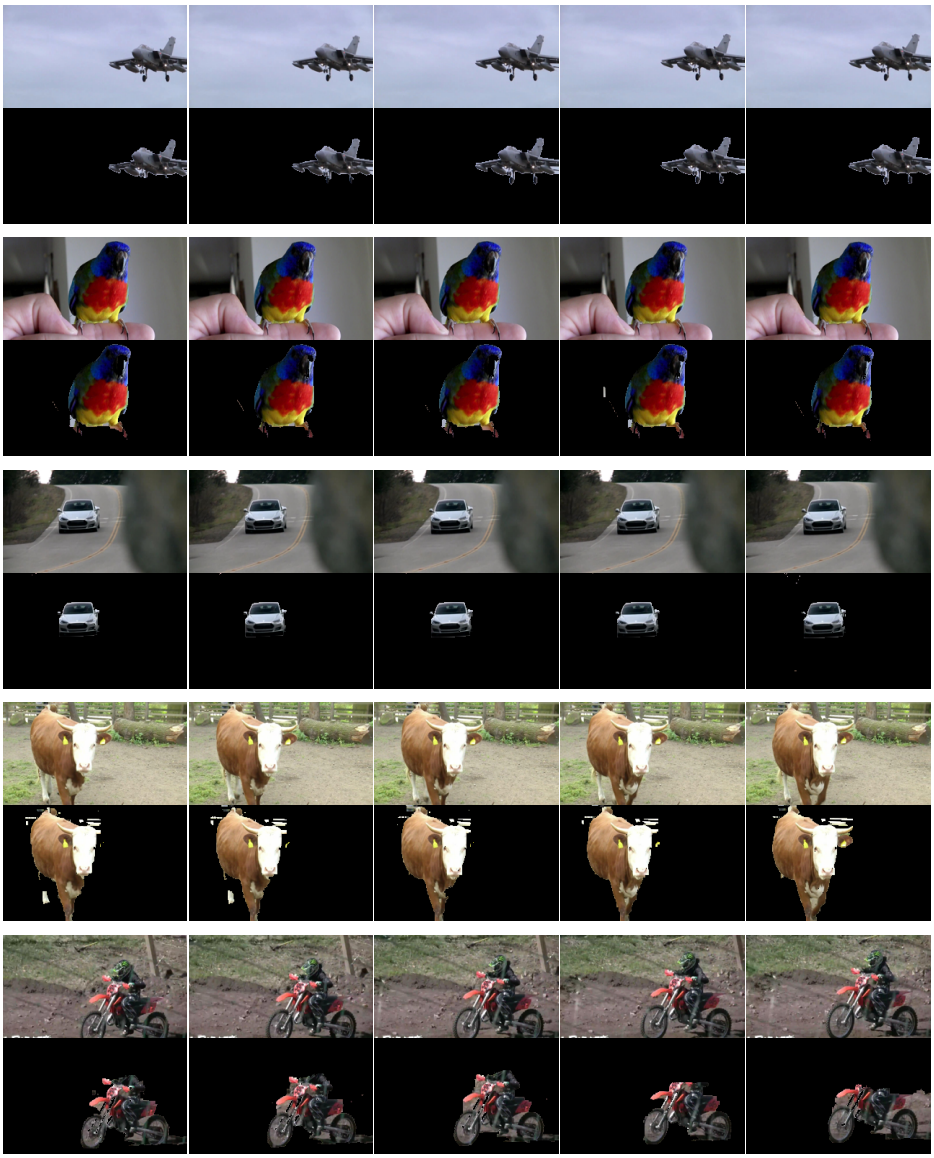


Figure 8: Example results on videos tagged as (from top to bottom) “aeroplane”, “bird”, “car”, “cow”, and “motorbike” respectively. For each video, we show the original frames (1st row) and the segmentation results obtained after localization (2nd row).

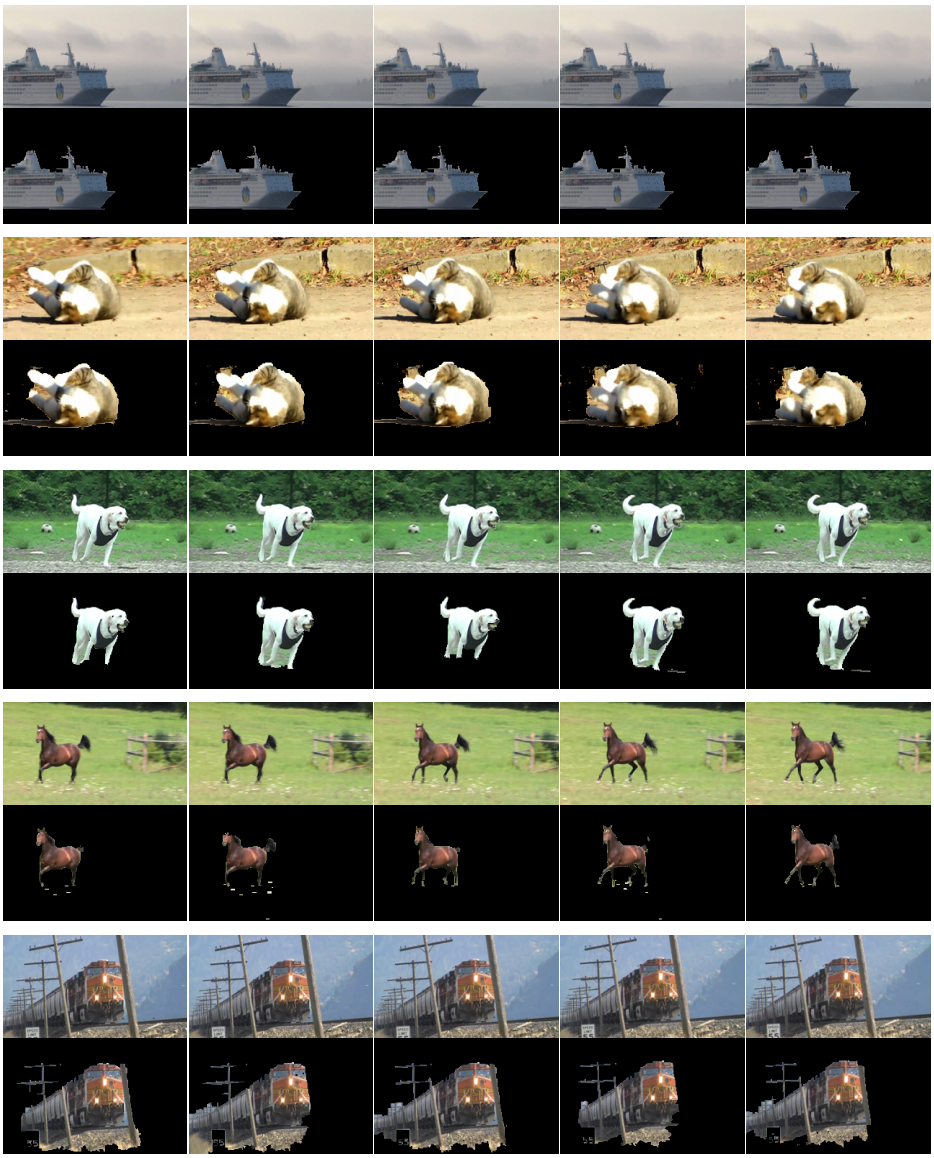


Figure 9: Example results on videos tagged as (from top to bottom) “boat”, “cat”, “dog”,“horse”, and “train”, respectively. For each video, we show the original frames (1st row) and the segmentation results obtained after localization (2nd row).

6. Acknowledgments

This work was supported by NSERC and the University of Manitoba Research Grants Program (URGP).

References

[1] G. Hartmann, M. Grundmann, J. Hoffman, D. Tsai, V. Kwatra, O. Madani, S. Vijayanarasimhan, I. Essa, J. Rehg, R. Sukthankar, Weakly supervised learning of object segmentations from web-scale video, in: ECCV Workshop on Web-scale Vision and Social Media, 2012, pp. 198–208.

[2] J. C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: European Conference on Computer Vision, 2010, pp. 392–405.

[3] A. Prest, C. Leistner, J. Civera, C. Schmid, V. Ferrari, Learning object class detectors from weakly annotated video, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3282–3289.

[4] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1250–1257.

[5] K. D. Tang, R. Sukthankar, J. Yagnik, F.-F. Li, Discriminative segment annotation in weakly labeled video, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2483–2490.

[6] Y. Wang, G. Mori, A discriminative latent model of image region and object tag correspondence, in: Advances in Neural Information Processing Systems, 2010, pp. 2397–2405.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1672–1645.

[8] J. Shotton, J. Winn, C. Rother, A. Criminisi, TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: European Conference on Computer Vision, 2006.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision, 2014.

[11] M. Rochan, S. Rahman, N. D. Bruce, Y. Wang, Segmenting objects in weakly labeled videos, in: The 11th Conference on Computer and Robot Vision, IEEE, 2014, pp. 119–126.

[12] M. Rochan, Y. Wang, Efficient object localization and segmentation in weakly labeled videos, in: International Symposium on Visual Computing, Springer, 2014, pp. 172–181.

[13] W. Brendel, S. Todorovic, Video object segmentation by tracking regions, in: IEEE International Conference on Computer Vision, 2009, pp. 833–840.

[14] M. Grundmann, V. Kwatra, M. Han, I. Essa, Efficient hierarchical graph-based video segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2141–2148.

[15] C. Xu, C. Xiong, J. J. Corso, Streaming hierarchical video segmentation, in: European Conference on Computer Vision, 2012, pp. 626–639.

[16] A. Vazquez-Reina, S. Avidan, H. Pfister, E. Miller, Multiple hypothesis video segmentation from superpixel flows, in: European Conference on Computer Vision, Springer, 2010, pp. 268–281.

[17] J. Lezama, K. Alahari, J. Sivic, I. Laptev, Track to the future: Spatio-temporal video segmentation with long-range motion cues, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3369–3376.

[18] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: European Conference on Computer Vision, 2010, pp. 282–295.

[19] S. H. Raza, M. Grundmann, I. Essa, Geometric context from videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3081–3088.

[20] V. Badrinarayanan, F. Galasso, R. Cipolla, Label propagation in video sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3265–3272.

[21] V. Badrinarayanan, I. Budvytis, R. Cipolla, Semi-supervised video segmentation using tree structured graphical models, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2751–2764.

[22] F. Perazzi, O. Wang, M. Gross, A. Sorkine-Hornung, Fully connected object proposals for video segmentation, in: IEEE International Conference on Computer Vision, 2015, pp. 3227–3234.

[23] D. Zhang, O. Javed, M. Shah, Video object segmentation through spatially accurate and temporally dense extraction of primary object regions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 628–635.

[24] W. Brendel, S. Todorovic, Learning spatiotemporal graphs of human activities, in: IEEE 11th International Conference on Computer Vision, 2011, pp. 778–785.

[25] Y. Ke, R. Sukthankar, M. Hebert, Event detection in crowded videos, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[26] Y. J. Lee, J. Kim, K. Grauman, Key-segments for video object segmentation, in: IEEE International Conference on Computer Vision, 2011, pp. 1995–2002.

[27] A. Joulin, K. Tang, L. Fei-Fei, Efficient image and video co-localization with frank-wolfe algorithm, in: European Conference on Computer Vision, 2014.

[28] D. Ramanan, D. A. Forsyth, Finding and tracking people from the bottom up, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[29] D. Ramanan, D. A. Forsyth, A. Zisserman, Strike a pose: Tracking people by finding stylized poses, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 271–278.

[30] D. Ramanan, D. A. Forsyth, Using temporal coherence to build models of animals, in: IEEE International Conference on Computer Vision, 2003.

[31] O. Maron, A. L. Ratan, Multiple-instance learning for natural scene classification, in: International Conference on Machine Learning, 1998.

[32] C. Galleguillos, B. Babenko, A. Rabinovich, S. Belongie, Weakly supervised object localization with stable segmentations, in: European Conference on Computer Vision, Springer, 2008, pp. 193–207.

[33] C. Rother, V. Kolmogorov, A. Blake, Grabcut: Interactive foreground extraction using iterated graph cuts, in: ACM Transactions on Graphics (TOG), Vol. 23, ACM, 2004, pp. 309–314.

[34] C. L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: European Conference on Computer Vision, 2014.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv:1408.5093.

[36] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, Acm computing surveys (CSUR) 38 (4) (2006) 13.

[37] K. Tang, V. Ramanathan, L. Fei-Fei, D. Koller, Shifting weights: Adapting object detectors from image to video, in: Advances in Neural Information Processing Systems, 2012, pp. 638–646.

[38] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: European Conference on Computer Vision, Springer, 2002, pp. 661–675.

[39] Y. Pang, H. Ling, Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms, in: IEEE International Conference on Computer Vision, IEEE, 2013, pp. 2784–2791.

[40] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with kernels, in: IEEE International Conference on Computer Vision, IEEE, 2011, pp. 263–270.

[41] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2011) 1619–1632.

[42] J. Berclaz, F. Fleuret, E. Türetken, P. Fua, Multiple object tracking using k-shortest paths optimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (9) (2011) 1806–1819.