See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/264557646

Segmenting Objects in Weakly Labeled Videos

Conference Paper · May 2014

DOI: 10.1109/CRV.2014.24

CITATIONS

2

READS

8

4 authors, including:

Shafin Rahman

North South University

13 PUBLICATIONS **27** CITATIONS

SEE PROFILE

Neil D. B. Bruce

University of Manitoba

46 PUBLICATIONS **1,103** CITATIONS

SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Neil D. B. Bruce

Retrieved on: 28 July 2016

Segmenting Objects in Weakly Labeled Videos

Mrigank Rochan, Shafin Rahman, Neil D.B. Bruce, Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, Canada
{*mrochan, shafin12, bruce, ywang*}@*cs.umanitoba.ca*

***Abstract*—We consider the problem of 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*-video understanding; weakly supervised; object segmentation**

I. Introduction

Due to the popularity of online video sharing websites (e.g. Youtube), an ever-increasing amount of video contents are 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 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 we cannot localize the dog in the video. In this paper, we consider the problem of 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]–[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 “cow”, we would like to segment out the pixels in the video corresponding to the “cow”. 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. Most of these work use 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 available on the training data. For example, the detection and segmentation tasks in the PASCAL challenge [9] only deal with 20 fixed object categories. The main difference of our work is that we do not require labeled training data. In principle, our method can be applied to videos of any object category.

II. Previous Work

Several methods have been proposed for spatio-temporal segmentation of videos [5], [10]–[14]. Our work is motivated by recent work that uses object annotation for various tasks in video understanding, including human activity recognition [15], event detection [16], and object segmentation [1], [17].

There are increasing interests for object annotation in images and videos. Wang and Mori [6] presented a discriminative latent model for capturing the relationships between image region and textual annotation. 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 to build effective approach for object annotation in Internet videos.

Our proposed method is also inspired by some work on tracking humans [18], [19] or animals [20] by learning video-specific appearance models. For example, the human kinematic tracking system in [19] first detects stylized human poses in a video, then build 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 operationalizes 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. [21] applied multiple-instance learning for scene classification.

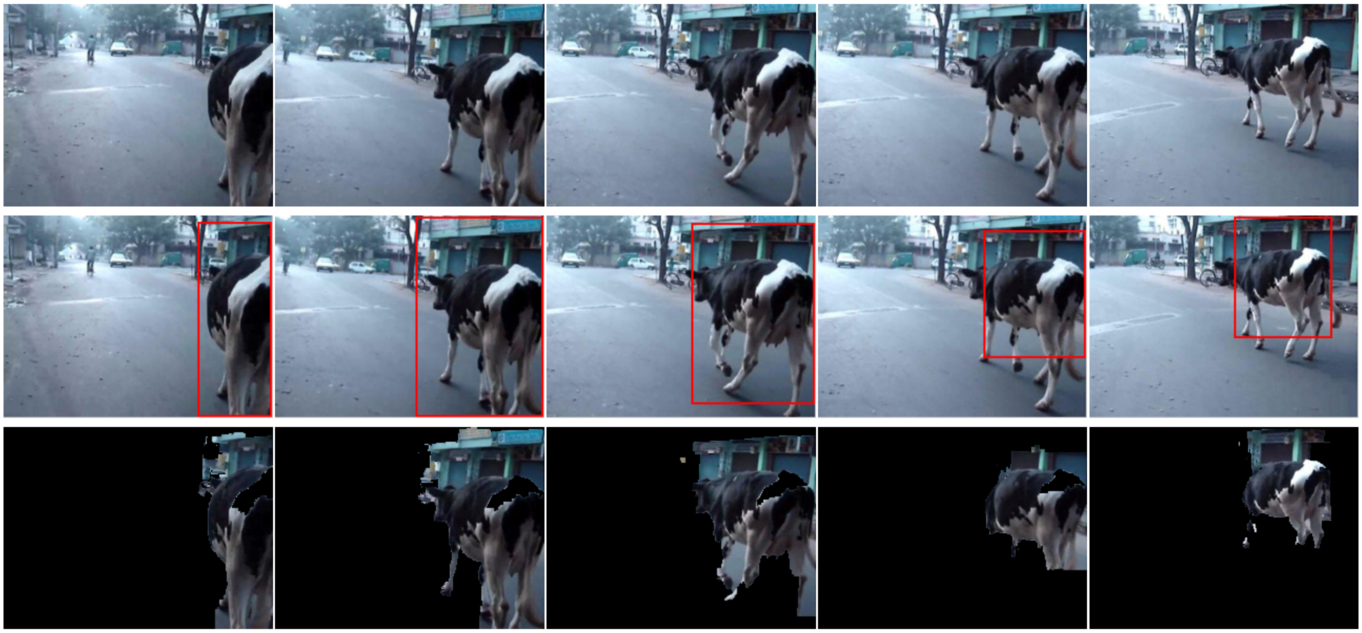

Figure 1. An illustration of our work. Given a video (1st row) with an object tag, e.g. "cow" in this example, our method will automatically localize the object in each frame (2nd row), and segment the object from the background (3rd row).

Recently, multiple-instance learning has been adopted in many computer vision applications, e.g. object detection [7], image annotation [6], etc.

III. Our Approach

The input of our method is a video with an object tag, e.g. "cow". In our work, we will 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 usually make similar assumptions. It is still an open question on how to handle videos with complex scenes and multiple object instances. We will leave it for future work.

Based on these assumptions, our proposed approach involves three 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 are generic and are 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 heuristic 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) Segmenting objects: In the third step, we use the appearance model to re-score object proposals obtained from the first step. Note that in the first step, a bounding box will have a high score if it is likely to contain *any* object. After re-scoring, 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". For each frame, we select the one bounding box according to this re-scoring. We consider this bounding box to be location of the object. Finally, the GrabCut [22] algorithm is applied on the bounding box to segment the object from the background.

We describe the details of each step in the following.

A. Generating Object Proposals

Given an input video, the first step of our approaches 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 detectors, 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

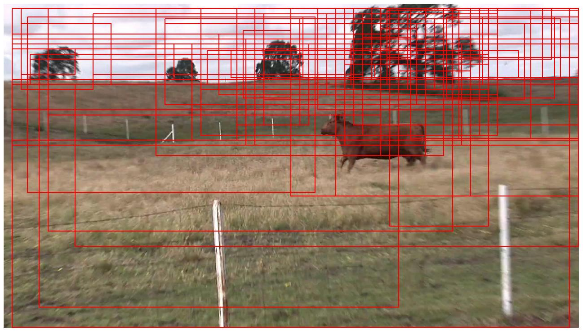

Figure 2. An example of generating object proposals. Given an frame in a video, the objectness in [23] is applied. It returns a collection of bounding boxes in the image that are likely to be *any* objects. For each bounding box, the objectness also assigns a score indicating how likely it is to be an object.

in segmenting objects in a video regardless of the object class, we choose not to use object detectors.

Instead, we use the objectness measurement proposed in [23]. The objectness is a generic measurement that quantifies how likely an image window to be an object of *any* category. Since the objectness is not restricted to any specific object classes, it fits our requirement. Alexe et al. [23] observe that objects (regardless categories) usually have one of the three characteristics: closed boundary, different appearance from the background, being visually salient. The objectness [23] defines several visual cues to capture these observations, including multi-scale saliency, color contrast, edge density, etc. These cues are combined together using a machine learning method. Given a new image, we can apply the learned objectness model on densely sampled image patches of this image. Those patches with high objectness measurements are considered as candidate object proposals. Figure 2 shows an example of generating object proposals on an image.

B. Building Object Appearance Model

Given a video, the objectness approach (see Section III-A) gives us a collection bounding boxes. Those bounding boxes correspond to image windows that are likely to contain *any* objects. However, since objectness is a generic measurement 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 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 object proposals returned by the objectness method. Ideally, the bounding boxes being selected will correspond to the object of interest in the video.

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.

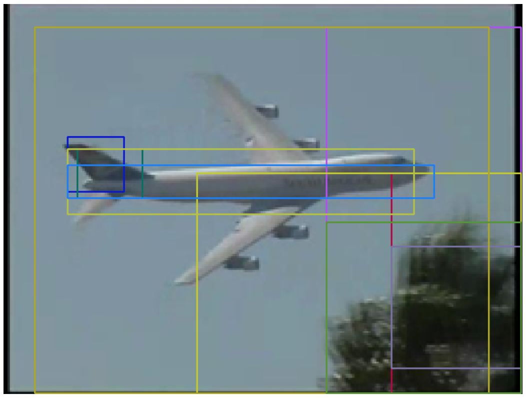

Figure 4. An example of top scored bounding boxes on a frame. We only show 10 out of the 100 selected bounding boxes in order to make the visualization less cluttered.

Note that since our goal is to build 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 for this object. In our 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. Figure 4 shows the examples of selected bounding boxes in a video.

We then build an color-based appearance model for the object. For each selected bounding box, we calculate the normalized color histogram. The appearance model of the object is simply the average of the color histograms from all selected bounding boxes.

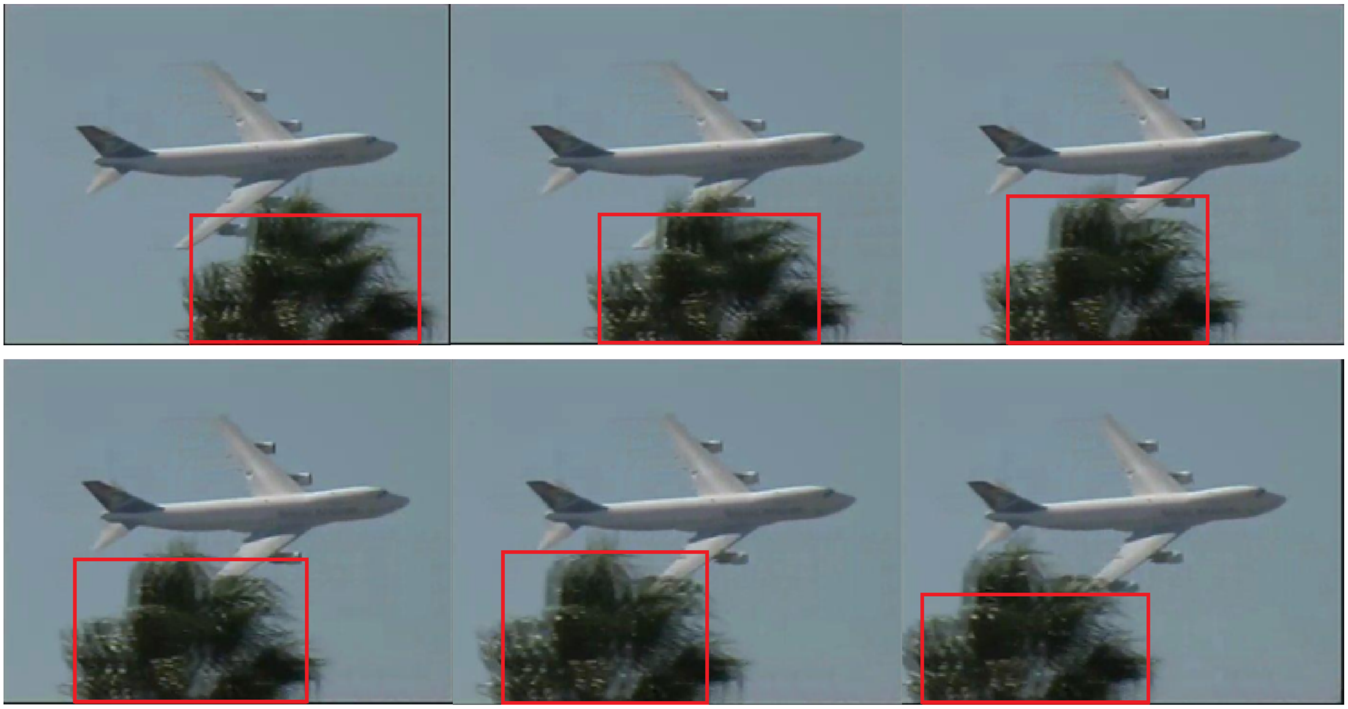

Figure 3. Example of high scored bounding boxes on an image that do not correspond to the object of interest (aeroplane).

C. Segmenting Objects

We now have an appearance model specifically tuned to the object in the given video. We then apply this appearance model to re-evaluate each object proposal returned by the objectness (see Section III-A). For each object proposal, we compute its color histogram, then compute the Euclidean distance to the object appearance model obtained from Section III-B. A small distance means that the bounding box is more likely to contain the *object of interest*. For each frame, we pick one bounding box with the minimum Euclidean distance.

Finally, we apply GrabCut [22] to segment out the object in each frame. GrabCut is an efficient algorithm for foreground segmentation in images. The 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 interactions. We can simply consider the one bounding box (with the minimum Euclidean distance) selected for each frame as the user input. Figure 5 illustrates the pipeline of our approach.

IV. Experiments

In this section, we first describe the dataset and evaluation metrics (Sec. IV-A). We then present our experimental results (Sec. IV-B).

A. 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, etc. Each frame of the video shot is annotated with the segmentation of the object of interest in the video. Table I shows the summary of this dataset. We use 150 video shots with a total of 25,761 frames in our experiments.

Table I
Summary of the dataset used in the experiments.

Class	Number of Shots	Number of Frames
Aeroplane	9	1423
Bird	6	1206
Boat	17	2779
Card	8	601
Cat	18	4794
Cow	20	2978
Dog	27	3803
Horse	17	3990
Motorbike	10	827
Train	18	3270
Total	150	25671

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 frame, let A be the foreground pixels returned by our method, B by the ground-truth foreground pixels provided by the annotation of the dataset. We measure the quality of

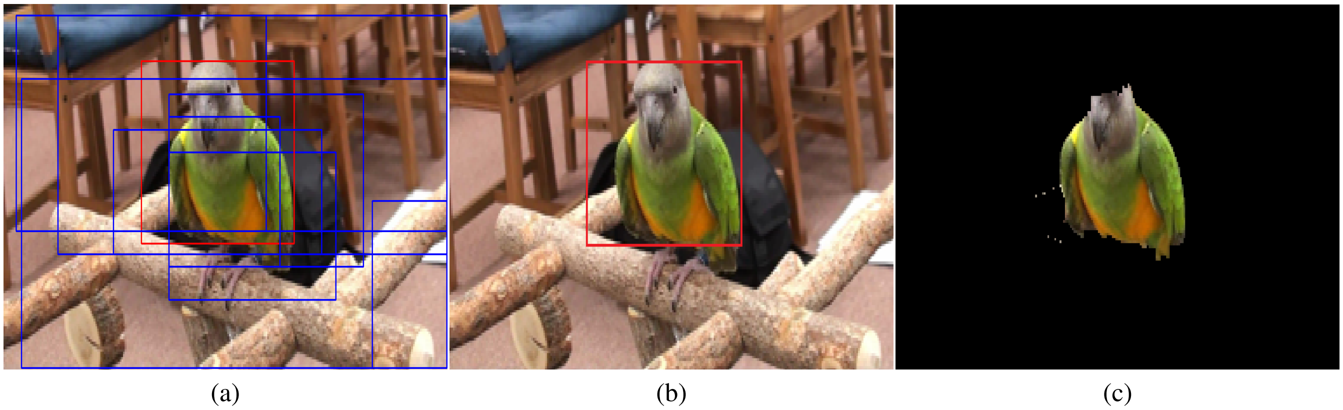

Figure 5. An illustration of our approach. (a) A frame in the video with selected bounding boxes (see Sec. III-B). An appearance model is built based on the selected bounding boxes from all frames of this video. (b) After applying the appearance model on this frame, we obtain a single bounding box that 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.

A by the ratio of $|A \cap B|$ and $|A \cup B|$:

$$r = |A \cap B| \; / \; |A \cup B| \qquad (1)$$

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

B. Results

In order to quantify our proposed approach, we first define a baseline method to compare with. For a given video, the baseline approach simply chooses the bounding box with the highest objectness score on each frame, then performs a GrabCut segmentation using the selected bounding box.

Figure 6 shows the comparison of the accuracies of our approach and the baseline method videos from each of the 10 object categories. We can see that our approach outperforms the baseline method in most of the object categories. Qualitative results of our approach on these 10 object classes are shown in Fig. 7 and Fig 8.

V. Conclusion and Future Work

We have introduced a new approach for segmenting the object of interest in a weakly labeled video. Our approach is fully automatic and does not require any user interactions. Our approach is based on two main observations. First, the main object in a video tend to be salient (i.e. object-like). Second, the object appearance does not change across different frames in a video.

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 as a starting point towards the grand goal of understanding contents of online videos (e.g. Youtube).

Acknowledgment

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, and 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, and 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, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3282–3289.

[4] K. Tang, L. Fei-Fei, and 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, and 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 and G. Mori, "A discriminative latent model of image region and object tag correspondence," in *Advances in Neural Information Processing Systems*, 2010, pp. 2397–2405.

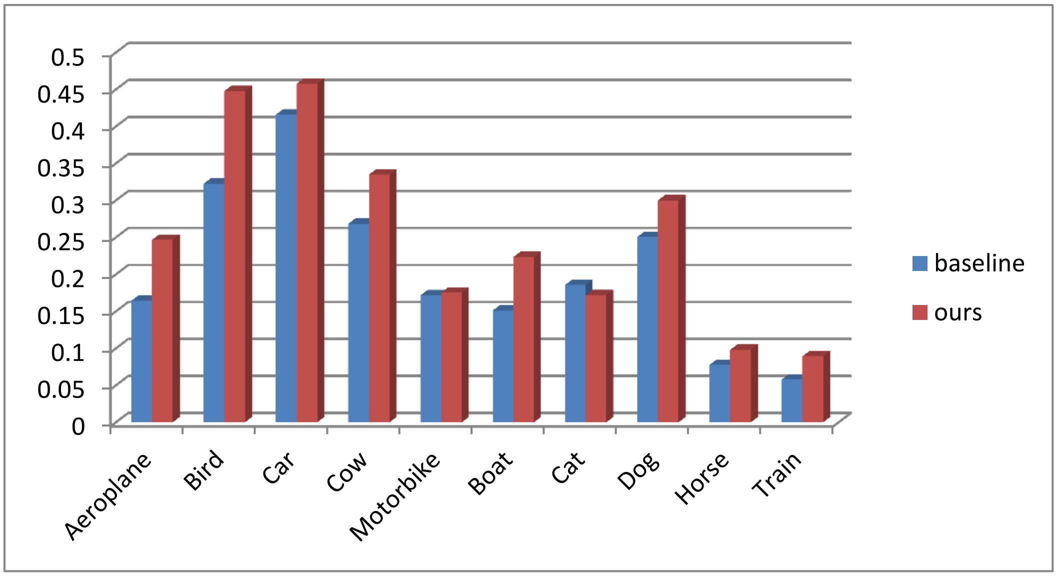

Figure 6. Quantitative comparison between our method and the baseline. For each object class, we compare the accuracy number of correctly segmenting the frames. A frame is considered to be correctly segmented if ratio of intersection over union defined in Eq. 1 is greater than 50%.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1672–1645, 2010.

[8] J. Shotton, J. Winn, C. Rother, and 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, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

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

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

[12] J. Lezama, K. Alahari, J. Sivic, and 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.

[13] J. Xiao and M. Shah, "Motion layer extraction in the presence of occlusion using graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1644–1659, 2005.

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

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

[16] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

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

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

[19] D. Ramanan, D. A. Forsyth, and 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.

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

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

[22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[23] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

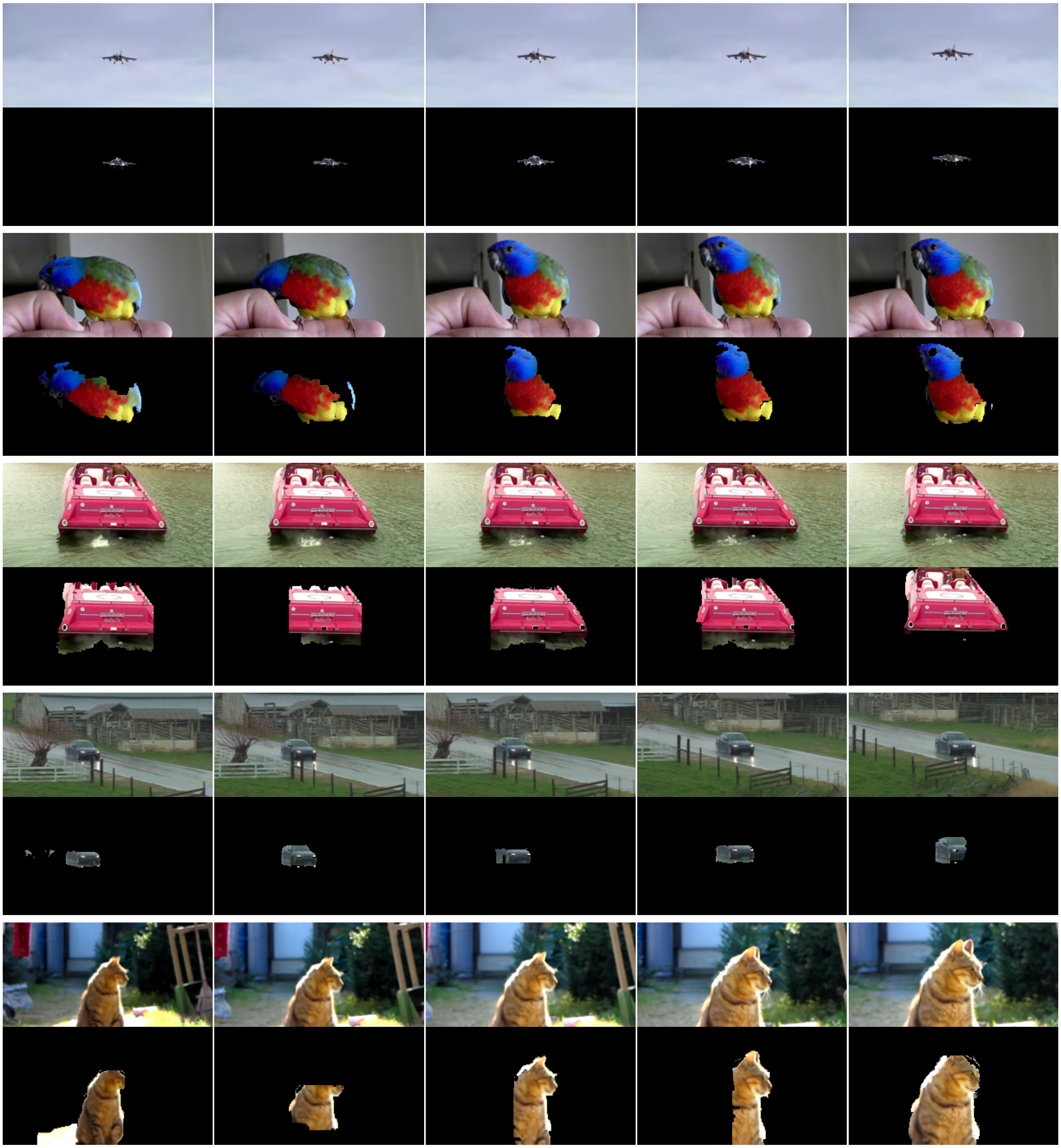

Figure 7. Example results on videos tagged as (from top to bottom) “aeroplane”, “bird”, “boat”, “car”, and “cat”, respectively. For each video, we show the original frames (1st row) and the segmentation results (2nd row).

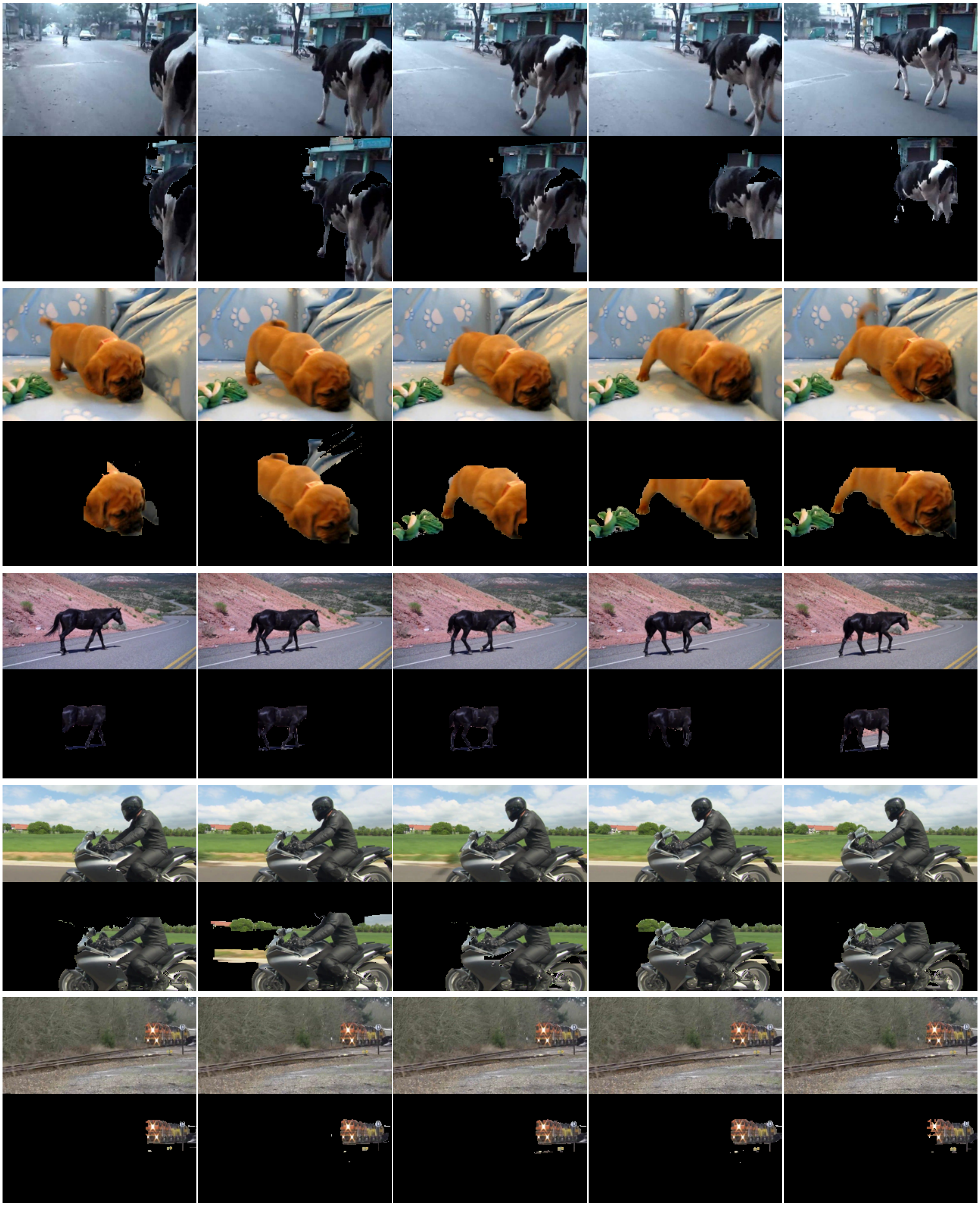

Figure 8. Example results on videos tagged as (from top to bottom) “cow”, “dog”, “horse”, “motorbike”, and “train”, respectively. For each video, we show the original frames (1st row) and the segmentation results (2nd row).