Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computational Engineering and Technical Physics

Master's Thesis

Huiling Wang

DEEP CONVOLUTIONAL NEURAL NETWORKS FOR SEMANTIC VIDEO OBJECT SEGMENTATION

Examiner: Prof. Lasse Lensu
Assoc. Prof. Arto Kaarna

Supervisor: Asst. Prof. Tapani Raiko
Prof. Lasse Lensu

ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computational Engineering and Technical Physics

Huiling Wang

Deep Convolutional Neural Networks for Semantic Video Object Segmentation

Master's Thesis

2016

77 pages, 47 figures, and 2 tables.

Examiner: Prof. Lasse Lensu
Assoc. Prof. Arto Kaarna

Keywords: deep learning, convolutional neural networks, video object segmentation, domain adaptation

In this thesis, we propose to infer pixel-level labelling in video by utilising only object category information, exploiting the intrinsic structure of video data. Our motivation is the observation that image-level labels are much more easily to be acquired than pixel-level labels, and it is natural to find a link between the image level recognition and pixel level classification in video data, which would transfer learned recognition models from one domain to the other one. To this end, this thesis proposes two domain adaptation approaches to adapt the deep convolutional neural network (CNN) image recognition model trained from labelled image data to the target domain exploiting both semantic evidence learned from CNN, and the intrinsic structures of unlabelled video data. Our proposed approaches explicitly model and compensate for the domain adaptation from the source domain to the target domain which in turn underpins a robust semantic object segmentation method for natural videos. We demonstrate the superior performance of our methods by presenting extensive evaluations on challenging datasets comparing with the state-of-the-art methods.

PREFACE

This thesis consists of the work I have developed as my final research project toward a Master's degree under the supervision of Asst. Prof. Tapani Raiko and Prof. Lasse Lensu. My sincerest gratitude goes to them.

I would like to thank Harri Valpola, Antti Rasmus, Miquel Perelló Nieto, Vikram Kamath, Mudassar Abbas and Mathias Berglund, for the comments and discussions related to this thesis.

Finally, thanks to Curious AI Company and NVIDIA Deep Learning Applied Research team in Helsinki for hosting the talk on this thesis project.

Tampere, June 17, 2016

Huiling Wang

CONTENTS

ABBREVIATIONS

CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
DAG	Directed Acyclic Graph
DPM	Deformable Part Models
FCN	Fully Convolutional Networks
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
IoU	Intersection-over-Union
MLP	Multilayer Perceptrons
OOI	Objects of Interest
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks

1 Introduction

1.1 Background

Recent years have witnessed the proliferation of digital imaging devices. Massive amounts of visual data are being produced by mobile phones, consumer cameras, surveillance cameras and other commodity imaging devices. This motivates the development of autonomous systems to semantically analyse and process the explosively growing visual data — the goal of computer vision research. One of the central problems in computer vision is semantic object segmentation, the task of assigning pre-defined object class labels to pixels in images or videos.

Semantic video object segmentation poses higher challenges than its image counterpart, in terms of the huge volume of information and the requirement of spatio-temporal coherence in labelling. Yet fully autonomous approach remains advantageous in application scenarios where the human in the loop is expensive or impractical, such as video recognition or summarisation, 3D reconstruction and background replacement.

While effortless for humans, semantic video object segmentation can be challenging for machines owing to several reasons. Firstly, acquiring the prior knowledge about object is difficult; gaining pixel-level annotation for training supervised learning algorithms is prohibitively expensive comparing with image-level labelling. Secondly, background clutters and object appearance variations due to scale, translation, rotations, and illumination effects introduce visual ambiguities that in turn cause mis-segmentations. Thirdly, camera motion as well as occlusions bring geometry ambiguities to consistent visual analysis. Recent years have seen encouraging progress, particularly in terms of generic object segmentation [1–8] which segments video foreground objects regardless of semantic labels, and the success of deep learning, especially convolutional neural networks, in image recognition [9–12] also sheds light on semantic video object segmentation.

Generic object segmentation methods largely leverage generic object detection, i.e., category independent region proposal methods [13–15], to capture object-level description of the generic object in the scene incorporating motion cues. These approaches address the challenge of visual ambiguities to some extent, seeking the weak prior knowledge of what the object may look like and where it might be located. However, there are generally two major issues with these approaches. Firstly, the generic detection is virtually ranking and proposing hundreds to thousands of coherent segments at various scales, typically based

on hierarchical segmentation; thus it has very limited capability to determine the presence of an object. Secondly, such approaches are generally unable to determine and differentiate unique multiple objects, regardless of categories. These two bottlenecks limit these approaches to segmenting one single object or all foreground objects regardless classes or identifies.

The deep convolutional neural networks (CNNs) have been immensely successful in various high-level tasks in computer vision such as image level recognition [9–11] and bounding box level object detection [16]. However, stretching this success to pixel-level classification or labelling, i.e., semantic segmentation, is not naturally straightforward. This is not only owing to the difficulties of collecting pixel-level annotations, but also due to the nature of large receptive fields of convolutional neural networks. Furthermore, due to the aforementioned challenges present in video data, CNNs need to learn a spatio-temporal representation of the video in question in order to give a coherent segmentation.

1.2 Contributions

The core contribution of this thesis is to develop a novel method which transcends the generic object segmentation paradigm and achieves semantic object segmentation by harnessing deep convolutional neural networks. The main insight is to bridge the gap between image classification and object segmentation, leveraging the ample image annotations and good discriminative pre-training. This is interesting considering that one might be able to circumvent the necessity of using the expensive pixel-level annotation datasets and use only image-level recognition model. This goal is achieved by proposing two domain adaptation approaches, exploiting the image recognition model from the source domain, and the intrinsic structure of video data in the target domain, forming a visual representation which captures the synergy of the same object instances in deep feature space from continuous frames. Our proposed approaches explicitly model and compensate for the domain adaptation from the source domain to the target domain which in turn underpins a robust semantic object segmentation method in natural videos. In investigating these goals, the thesis focuses on developing an autonomous method to semantically extract objects of known categories out of natural video, which has been weakly labelled with a semantic concept. An overview of our proposed method is shown in Fig. 1, which consists of object discovery, domain adaptations and object segmentation components, bridging the gap between the source domain of image recognition and the target domain of semantic object segmentation.

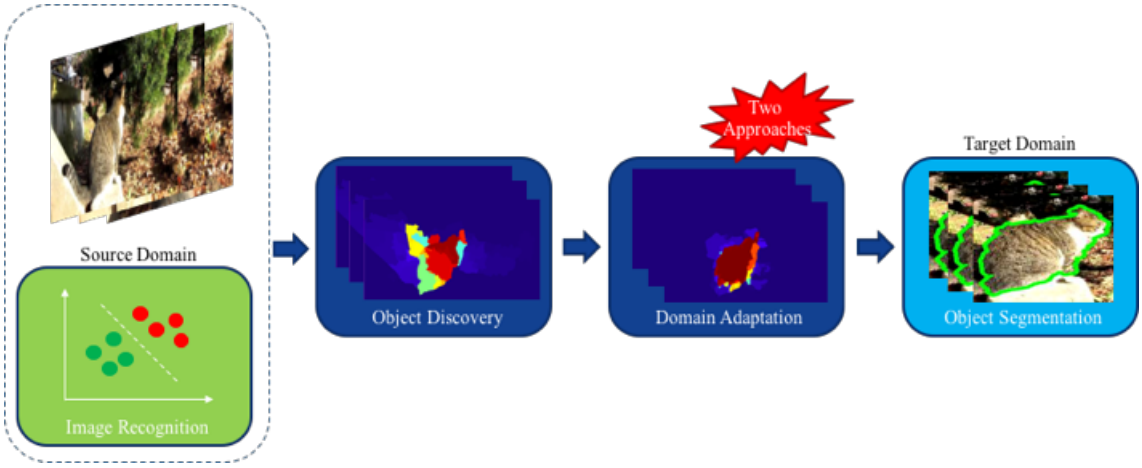


Figure 1. Overview of our proposed method. Our system takes video frames (top left) as input and applies pre-trained image recognition model and generic object detector for object discovery. Two domain adaptation approaches are proposed to bridging the gap between the source domain of image recognition and the target domain of semantic object segmentation. The adapted semantic confidence maps are employed to achieve robust semantic object segmentation on consecutive video frames (right).

1.3 Structure

This thesis is structured as follows. We first give a literature review of related work in Sec. 2, forming observations on the previous works. In Sec. 3, we introduce our approach to discovering weakly labelled objects of interest (OOI) in video. Sec. 4 describes the two approaches of domain adaptation, which results in semantic evidence which underpins a robust semantic object segmentation in Sec. 5. Sec. 6 evaluates the proposed method on benchmark datasets comprising challenging video clips exhibiting various challenges, comparing against state-of-the-art algorithms. Sec. 7 summarises the contributions of the thesis.

2 Semantic Video Object Segmentation

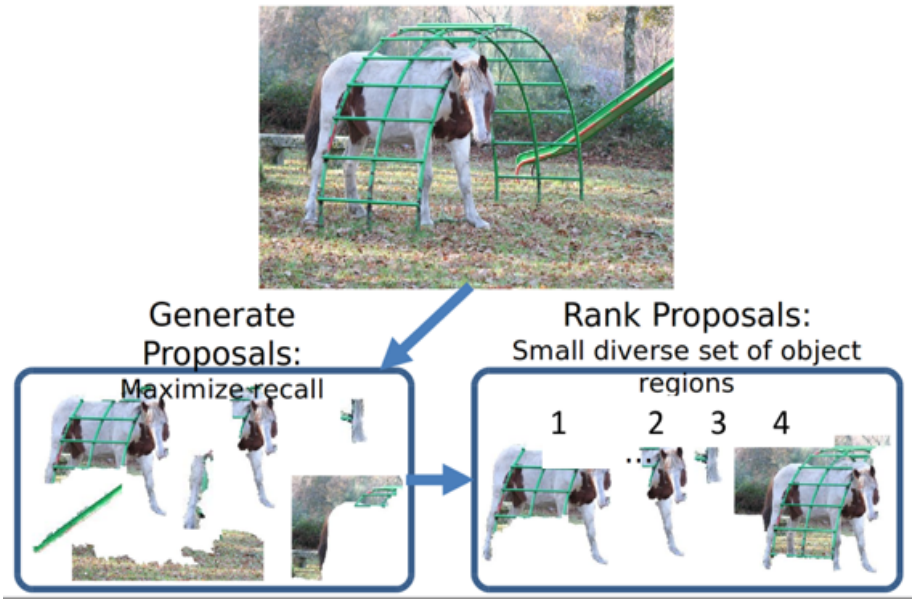
In this section we review the related works and clarify the relevance of the proposed approaches to semantic video object segmentation within the context of the related works.

This literature review presents the technical background for the thesis by introducing the notations and reviewing the related key techniques, without intending to give a thorough and complete survey for each related area.

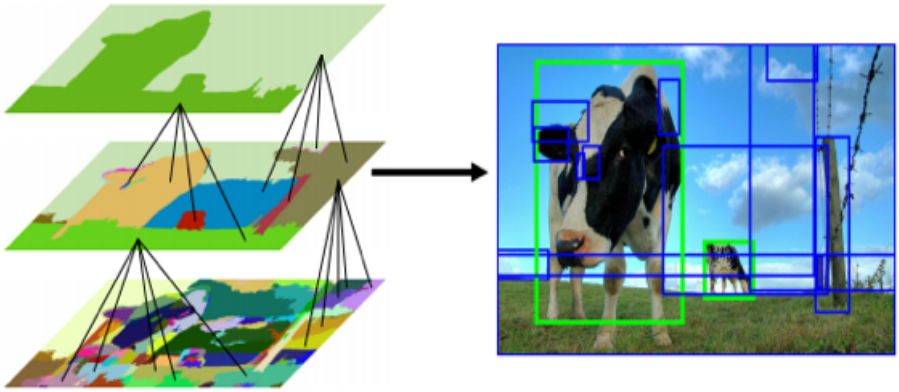
2.1 Generic Object Proposal

Generic object detection learns generic properties of objects from a set of examples, and proposes segment regions which may contain generic objects. Generic object detection has attracted a lot of attentions in context of still images recently [13–15, 17–20]. The objectness measure was initially introduced by Alexe *et al.* [17], which used Bayesian classifier applied on multiple cues to compute the probability that a bounding box contains generic object. Endres and Hoiem [13] generated multiple figure-ground segmentation using conditional random fields (CRF) with random seeds and learned to score each segment based multiple cues as shown in Fig. 2a. Similarly, constrained parametric min-cut (CPMC) method [18] also generated multiple segments and ranked proposals according to a learned scoring function. Selective Search [14] applied a hierarchical agglomeration of regions in a bottom-up manner as shown in Fig. 2b. Manen *et al.* [15] constructed a connectivity graph based on superpixels with edge weights as the probability of neighbouring superpixels belonging to the same object, and generated random partial spanning trees with large expected sum of edge weights. Arbeláez *et al.* [20] employed a hierarchical segmenter and proposed a grouping strategy that combined regions at multi-scales into object hypotheses by exploring efficiently their combinatorial space. Cheng *et al.* [19] introduced method to compute the objectness score of each bounding box at various scale and aspect ratio.

Generic object proposal methods incorporating temporal information in video data have also been investigated [21–23]. Approach of extracting region proposals from each frames independently which were linked across temporal domain into object hypotheses has been proposed by Tuytelaars [23]. This approach suffers from the mis-segmentations from independent frames which was addressed by Oneata *et al.* [21] by proposing a supervoxel method to incorporate temporal information during the initial segmentation step,



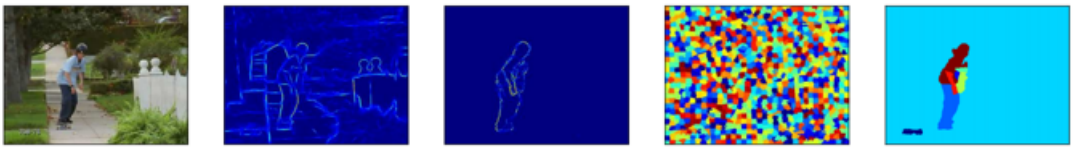
(a)



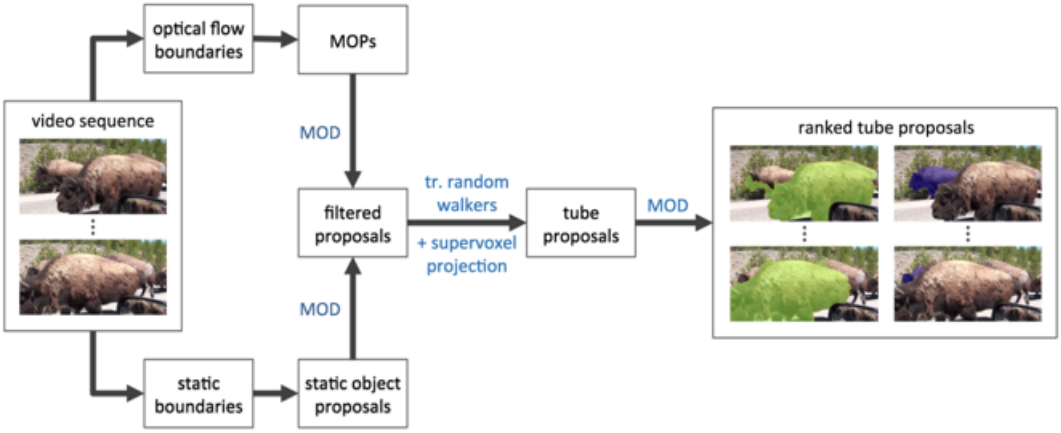
(b)

Figure 2. Key image based generic object proposal methods in the literature: (a) region based approach by [13] which generates multiple figure-ground segmentation using CRF with random seeds and learns to score each segment based multiple cues. (b) bounding box based approach by [14] which applies a hierarchical agglomeration of regions in a bottom-up manner and returns bounding boxes with scores.

as shown in Fig. 3a. Despite of the improvements in quality, supervoxel based methods typically become computationally infeasible for longer videos, and prone to over-segment fast moving objects. Fragkiadaki *et al.* [22] generate region proposals based on motion boundaries per frame which are ranked by a trained moving objectness detector. The top ranked segments are extended into spatio-temporal voxels utilising random walkers on affinities of trajectories formed by dense points, as shown in Fig. 3b.



(a)



(b)

Figure 3. Key video based generic object proposal methods in the literature: (a) supervoxel approach by [21]; from left to right: video frame, detected edges, flow boundaries, superpixels, and hierarchical clustering result at the level with eight supervoxels. (b) motion boundaries approach by [22]; initially a set of region proposals are generated in each frame using multiple segmentations on optical flow and static boundaries, which is called per frame Moving Object proposals (MOPs) and static proposals. A Moving Objectness Detector (MOD) then rejects proposals on static background or obvious under or over segmentations. The filtered proposals are extended into spatio-temporal pixel tubes using dense point trajectories. Finally, tubes are ranked by MOD using score aggregation across their lifespans.

2.2 Optical Flow

The goal of optical flow estimation is to compute an approximation to the motion field from time-varying image intensity. Optical flow estimation has been dominated by variational approaches since Horn and Schunck [24]. Recent works have been focusing on large displacement optical flow methods which integrated combinatorial matching into the traditional variational approaches [25–29]. Convolutional neural networks have been used by DeepMatching and DeepFlow [27] to aggregate information in a fine-to-coarse manner with all parameters are set manually. The problem with [27] is that the matches are simply interpolated to dense flow fields, which was addressed by EpicFlow [28] which improved on the quality of sparse matching. FlowNet [29], as shown in Fig. 4, used CNN for the flow field prediction without requiring any hand-crafted methods for aggregation, matching and interpolation.

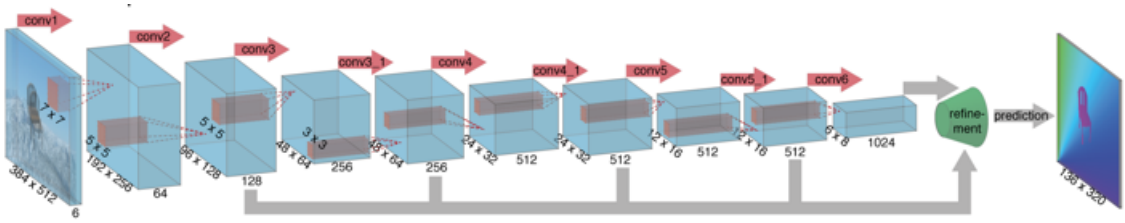


Figure 4. Network architecture of FlowNet [29] with w, h, and c being their width, height and number of channels at each layer. FlowNet uses CNN for the flow field prediction without requiring any hand-crafted methods for aggregation, matching and interpolation.

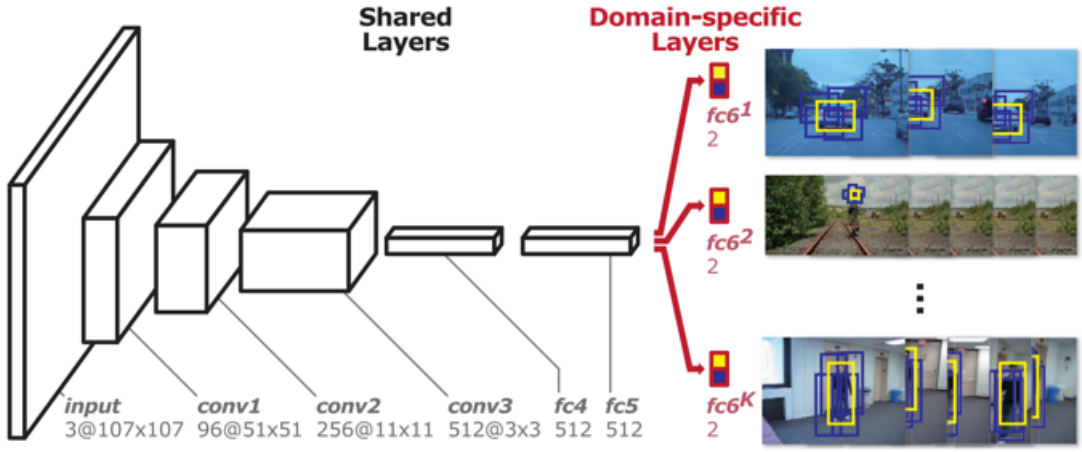


Figure 5. Network architecture of the CNN based visual tracker [37], where for each layer $c@w \times h$, notations w, h, and c represent the number of channels, width and height at each layer; 2 indicates each $fc6$ layer contains a binary classification layer. This method pre-trains a CNN using a large set of videos with tracking ground truths to obtain a generic target representation.

2.3 Object Tracking

Object tracking is the process of locating one or multiple moving objects over time using a camera, which has many practical applications (e.g. surveillance, HCI) and has long been studied in computer vision. Due to the computational efficiency and competitive performance, correlation filter based approaches [30–33] have gained attention in the area of visual tracking in recent years. Deep learning based methods [34–38] have also been developed. [34] proposed an online method based on a pool of CNNs, which suffers from lack of training data to train deep networks. [35, 36] transferred CNNs pre-trained on a large-scale dataset constructed for image classification, however the domain shift was not properly compensated [37]. [38] empirically studied some important properties of CNN features under the viewpoint of visual tracking and proposed a tracking algorithm using fully convolutional networks pre-trained on the image classification task. [37] took a different approach and pre-trained a CNN using a large set of videos with tracking ground truths to obtain a generic target representation, as shown in Fig. 5.

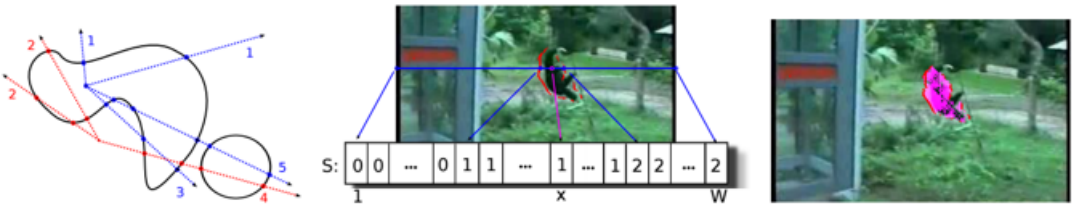
2.4 Video Object Segmentation

Video object segmentation is the problem of automatically segmenting the objects in a video. The majority of research efforts in video object segmentation can be categorised into three groups (semi-)supervised, unsupervised and weakly supervised methods, based on the level of automations.

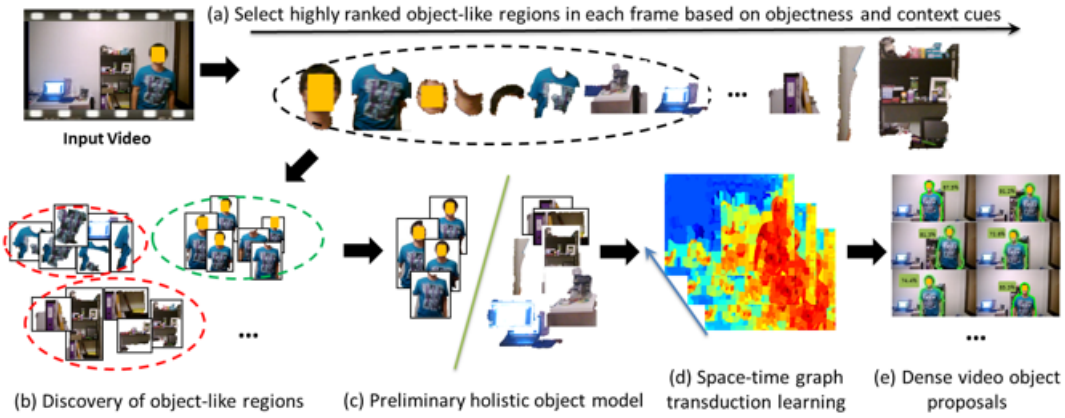
Methods in the first category normally require an initial labelling of the first frame, which either perform spatio-temporal grouping [39, 40] or propagate the labelling to guide the segmentation in consecutive frames [41–44].

Autonomous methods have been proposed due to the prohibitive cost of human-in-the-loop operations when processing ever-growing large-scale video data. Bottom-up approaches [6, 45, 46] largely utilise spatio-temporal appearance and motion constraints, while motion segmentation approaches [47, 48] perform long-term motion analysis to cluster pixels or regions in video data. Giordano *et al.* [49] extended [6] by introducing 'perceptual organization' to improve segmentation performance. Taylor *et al.* [50] inferred object segmentation through long-term occlusion relations, and introduced a numerical scheme to perform partition directly on pixel grid. Wang *et al.* [51] exploited saliency measure using geodesic distance to build global appearance models. Several methods [3, 5, 7, 8, 52] propose to introduce a explicit notion of object by exploring those recurring region proposals from still images by measuring appearance and motion cues of generic objects (e.g., [13]) to achieve state-of-the-art results. However, due to the limited recognition capability of generic object detection, these methods normally can only segment foreground objects regardless of semantic label.

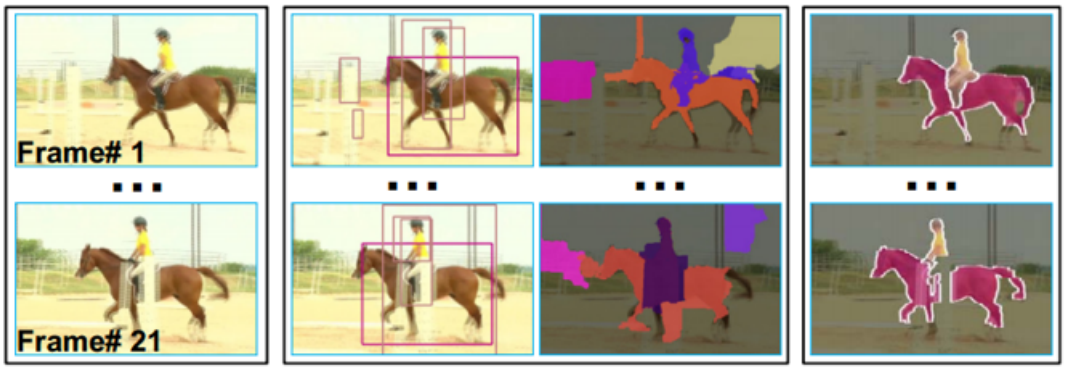
The proliferation of user-uploaded videos which are frequently associated with semantic tags provides a abundant resource for computer vision research. These semantic tags, albeit not spatially or temporally located in the video, suggest semantic concepts present in the video. This social trend has led to an increasing interest in exploring the idea of segmenting video objects with weak supervision or labels. Hartmann *et al.* [53] firstly formulated the problem as learning a set of weakly supervised classifiers for spatio-temporal segments. Tang *et al.* [54] learned discriminative model by leveraging labelled positive videos and a large set of negative examples based on distance matrix. Liu *et al.* [55] extended the traditional binary classification problem to multi-class and proposed an algorithm of nearest-neighbour-based label transferring which encourages smoothness between spatio-temporally adjacent regions which are similar in appearance. Zhang *et al.* [56] utilised the pre-trained object detector to generate a set of detections and then pruned



(a)



(b)



(c)

Figure 6. Key automatic video object segmentation methods in the literature: (a) motion boundaries approach by [6] which is derived as follows (left) the ray-casting method given the fact that any ray originating outside intersects it an even number of times; (middle) illustration of the integral intersections data structure for the horizontal direction to speed up the ray-casting; (right) the output inside-outside biary map. (b) framework by [7, 8] to rank, cluster and learn a holistic model from noisy object regions and generate consistent object proposals through graph transduction learning. (c) still-image object detection approach by [56] which follows the following steps (left) the input video is weakly labelled with semantic tags, making it difficult to locate and segment the desired objects; (middle) per-frame detection and segmentation proposals provide location information but are often very noisy; (right) the proposed segmentation-by-detection framework can generate consistent object segmentation results from noisy detection and segmentation proposals.

noisy detections and regions by preserving spatio-temporal constraints.

2.5 Convolutional Neural Networks

Convolutional neural networks (LeCun *et al.* [57]) are biologically-inspired variants of multilayer perceptrons (MLPs), which belong to the family of deep learning, mapping a set of observations to a set of targets via multiple non-linear transformations. We first briefly introduce general concepts of deep learning and then focus on CNNs in this section.

Hubel and Wiesel's early work [58] suggested a layered structure in the mammalian visual cortex, which has inspired the formulation of a few computational architectures aimed at emulating the brain. Arguably the most successful of those formulations is the introduction of neural networks [59]. A neural network consists of multiple layers of "neurons", where the output of a neuron can be the input of another. This thesis will use the terms "neurons" and "units" interchangeably. Each neuron is a computational unit which takes in weighted inputs and produces the output according to its associated activation function. A simple three-layer neural network is illustrated in Fig. 7.

Deep learning algorithms employ multiple layer representations to transform data into high level concepts through a hierarchical learning process. The "deep" networks typically consist of multiple layers of non-linear transformations from input to output, gradually deriving higher level feature from the lower level features, leading to a hierarchical representation. This multiple levels representation is highly motivated by nature.

One of the earliest neural networks developed for computer vision task was the Neocognitron [61], which extracted local features of the input in a lower stage and gradually aggregated local features into global features. Each neuron of the highest stage aggregates the information of the input, and responds only to one specific pattern. The Neocognitron was able to learn to perform simple pattern recognition. However, it lacked a supervised training algorithm. LeCun *et al.* [57, 62] later developed Convolutional neural networks, a similar formulation to Neocognitron, for visual recognition. These convolutional models have proved to be immensely successful in computer vision. The CNN is a feed-forward network, where none of its units forms a directed cycle. Other types of deep networks have also been successfully applied to computer vision, which include the the recurrent neural network (Hochreiter and Schmidhuber [63]) and the deep belief network (Graves and Schmidhuber [64]). CNNs exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers which allows for features to be detected regardless of their position in the visual field. Additionally, weight sharing enables learning efficiency by significantly reducing the number of parameters to be learned.

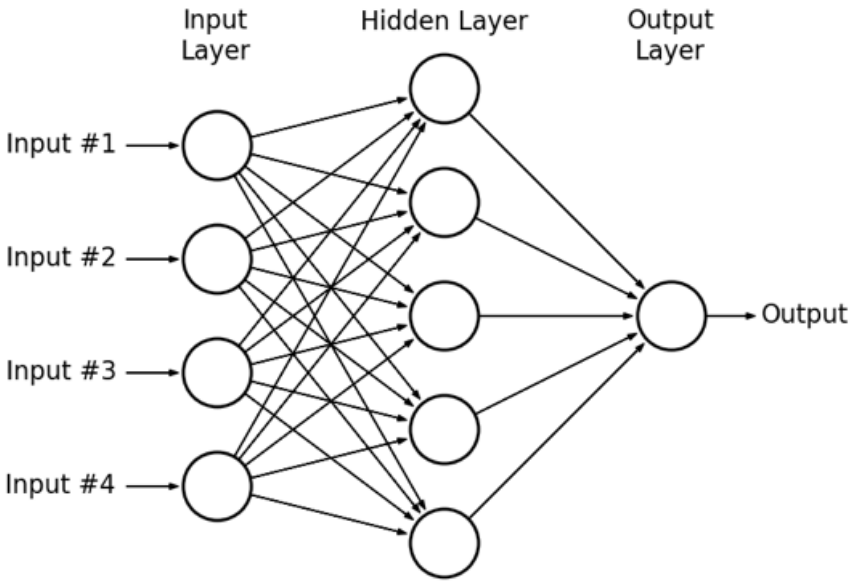


Figure 7. A simple three-layer neural network. Each circle represents a neuron, taking a number of inputs and producing a single output. The middle layer of nodes is referred as the hidden layer, as its values are not observed in the training set.

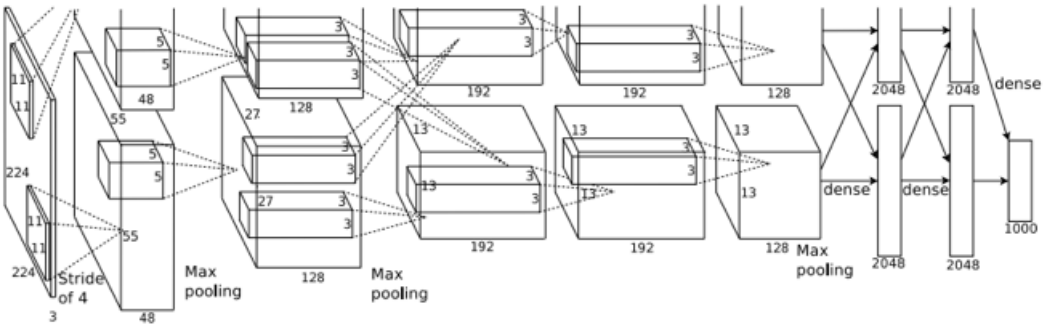


Figure 8. An illustration of the CNN architecture in Krizhevsky *et al.* [60], explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure (partly drawn) while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150, 528-dimensional, and the number of neurons in the network's remaining layers is given by 253, 440-186, 624-64, 896-64, 896-43, 264-4096-4096-1000.

These merits enable CNNs to achieve better generalization on vision problems.

However, due to the limited computing resource and a lack of large datasets, deep neural networks had been prevented being successfully deployed which, however, has changed with the emergence of the large-scale ImageNet dataset and the increasing GPU computing power. Krizhevsky *et al.* [60] successfully trained a CNN utilising GPU parallel

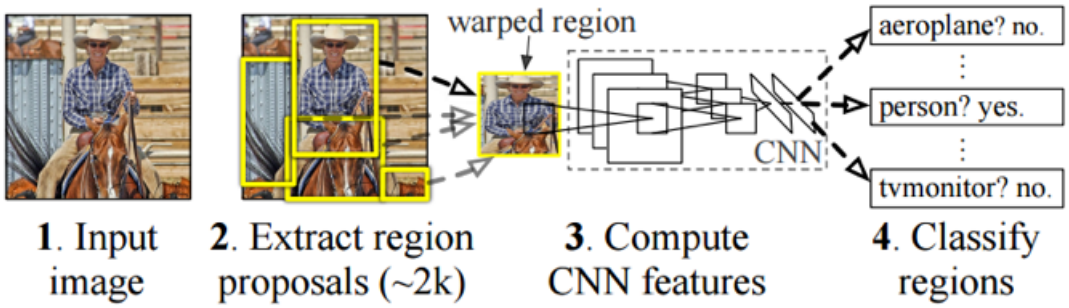


Figure 9. An illustration of R-CNN which combines region proposals with CNN by Girshick *et al.* which fine-tunes pre-trained image recognition model with pixel-level annotations. [16].

computing and achieved a performance leap in image classification problem on the ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012), instantly making CNN the central attention of the computer vision community. The CNN architecture in Krizhevsky *et al.*, often dubbed AlexNet, is illustrated in Fig. 8.

Despite the adoption of modern GPUs, training a Convolutional Network from scratch with randomly initialized parameters for a large scale dataset is still too slow. Furthermore, it is relatively rare to have a dataset of sufficient amount of labelled data to train a CNN of a similar size as in Krizhevsky *et al.* [60]. Alternatively, it is common to use the pre-trained CNN on a very large dataset, e.g., ImageNet, in a transfer learning paradigm. There are normally two transfer learning scenarios using CNN pre-training, i.e., either as fixed feature extractor or initialization for fine-tuning.

Using CNN pre-training as feature extractor normally removes the last fully-connected layers which outputs the class scores for a classification task. It is assumed that bottom convolutional layers correspond to generic image representations whilst the later layers are task-specific. There have been significant improvement over the traditional hand-crafted features on visual recognition tasks, e.g., object detection [16] (as illustrated in Fig. 9), tracking [65], scene recognition [66], action recognition [67].

Fine-tuning the CNN pre-training effectively exploits its representation power for classification purposes. Instead of taking the responds of a mid-layer as feature, one can fine-tune the network for new tasks without the need for a large training dataset. Several works have adopted the fine-tuning approach. Farfade *et al.* [70] fine-tuned AlexNet for face detection, replacing the last layer with a new fully connected layer of two outputs. Ren *et al.* [71] performed training by fine-tuning alternating between the region proposal and object detection. Notably, Long *et al.* [69] converted the fully connected layers of a pre-trained CNN into convolutional layers which accept inputs of arbitrary size and output

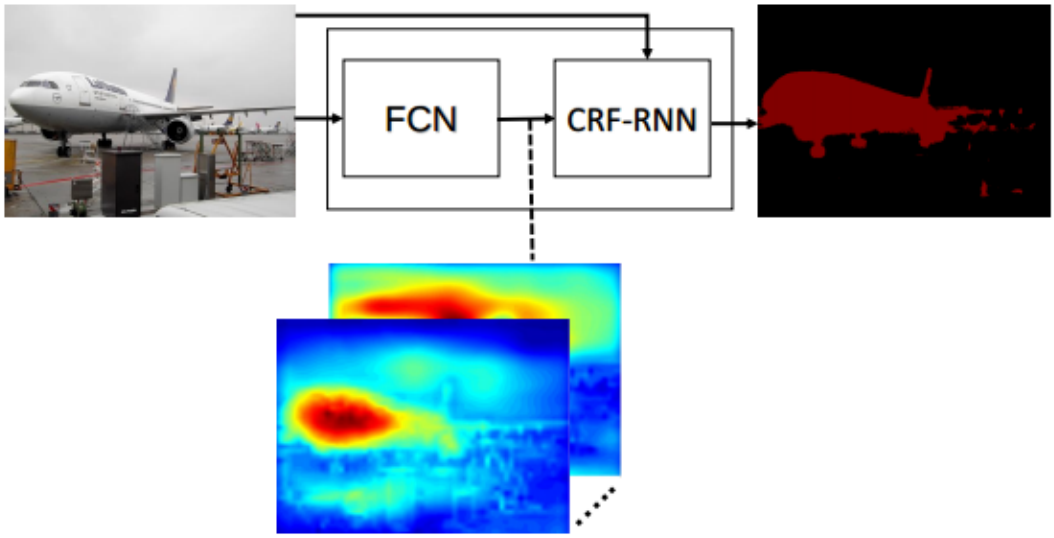


Figure 10. CRF-RNN (Zheng *et al.* [68]) which combines fully convolutional neural network (FCN) (Long *et al.* [69]) and conditional random field (CRF) which is implemented as a RNN architecture.

classification probabilities, with the pre-trained weights as initialization.

Convolutional neural networks are not only advancing image-level classification, they are also driving advances in on local tasks with structured output, e.g., semantic segmentation. Some recent approaches including FCN (Long *et al.* [69]), DeepLab (Chen *et al.* [72]) and CRF-RNN (Zheng *et al.* [68], as illustrated in Fig. 10) have shown significant accuracy boosts by retraining state-of-the-art CNN based image classifiers. One important observation is that all the aforementioned methods require significant amount of pixel-level annotations for retraining CNN models on as less as 20 categories. This motivates us to explore how to effortlessly transfer the success of image recognition on 1000 categories or more to video semantic object segmentation without pixel-level annotations. This motivation shapes the goal of this thesis. Details of related works in semantic image segmentation are presented in Sec. 2.7.

2.6 Unsupervised Visual Representation Learning

Unsupervised learning of visual representations is a well researched area starting from the original auto-encoder work [73]. The most related works to this thesis are those learning feature representations from the videos using deep learning approaches [74–81]. The most common constraint of these works is enforcing learned representation to be temporally smooth. Among these works, Wang and Gupta [79] (Fig. 11) adopted visual tracking

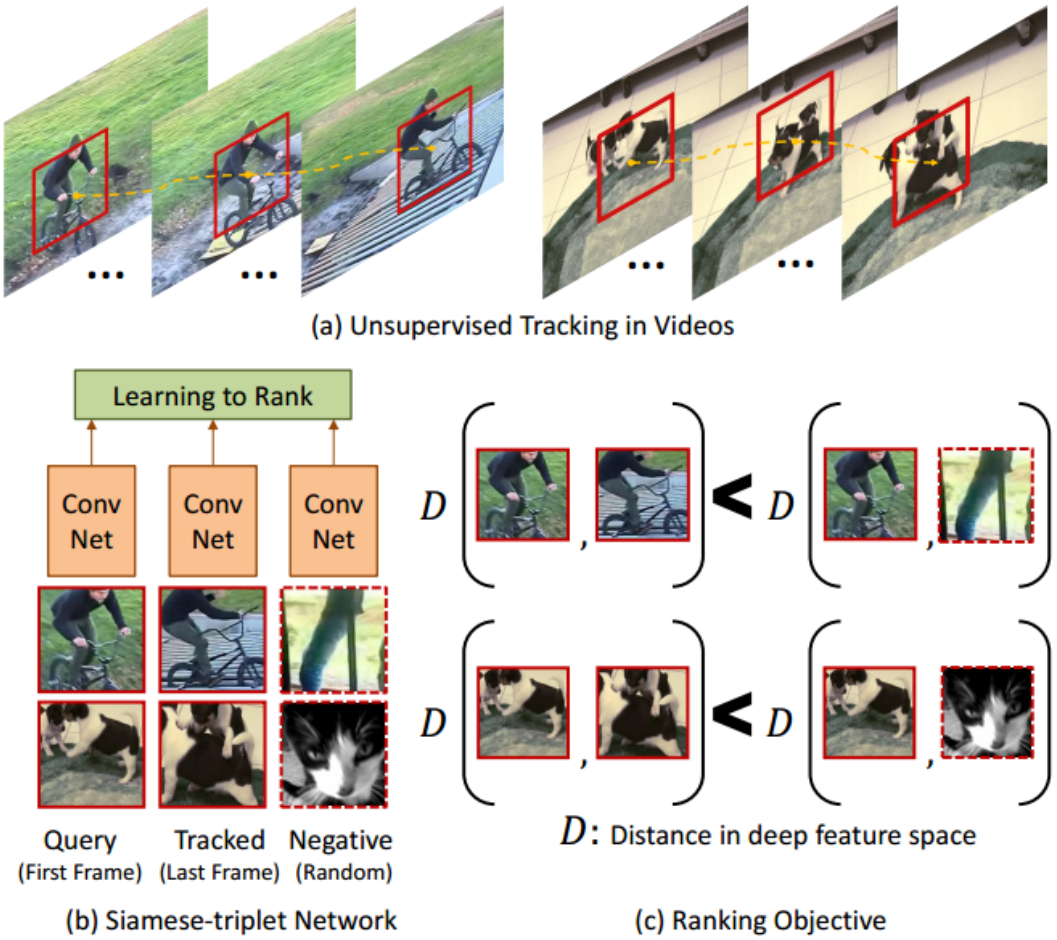


Figure 11. Wang and Gupta [79] adopts visual tracking to capture distinct frames of the same object instance, and they extract fixed-size image patches from the first and last frame with distinct appearances of the same object instance.

to capture frames of the same object instance. Although our method also makes use of tracking for representation learning, we are different from [79] in that they sought after fixed-size image patches from the first and last frame with distinct appearances of the same object instance, whereas we aim to collect region proposals as training instances from multiple stable tracks enforcing temporal coherence.

2.7 Semantic Image Segmentation

In this section, we review the recent developments in semantic image segmentation based on deep learning, especially CNNs. These approaches can mainly be categorised into two strategies.

The first strategy is utilising image segmentation to exploit the middle-level represen-

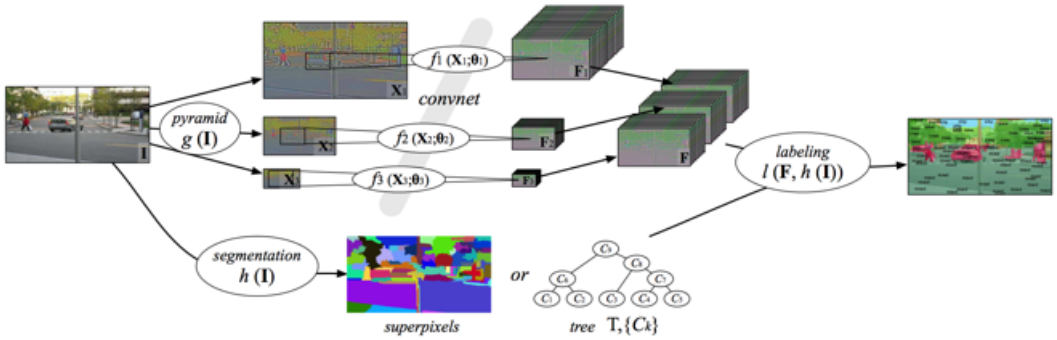


Figure 12. Farabet *et al.* [82] adopts hierarchical feature trained from raw pixels in a multi-scale convolutional network and multiple-scale superpixels to encode local information for semantic labelling.

tations, such as superpixels or region proposals, to account for the structured patterns of images. Farabet *et al.* [82] adopted a multi-scale convolutional network which was trained from raw pixels to extract dense features and utilised multiple-scale superpixels to encode local information for semantic labelling, as illustrated in Fig. 12. Similar approach has also been taken by [83], which converted segmentation as classification by exploiting multi-scale superpixels and multi-layer neural networks. The main advantage of this strategy is that superpixels encode local contextual information and make the inference more efficient, whilst the disadvantage is the error arising from under-segmentation which could not be eliminated in the later stage.

The second strategy is training an end-to-end neural network to model the nonlinear mapping from raw pixels to label map, without relying on any segmentation or task-specific features. Recurrent neural network (RNN) based approach has been proposed by [84] which utilised RNN to capture long range image dependencies. The fully convolutional neural network was proposed by [69] who replaced the last fully connected layers of a CNN by convolutional layers to keep spatial information and introduced multi-scale upsampling layers to alleviate the coarse output problem, as shown in Fig. 13. [68] incorporated CRF as part of an end-to-end trainable network in the form of a recurrent neural network and jointly learned the parameters in one unified deep network.

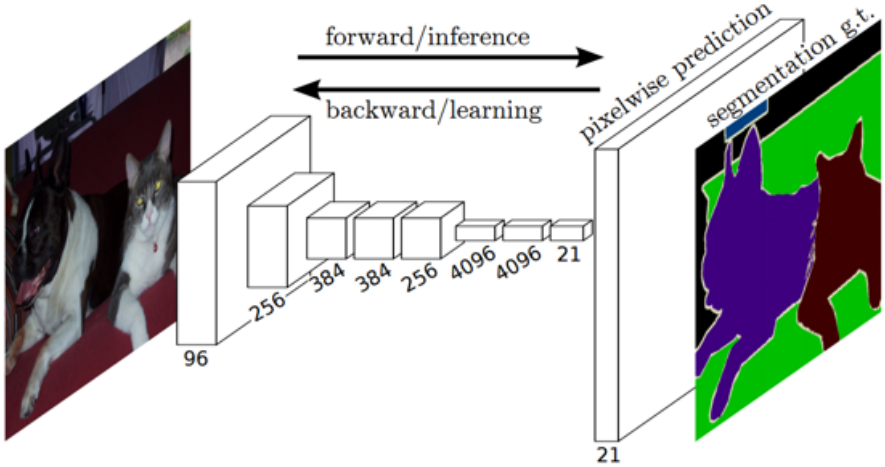


Figure 13. Long *et al.* [69] replaced the last fully connected layers of a CNN by convolutional layers to keep spatial information. FCN accepts inputs of arbitrary size and output classification probabilities.

3 Object Discovery

We set out our approach to first semantically discovering possible objects of interest from video as illustrated in Fig. 1, which consists of the following key steps as illustrated in Fig. 14: (a) generating region proposals (b) scoring and selecting candidate proposals using generic properties (c) classifying proposals with regard to semantic labels applying image recognition model and (d) generating semantic confidence map by aggregating both multiple proposals and spatial distributions. Each step is detailed in the following sections.

3.1 Proposal Scoring

Unlike image classification or object detection, semantic object segmentation requires not only localising objects of interest within an image, but also assigning class label for pixels belonging to the objects. One potential challenge of using image classifier to detect objects is that any regions containing the object or even part of the object might be "correctly" recognised, which results in a large search space to accurately localise the object. To narrow down the search of targeted objects, we adopt bottom-up category-independent object proposals.

In order to produce segmentations, we require region proposals rather than bounding boxes. We consider those regions as candidate object hypotheses. Exemplar region proposals are shown in Fig. 15. The objectness score associated with each proposal from [13] indicates probability that an image region contain an object of any class. However, this objectness does *not* consider context cues and only reflects the generic object appearance of the region. We incorporate motion information as a context cue for video objects. There has been many previous works on estimating local motion cues. We adopt a motion boundary based approach as introduced in [6] which roughly produces a binary map indicating whether each pixel is inside the motion boundary after compensating camera motion.

The motion cue estimation begins by calculating optical flow between consecutive frames. We firstly estimate motion boundaries which identify the location of occlusion boundaries implied by motion discontinuities which might correspond to physical object boundaries, following [6].

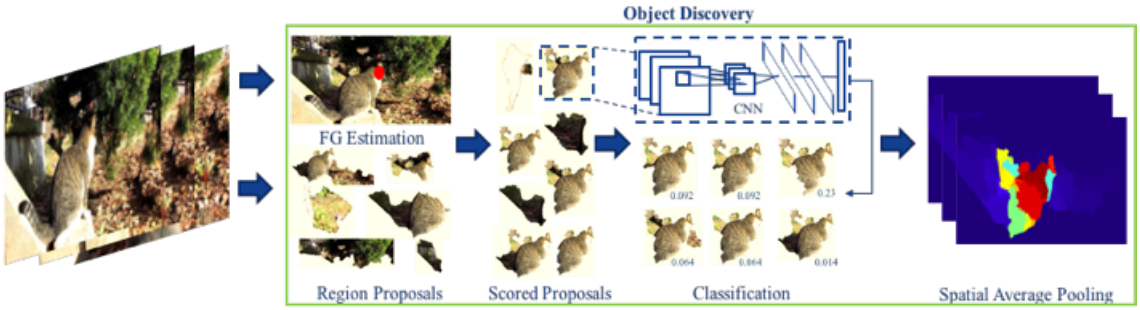


Figure 14. Overview of object discovery which consists of the following key steps: generating region proposals; scoring and selecting candidate proposals using generic properties; classifying proposals with regard to semantic labels applying image recognition model; generating semantic confidence map by aggregating both multiple proposals and spatial distributions.

The motion boundaries consist of two measures which account for two common types of motion, i.e., agile motion and modest motion. Let $\vec{f}_i$ be the optical flow vector at pixel i. The motion boundaries can be simply computed as the magnitude of the gradient of the optical flow motion vectors:

$$b_i^m = 1 - \exp(-w_m || \bigtriangledown \vec{f}_i ||) \tag{1}$$

where $b_i^m \in [0, 1]$ indicates the strength of the motion boundary at pixel i; w_m is a parameter to control its sensitivity to motion strength. This simple measure correctly detects boundaries at rapid moving pixels where b_i^m is close to 1, albeit it becomes unreliable when b_i^m is around 0.5 which can either be explained as boundaries or motion estimation error in optical flow [6]. To account for the second case, i.e., modest motions, a second estimator is computed to measure the difference of orientations between the motion vector of pixel x and its neighbours $j \in \mathcal{N}$. The insight is that if a pixel is moving in a different direction than all its surrounding pixels, it is probable located on a motion boundary. The orientation indicator is defined as

$$b_i^\theta = 1 - \exp(-w^\theta \max_{j \in \mathcal{N}} \Delta\theta_{i,j}^2), \tag{2}$$

where $\Delta\theta_{i,j}^2$ is the angle between the optical flow vectors at pixel i and j.

Combining these two measures above forms a measure which is more reliable than either alone to cope with various patterns of motion:

$$b_i = \begin{cases} b_i^m & \text{if } b_i^m > \eta^m \\ b_i^m \cdot b_i^\theta & \text{otherwise} \end{cases} \tag{3}$$

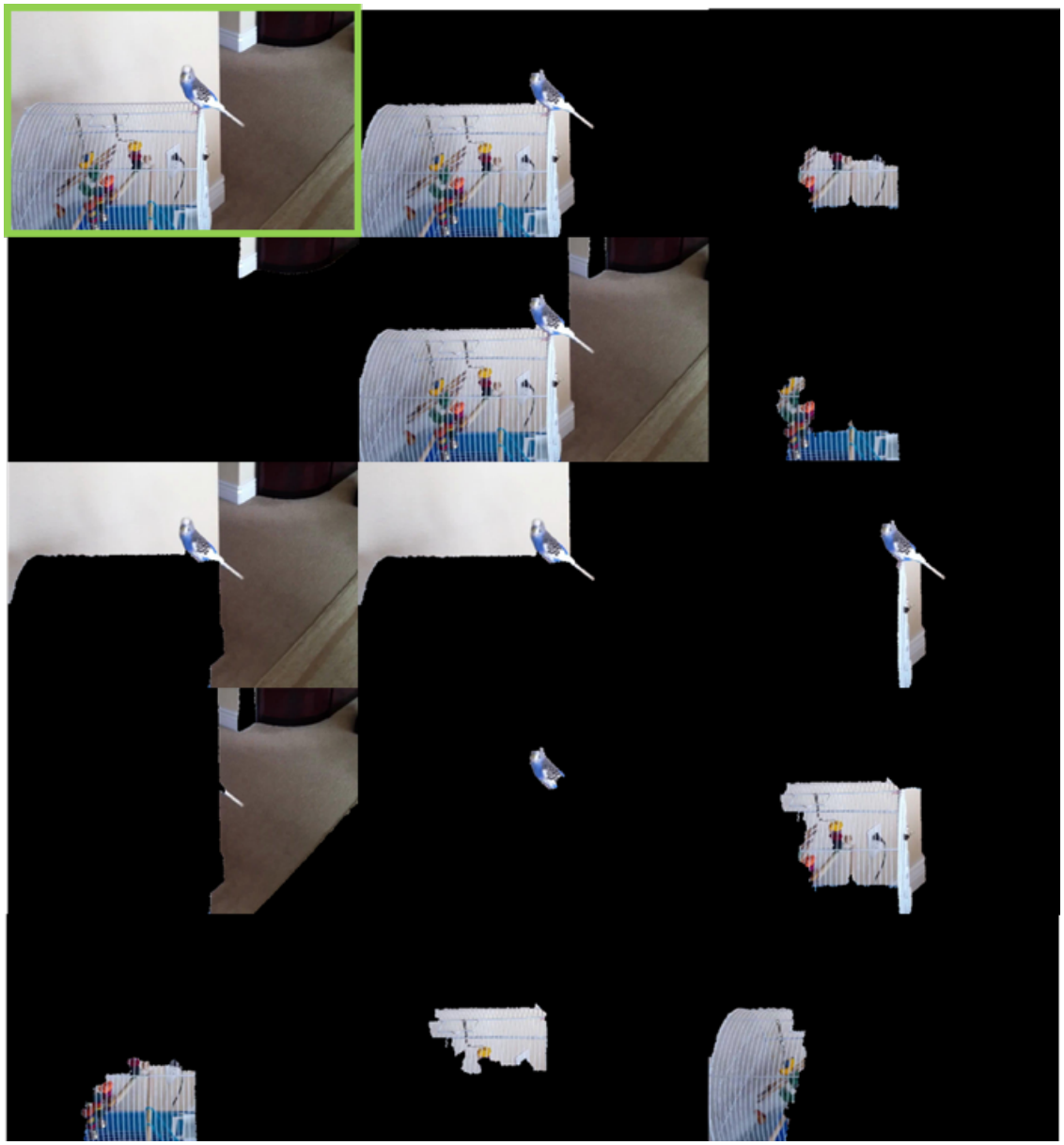
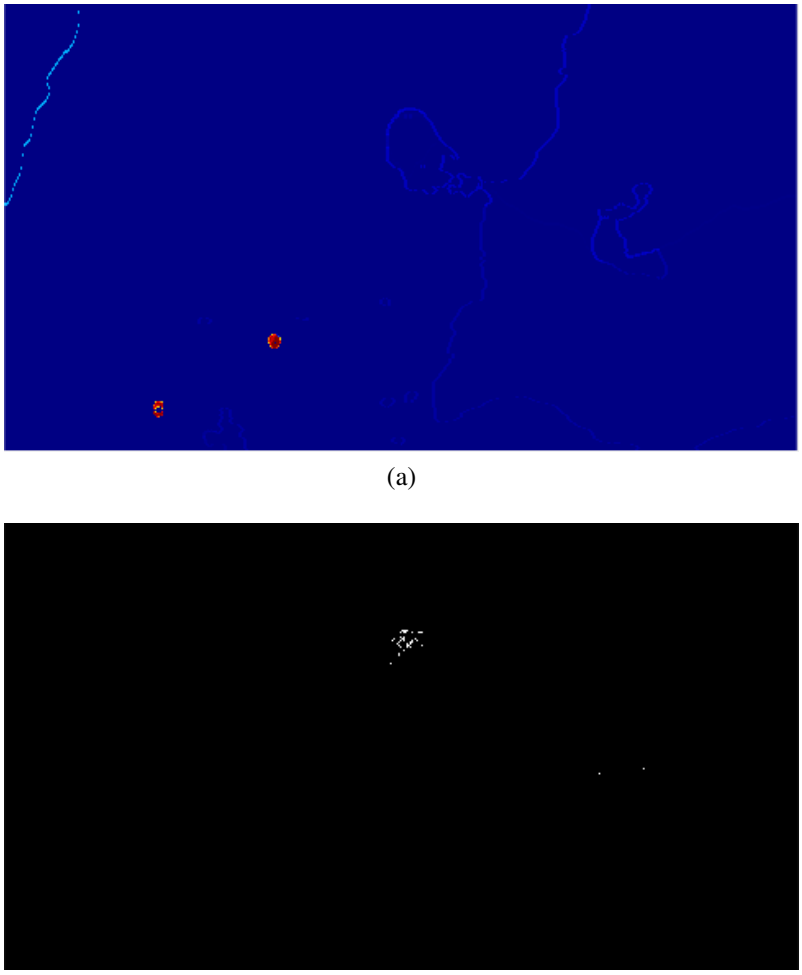


Figure 15. Exemplar region proposals randomly selected from the source image on top-left corner, extracted by [13]. The proposals are quite noisy with some regions corresponding to the objects of interest.

where η^m is a threshold. Finally, thresholding b_i at 0.5 produces a binary motion boundary. This threshold is chosen [6] empirically. Fig. 16a the motion boundaries estimated by this approach.

The resulted motion boundaries normally do not completely or correctly coincide with the whole object boundary due to inaccuracy of the optical flow estimation. This problem is further handled by an efficient algorithm by [6], determining whether one pixel is inside of the motion boundary. The key idea is that any ray starting from a pixel inside object intersects the boundary an odd number of times. Due to the incompleteness of detected



(a)

(b)

Figure 16. Local motion cues estimated using [6]: (a) motion boundaries (b) binary motion map.

motion boundaries, a number of rays, e.g., 8, are used to reach a majority voting. The final result is a binary map M^t for frame t indicating whether each pixel lies inside an object, which we use as motion cues. Fig. 16b shows such a binary map underpinned by the estimated motion boundaries.

After acquiring the motion cues, we score each proposal r by both appearance and context,

$$s_r = \mathcal{A}(r) + \mathcal{C}(r)$$

where $\mathcal{A}(r)$ stands for the score of appearance for region r computed using [13] and $\mathcal{C}(r)$ represents the contextual score of region r which is defined as:

$$\mathcal{C}(r) = \mathrm{Avg}(M^t(r)) \cdot \mathrm{Sum}(M^t(r))$$

where $\mathrm{Avg}(M^t(r))$ and $\mathrm{Sum}(M^t(r))$ compute the average and total amount of motion

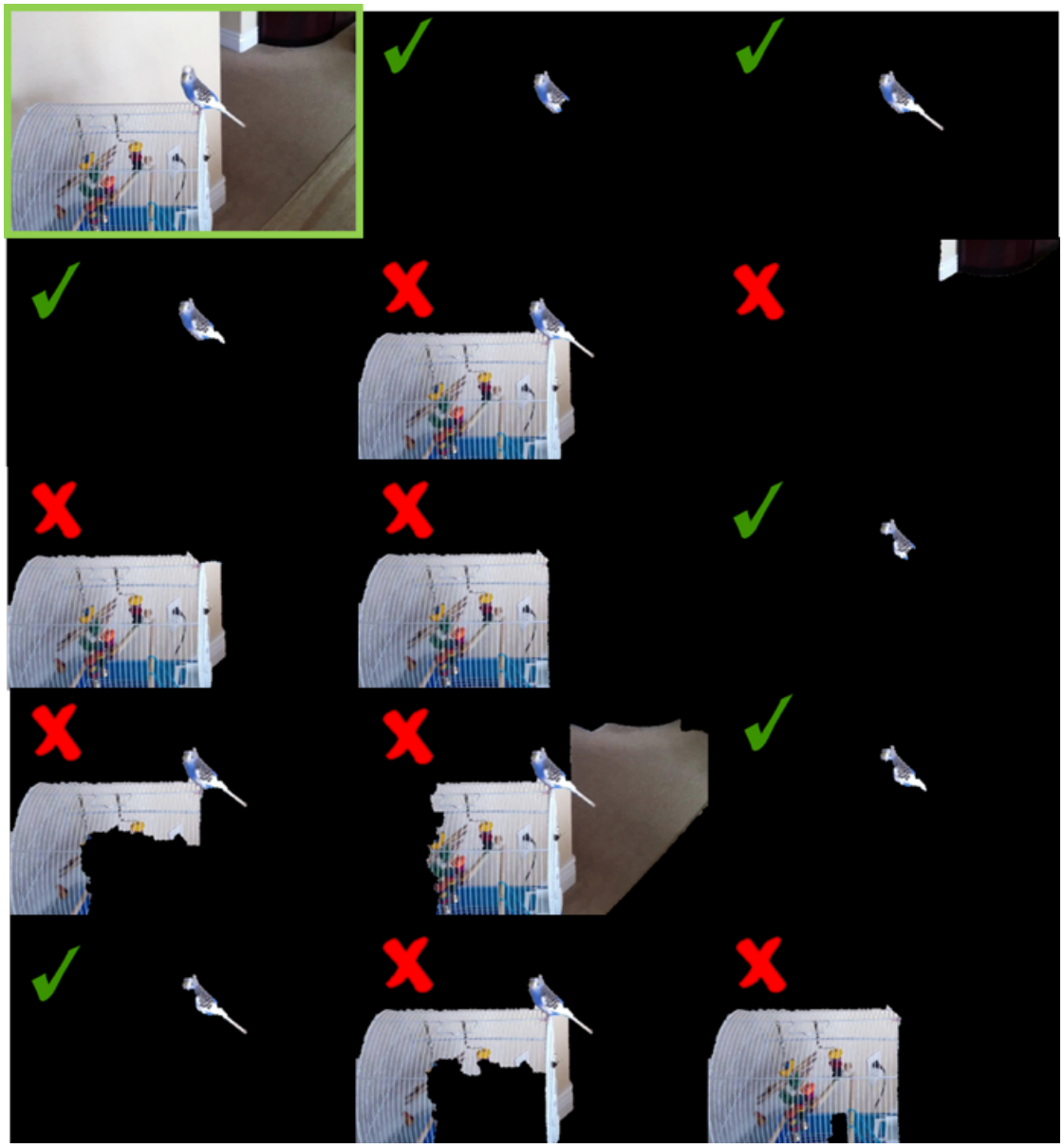


Figure 17. Scored and top ranked region proposals applying our scoring scheme, where even the top dozen contain good proposals (ticked) of the object of interest.

cues [6] included by proposal r on frame t respectively. Note that, appearance, contextual and combined scores are normalised. Exemplars of scored and top ranked region proposals applying our scoring scheme are shown in Fig. 17.

3.2 Proposal Classification

On each frame t we have a collection of region proposals scored by their appearance and contextual information. These region proposals may contain various objects present in the

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 18. VGG network configurations (listed in [10]). The depth of the configurations increases from left (A) to right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The rectified linear unit (ReLU) activation function is not shown for brevity. FC-N indicates fully connected layers with N neurons.

video. In order to identify the objects of interest specified by the video level tag, region level classification is performed. We consider proven classification architectures such as VGG-16 nets [10] which did exceptionally well in ILSVRC14. VGG-16 net uses 3×3 convolution interleaved with max pooling and three fully-connected layers. The various network architectures are shown in Fig. 18.

In order to classify each region proposal, we firstly warp the image patch in each region proposal into a form which is compatible with the pre-trained CNN (VGG-16 net requires fixed size input of 224×224 pixels). Despite that there are various possible transforma-

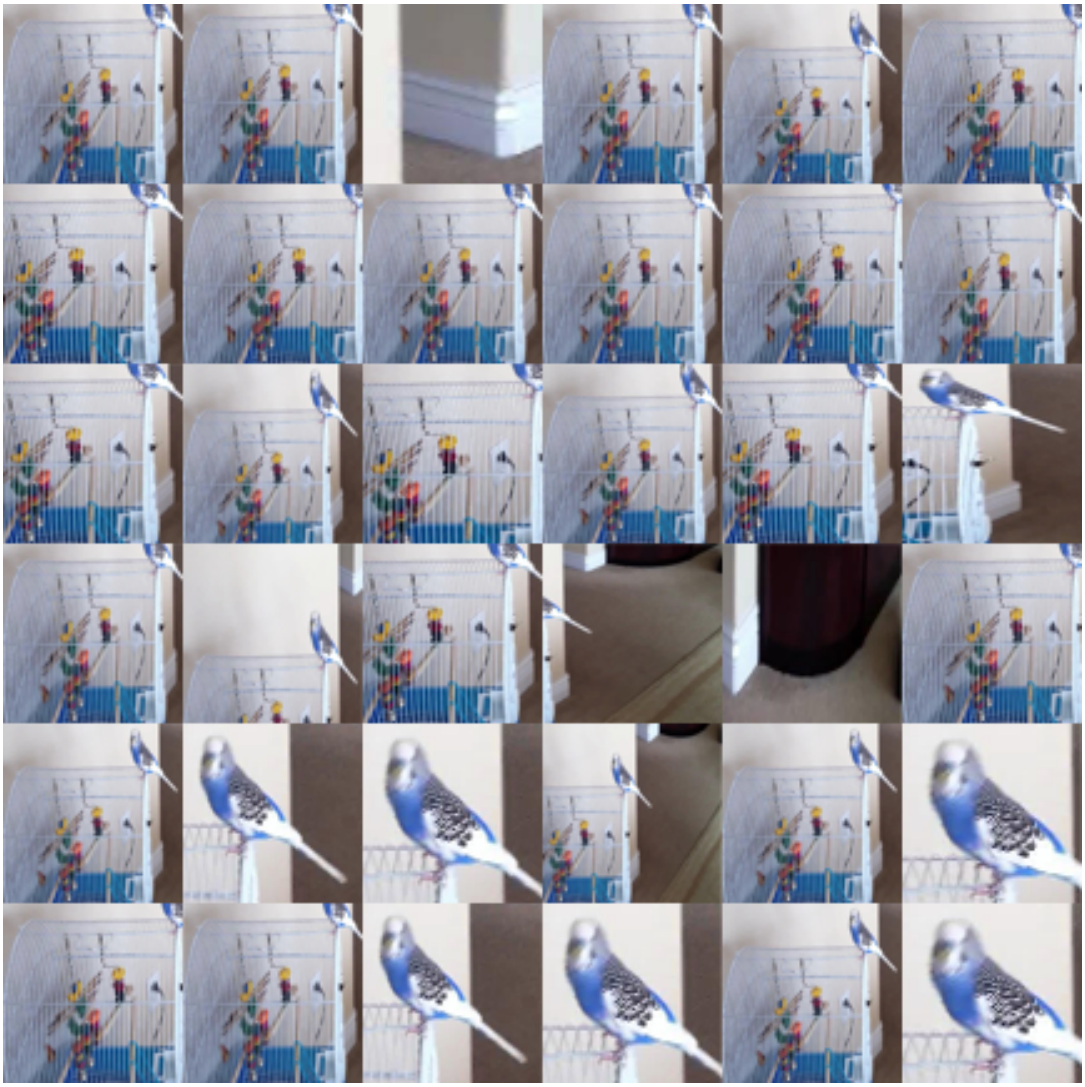


Figure 19. Warped regions from the source image in Fig. 15.

tions of those region proposals with arbitrary shapes, we warp the bounding box around each region to the required dimensions, regardless of its original size or shape. The tight bounding box is expanded to four directions by a certain number of pixels (10 in our system) around the original box before warping, which was proven effective in the task of using image classifier for object detection task [16]. Fig. 19 shows some examples of warped training regions.

After the classification, we collect the confidence of all the regions with respect to the specific classes associated with the video and form a set of scored regions,

$$\{\mathcal{H}_{w_1}, \ldots, \mathcal{H}_{w_K}\}$$

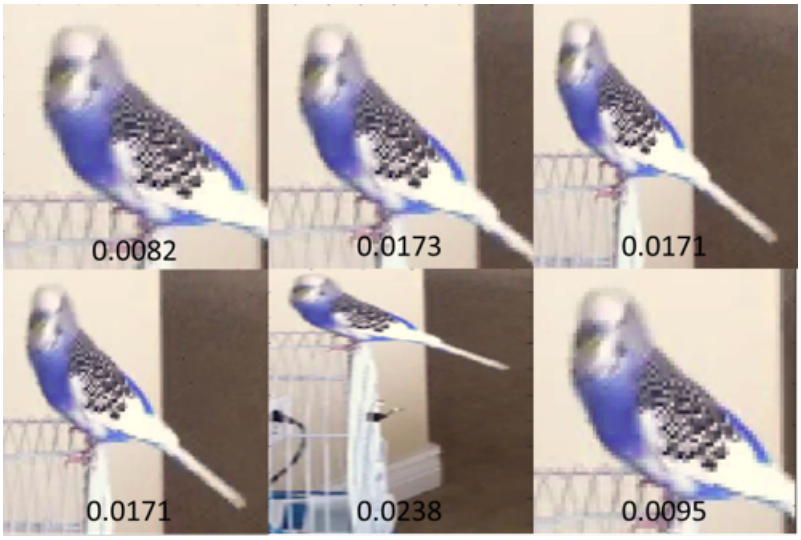


Figure 20. Positive detections with thresholded confidences using the pre-trained VGG-16. Due to the nature of image classifier, higher confidence does not necessarily correspond to good proposals.

where

$$\mathcal{H}_{w_k} = \{(r_1, s_{r_1}, c_{r_1,w_k}), \ldots, (r_N, s_{r_N}, c_{r_N,w_k})\}$$

with s_{r_i} is the original score of proposal r_i and c_{r_i,w_k} is its confidence from CNN classification with regard to keyword or class w_k. Fig. 20 shows the positive detections with confidence higher than a predefined threshold (0.01), where higher confidence does not necessarily correspond to good proposals. This is mainly due to the nature of image classification where the image frame is quite often much larger than the tight bounding box of the object. In the following discussion we drop the subscript of classes, and formulate our method with regard to one single class for the sake of clarity, albeit our method works on multiple classes.

3.3 Spatial Average Pooling

After the initial discovery, a large number of region proposals are positively detected with regard to a class label, which include overlapping regions on the same objects and spurious detections. We adopt a simple weighted spatial average pooling strategy to aggregate the region-wise score, confidence as well as their spatial extent, as shown in Fig. 21. For each proposal r_i, we rescore it by multiplying its score and classification confidence, which is denoted by $\tilde{s}_{r_i} = s_{r_i} \cdot c_{r_i}$. We then generate score map $\mathcal{S}_{r_i}$ of the size of image frame, which is composited as the binary map of current region proposal multiplied by

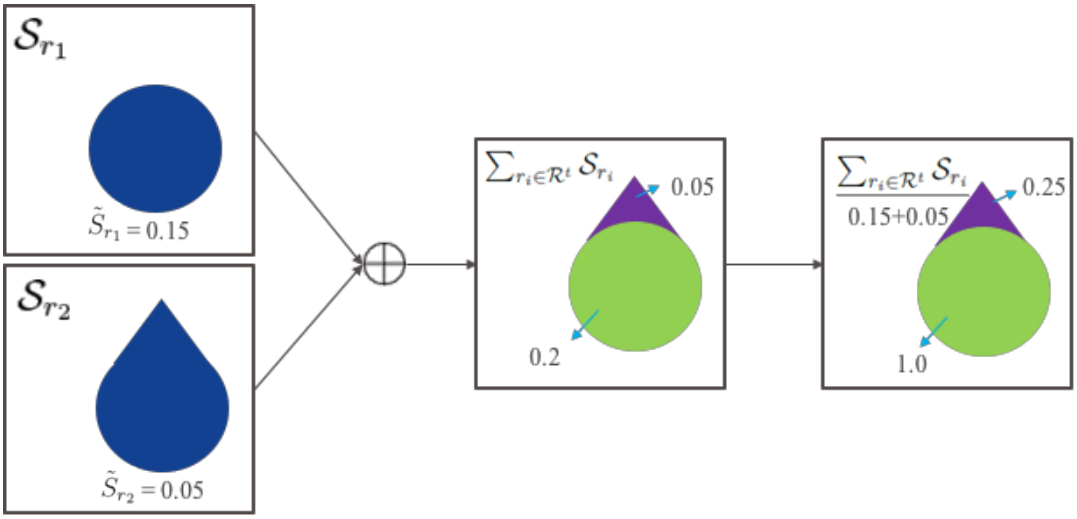


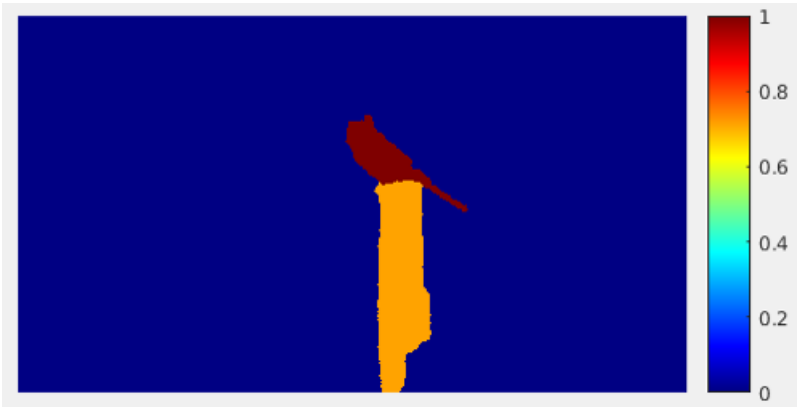
Figure 21. An illustration of the weighted spatial average pooling strategy. Regions having better spatial support from high-confidence proposals have higher confidence after the pooling.

its score $\tilde{s}_{r_i}$. We perform an average pooling over the score maps of all the proposals to compute a confidence map,

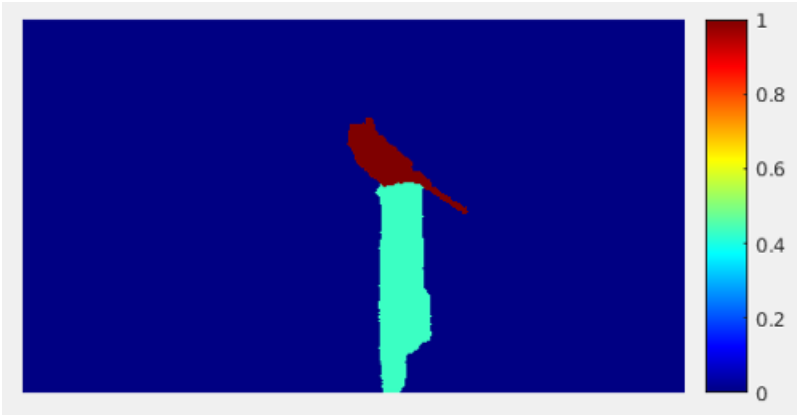
$$C^t = \frac{\sum_{r_i \in \mathcal{R}^t} \mathcal{S}_{r_i}}{\sum_{r_i \in \mathcal{R}^t} \tilde{s}_{r_i}} \tag{4}$$

where $\sum_{r_i \in \mathcal{R}^t} \mathcal{S}_{r_i}$ performs element-wise operation and $\mathcal{R}^t$ represents the set of candidate proposals from frame t.

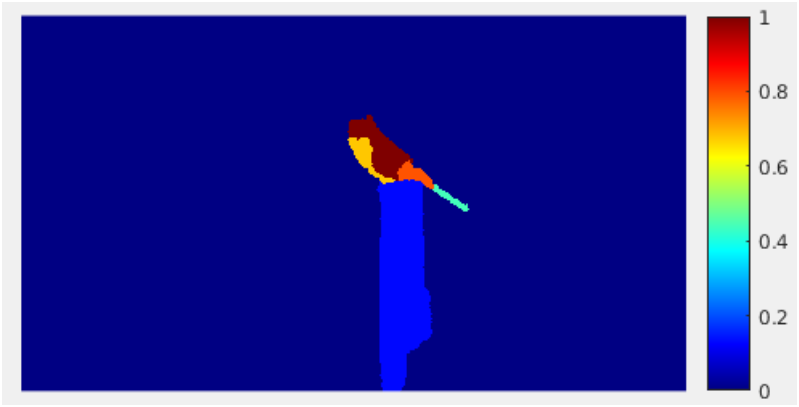
The resulted confidence map $\mathcal{C}^t$ aggregates not only the region-wise score but also their spatial extent. The key insight is that good proposals coincide with each other in the spatial domain (after restoring to the original size) and their contribution to the final confidence map are proportional to their region-wise score. An illustration of the weighted spatial average pooling is shown in Fig. 21. In this exemplar illustration, the confidence in some region is high (Fig. 22a) although it does not belong to the object. By employing our weighted average pooling strategy, the resulted confidence is lowered to a reasonable level (Fig. 22c) by both its lower score (Fig. 22b) and the lack of spatial support from other proposals.



(a)



(b)



(c)

Figure 22. (a) Averaged score map based on proposal scores (b) Averaged confidence map based on CNN-16 confidence (c) Weighted spatial average pooling confidence map.

4 Domain Adaptation

In this section, we adapt the source domain from image recognition to the target domain, i.e., pixel or superpixel level labelling. We develop two approaches for domain adaptation. The first approach is built by incorporating constraints in the spatial-temporal domain obtained from a connectivity graph defined on unlabelled target instances, whilst the second approach exploits the multiple instance of the same objects recurring in continuous frames and learns an object-specific representation in deep feature space.

4.1 Approach I: Semi-Supervised Graphical Model

4.1.1 Graph Construction

To perform domain adaptation from image recognition to video object segmentation, we define a undirected space-time graph of superpixels $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ spanning a video or a shot. Each node of the graph corresponds to a superpixel, and each weighted edge connects two superpixels according to spatial and temporal adjacencies in video data, as illustrated in Fig. 23. Temporal adjacency is determined given optical flow motion vectors, i.e., two superpixels are deemed temporally adjacent if they are connected by at least one motion vector.

To model the weighted edges, we firstly compute the affinity matrix A of the graph among spatial neighbours as

$$A^s_{i,j} = \frac{\exp(-d^c(s_i, s_j))}{d^s(s_i, s_j)} \tag{5}$$

where the functions $d^s(s_i, s_j)$ and $d^c(s_i, s_j)$ computes the spatial and colour distances between spatially neighbouring superpixels s_i and s_j respectively:

$$d^s(s_i, s_j) = ||r_i - r_j|| \tag{6}$$

$$d^c(s_i, s_j) = \frac{||c_i - c_j||^2}{2 < ||c_i - c_j||^2 >} \tag{7}$$

where $||r_i - r_j||$ is the Euclidean distance between two superpixel centres r_i and r_j respectively; $||c_i - c_j||^2$ is the squared Euclidean distance between two adjacent superpixels in RGB colour space, and $< \cdot >$ computes the average over all pairs i and j.

For affinities among temporal neighbours s_i^{t-1} and s_j^t, we consider both the temporal and

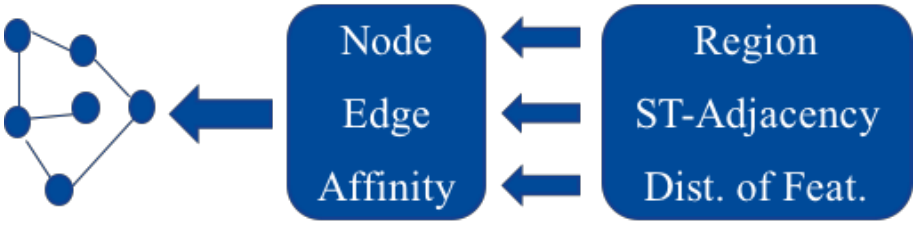


Figure 23. Graph construction for domain adaptation.

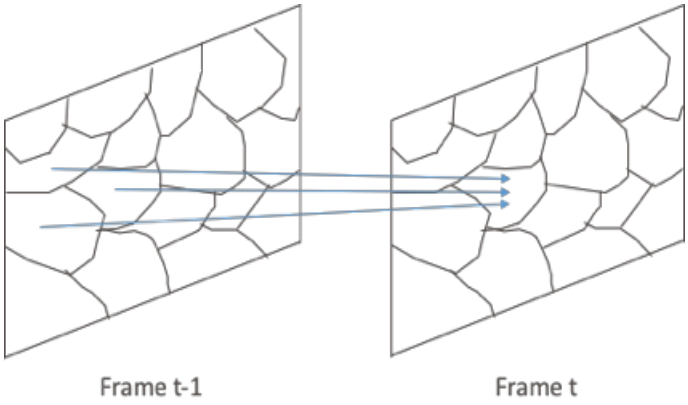
colour distances between s_i^{t-1} and s_j^t,

$$A_{i,j}^t = \frac{\exp(-d^c(s_i, s_j))}{d^t(s_i, s_j)}$$

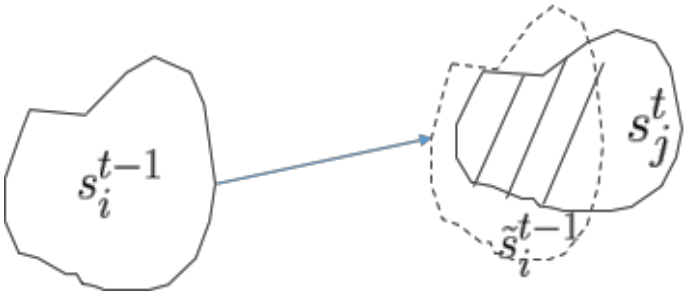
where

$$\begin{aligned} d^t(s_i, s_j) &= \frac{\rho_{i,j}}{m_i}, \\ m_i &= \exp(-w_c \cdot \pi_i), \\ \rho_{i,j} &= \frac{|\tilde{s}_i^{t-1} \cap s_j^t|}{|\tilde{s}_i^{t-1}|}. \end{aligned} \tag{8}$$

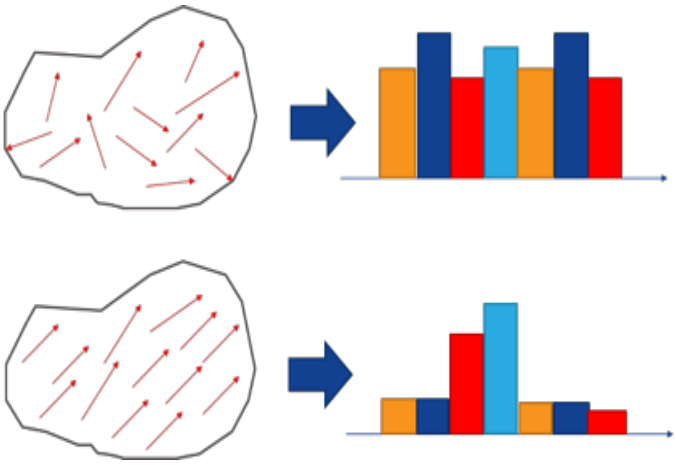
Specifically, we define the temporal distance $d^t(s_i, s_j)$ by combining two factors, i.e., the temporal overlapping ratio $\rho_{i,j}$ and motion accuracy m_i. π_i denotes the motion non-coherence, and $w_c = 2.0$ is a parameter. The larger the temporal overlapping ratio is between two temporally related superpixels, the closer they are in temporal domain, subject to the accuracy of motion estimation. The temporal overlapping ratio $\rho_{i,j}$ is defined between the warped version of s_i^{t-1} following motion vectors and s_j^t, where $\tilde{s}_i^{t-1}$ is the warped region of s_i^{t-1} following optical flow to frame t, and $|\cdot|$ denotes the cardinality of a superpixel, as shown in Fig. 24b. The reliability of motion estimation inside s_i^{t-1} is measured by the motion non-coherence. A superpixel, i.e., a small portion of a moving object, normally exhibits coherent motions. We correlate the reliability of motion estimation of a superpixel with its local motion non-coherence as illustrated in Fig. 24c. We compute quantised optical flow histograms h_i for superpixel s_i^{t-1}, and compute π_i as the information entropy of h_i. Larger π_i indicates higher levels of motion non-coherence, i.e., lower motion reliability of motion estimation. An example of computed motion reliability map is shown in Fig. 25.



(a) Temporal neighbours are roughly determined by optical flow motion vectors.



(b) Temporal warping by following motion vectors in the temporal domain.



(c) Motion Accuracy and histogram of motion vectors. Lower motion estimation accuracy is indicated by flat histogram of motion vectors.

Figure 24. Various factors considered in measuring temporal distance.

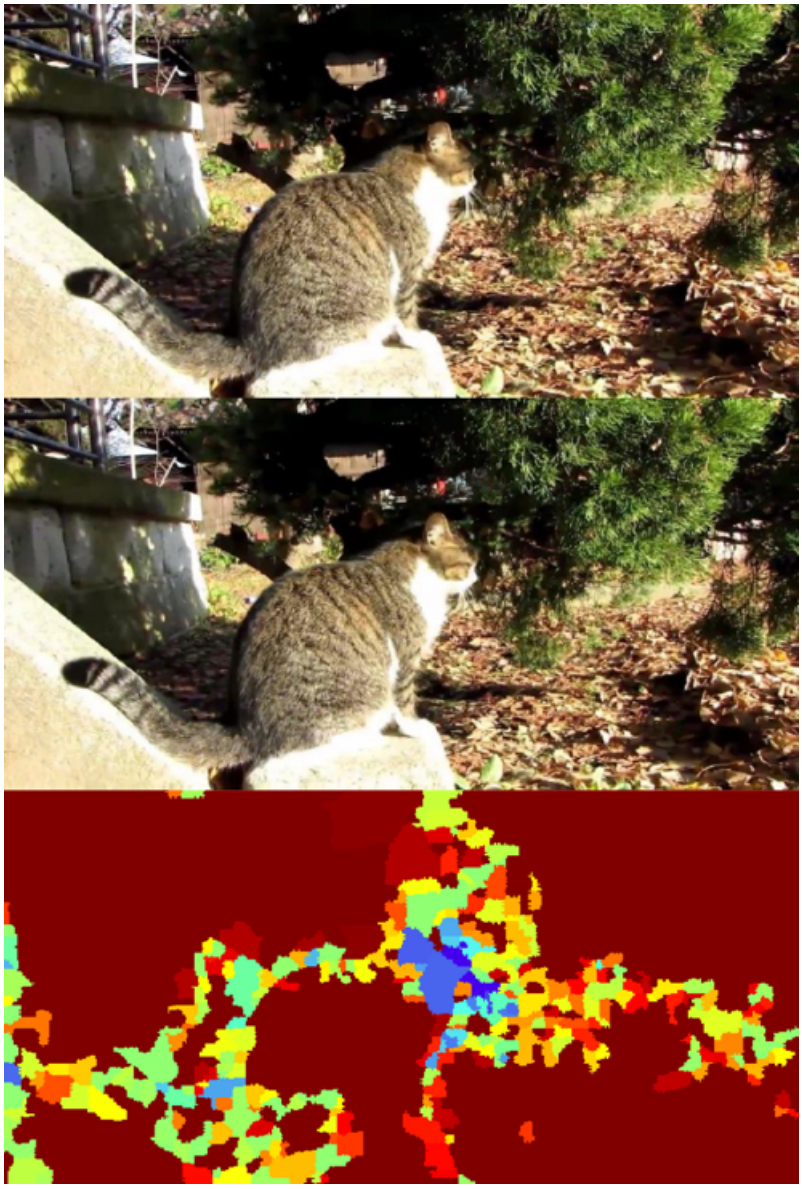


Figure 25. Motion reliability map (bottom) computed given the optical flow between two consecutive frames (top and middle).

4.1.2 Semi-Supervised Learning

We minimise an energy function $E(X)$ with respect to all superpixels confidence X ($X \in [-1, 1]$) following a formulation similar with [85]:

$$E(X) = \sum_{i,j=1}^{N} A_{ij} ||x_i d_i^{-\frac{1}{2}} - x_j d_j^{-\frac{1}{2}}||^2 + \mu \sum_{i=1}^{N} ||x_i - c_i||^2, \tag{9}$$

where μ is the parameter to control the regularization, and X are the desirable confidence of superpixels which are imposed by noisy confidence C in Eq. 4. We set $\mu = 0.5$

empirically. Let the node degree matrix $D = \mathrm{diag}([d_1, \ldots, d_N])$ be defined as $d_i = \sum_{j=1}^{N} A_{ij}$, where $N = |\mathcal{V}|$.

Denoting $S = D^{-1/2} A D^{-1/2}$, this energy function can be minimised iteratively as

$$X^{t+1} = \alpha S X^t + (1 - \alpha) C$$

until convergence, where α controls the relative amount of the confidence from its neighbours and its initial confidence. Specifically, the affinity matrix A of $\mathcal{G}_d$ is symmetrically normalized in S, which enables the convergence of the consecutive iteration. In each iteration, each superpixel adapts itself by receiving the confidence from its neighbours while preserving its initial confidence. The confidence is adapted symmetrically since S is symmetric. After convergence, the confidence of each unlabelled superpixel is adapted to be the neighbour of which it has received most confidence during the iterations.

We alternatively solve the optimization as a linear system of equations, which is more efficient. Differentiating $E(X)$ with respect to X we have

$$\nabla E(X)|_{X=X^*} = X^* - SX^* + \mu(X^* - C) = 0 \tag{10}$$

which can be transformed as

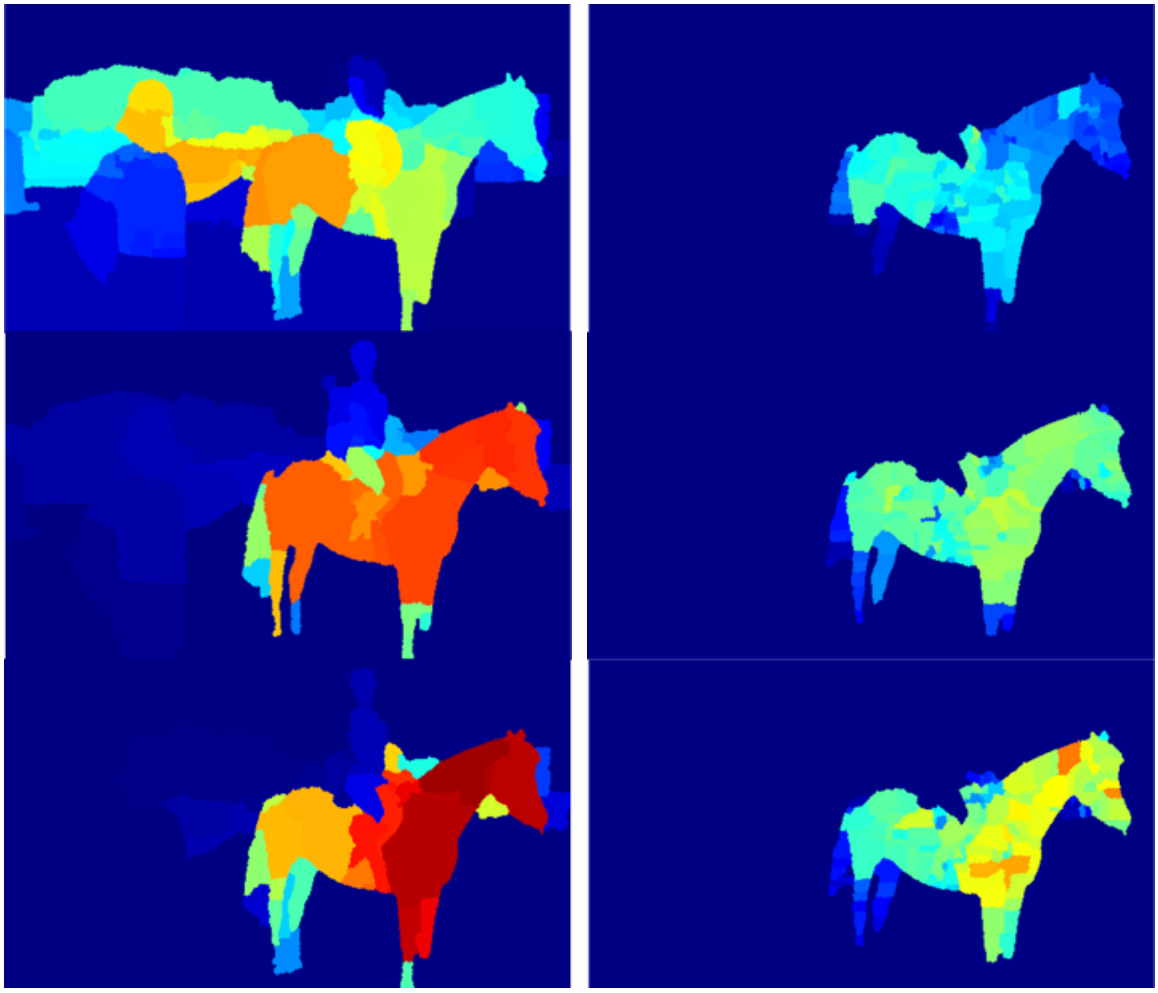
$$(I - (1 - \frac{\mu}{1+\mu})S)X^* = \frac{\mu}{1+\mu} C. \tag{11}$$

Finally we have

$$(I - (1 - \eta)S)X^* = \eta C. \tag{12}$$

where $\eta = \frac{\mu}{1+\mu}$.

The optimal solution for X can be found using the preconditioned (Incomplete Cholesky factorization) conjugate gradient method with very fast convergence. Fig. 26 shows the result applying the proposed domain adaptation which effectively adapts the noisy confidence map from image recognition to the video object segmentation domain. For consistency, still let C denote the optimal semantic confidence X for the rest of this thesis.



(a) Confidence maps of three consecutive frames, which exhibit noisy semantic confidences with regard to 'horse' class.

(b) Confidence maps after domain adaptation which demonstrate strong spatial-temporal coherence and semantic accuracy.

Figure 26. Proposed domain adaptation effectively adapts the noisy confidence map from image recognition to the video object segmentation domain.

4.2 Approach II: Video Object Representation Learning

After the object discovery, we have a noisy evidence of the object on each independent frame. We set about learning an object-specific representation which captures the synergy of the same object instances in deep feature space from continuous frames, as illustrated in Fig. 27.

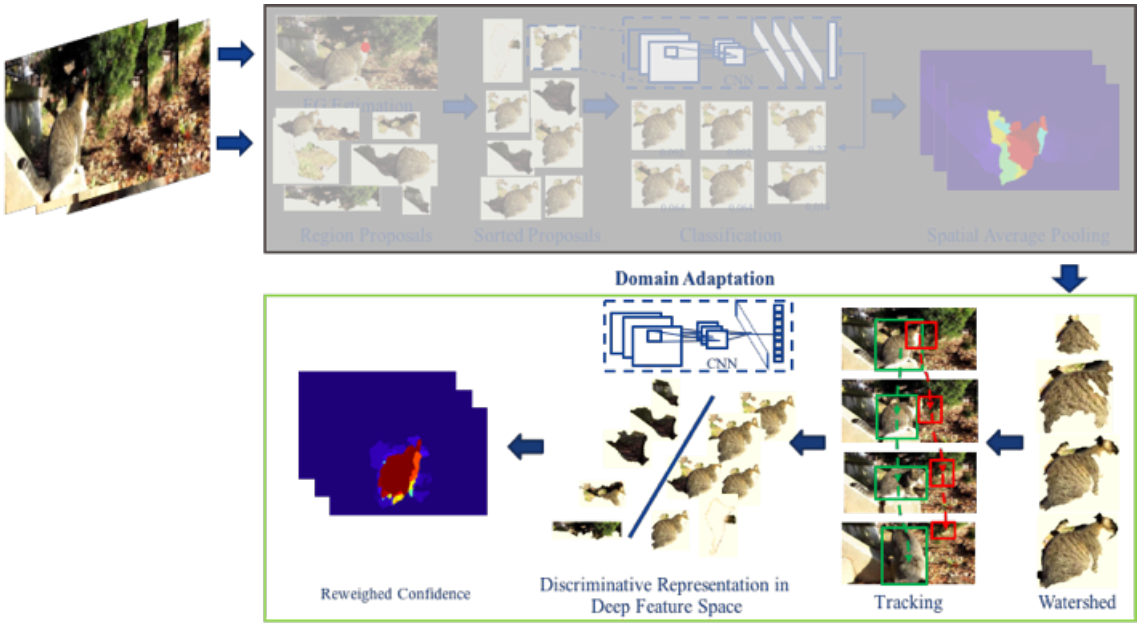


Figure 27. Overview of domain adaptation approach II which utilises watershed and object tracking to form an object-specific representation in deep feature space.

4.2.1 Proposal Generation

Based on the computed confidence map (Eq. 4), we generate a new set of region proposals in a process analogous to the watershed algorithm, i.e., we gradually increase the threshold in defining binary maps from confidence map $\mathcal{C}^t$. This approach effectively exploits the topology structure of the confidence map. The disconnected regions thresholded at each level form the new proposals. The confidence associated with these new region proposals $\mathcal{P}$ are computed by averaging the confidence values enclosed by each region. Fig. 28 shows an illustration of the process.

4.2.2 Tracking for Proposal Mining

The generated region proposals are still noisy, containing false positives or poor positives. Due to the 2D projections, it is not possible to learn a complete representation of the object in one frame, whereas multiple image frames encompassing the same object or part of the object provide more comprehensive information. Video data naturally encodes the rich information of the objects of interest. We perform visual tracking [32] on proposals to achieve two purposes: firstly, visual tracking can eliminate false positives since spurious detections normally do not appear very often on other frames; secondly, we are able to extract consistent proposals describing the same object instances to learn an object-

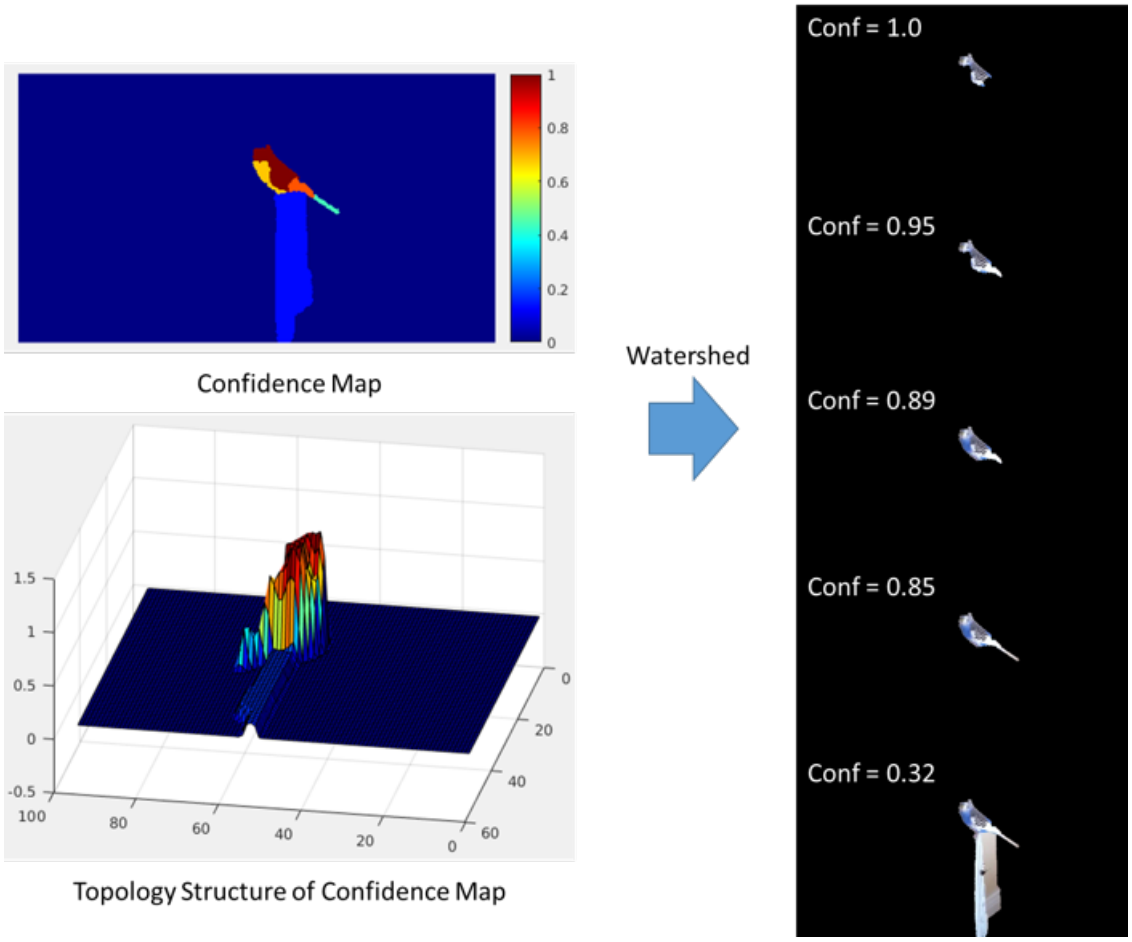


Figure 28. Applying a watershed-like process generates reliable new region proposals.

specific representation.

We propose an iterative tracking and eliminating approach to achieve these goals, as illustrated in Fig. 29. Proposals from all frames form a pool of candidates. Each iteration starts by randomly selecting a proposal on the earliest frame in the pool of candidate proposals, and it is tracked until the last frame of the sequence. Any proposals whose bounding boxes with a substantial intersection-over-union (IoU) overlap (0.5 is a generally accepted value in detector evaluation for selecting positive examples) with the tracked bounding box are chosen to form a track and removed from the pool. This process iterates until the candidate pool is empty, and forms a set of tracks $\mathcal{T}$ with single-frame tracks discarded. For each track $T_i \in \mathcal{T}$, we compute a stability indicator d_{T_i} which is measured by its tracking duration $|T_i|$ comparing to other tracks,

$$d_{T_i} = 1 - \exp(-\frac{(C_{T_i})^2}{< \mathcal{C} >^2}) \tag{13}$$

where $< \cdot >$ denotes the expectation over all track durations. This stability indicator is

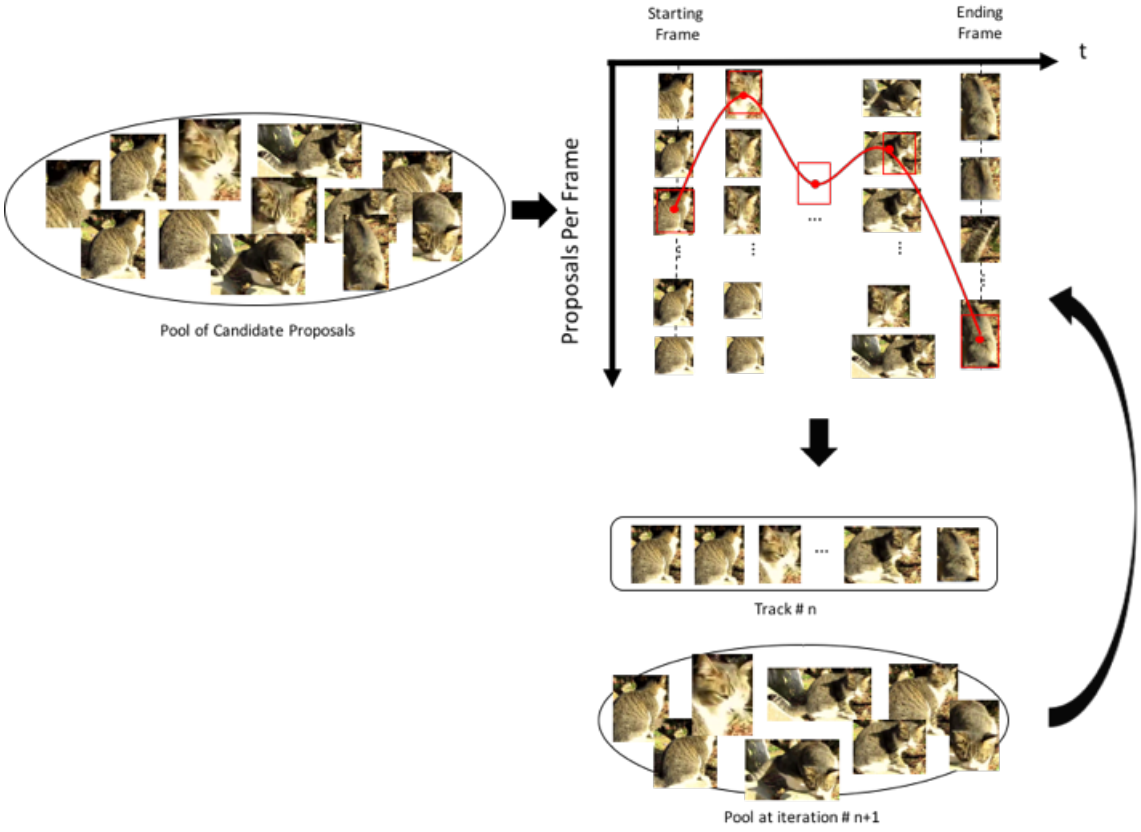


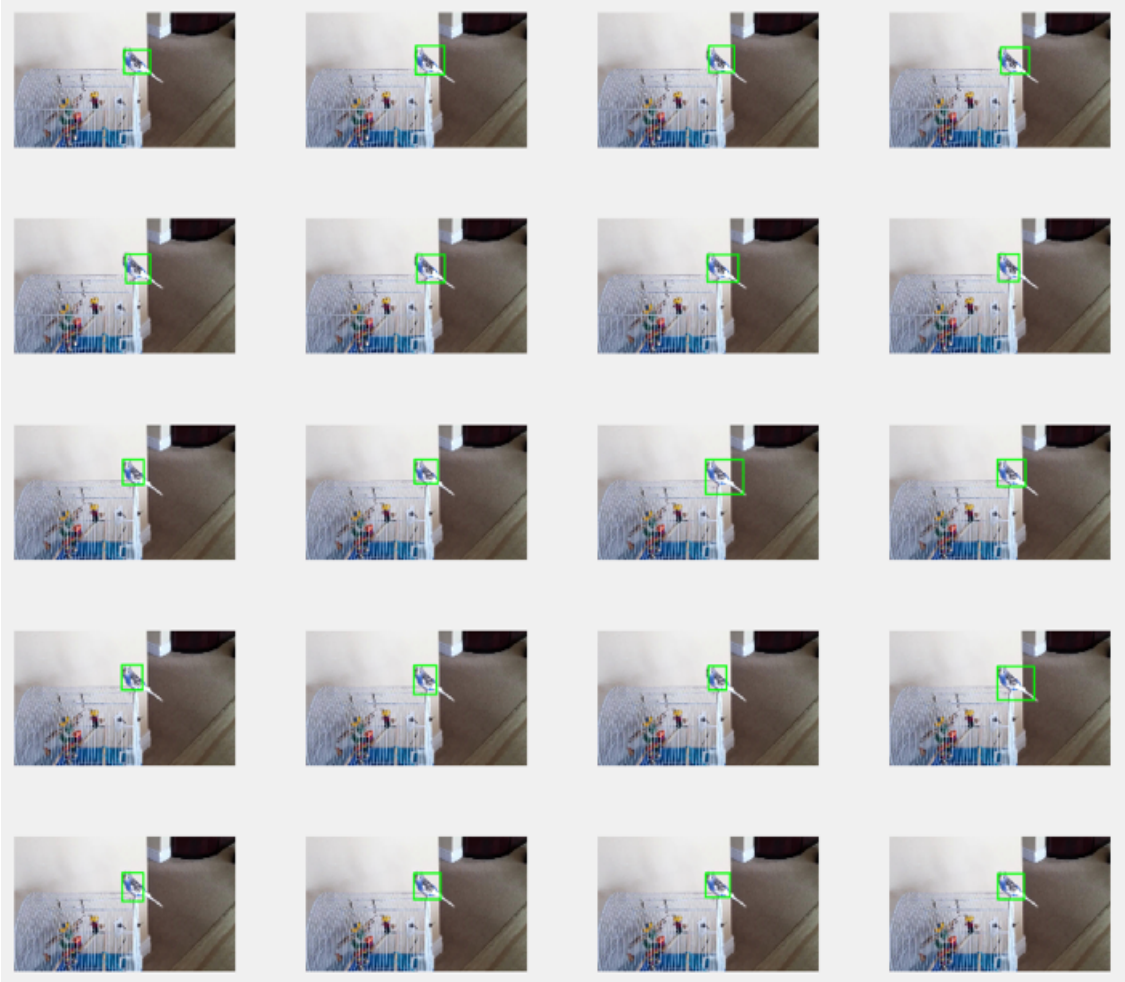
Figure 29. Iterative tracking to eliminate spurious detections and extract the consistent proposals. Each iteration starts by randomly selecting a proposal on the earliest frame in the pool of candidate proposals, and it is tracked until the last frame of the sequence. Any proposals with a substantial IoU overlap are chosen to form a track and removed from the pool. This process iterates until the candidate pool is empty.

used in the next subsection to sample the positive examples. Fig. 30 shows a track of proposals with the corresponding stability indicator.

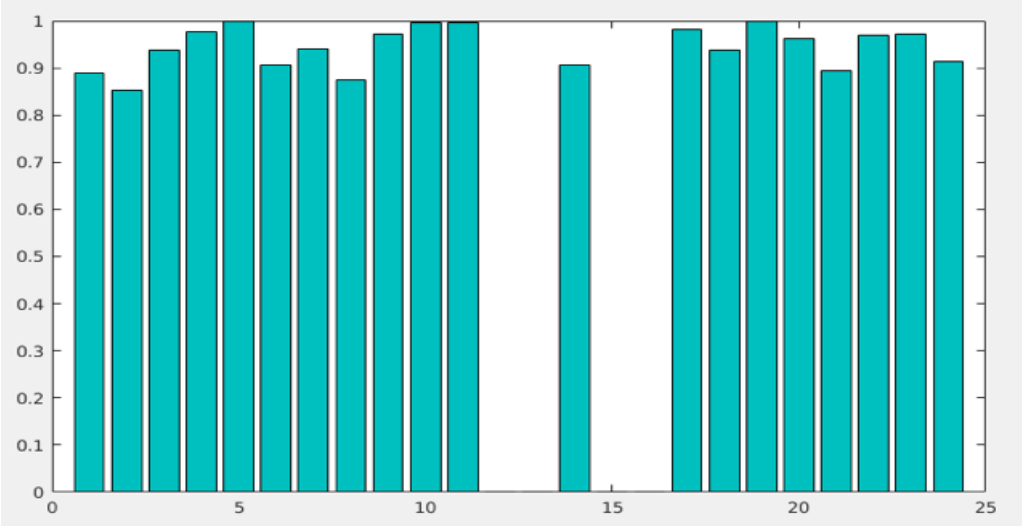
4.2.3 Discriminative Representation Learning

Our goal is to learn a discriminative object-specific representation such that the good proposals are closer to each other than to the bad or false positive proposals in the deep feature space. We now describe how the tracks can be used as training samples for representation learning.

We firstly sample positive examples from the set of tracks $\mathcal{T}$. For each track $T_i \in \mathcal{T}$, we sample proposals with respect to its stability indicator d_{T_i}; as a result, more proposals are sampled from the stabler tracks while unstable tracks contribute less. For negative examples, we randomly sample bounding boxes around these positive examples and take ones



(a)



(b)

Figure 30. (a) Exemplar track of proposals and (b) their confidence; zero values indicate that current track cannot find the overlapping (> 0.5) proposal on the corresponding frames.

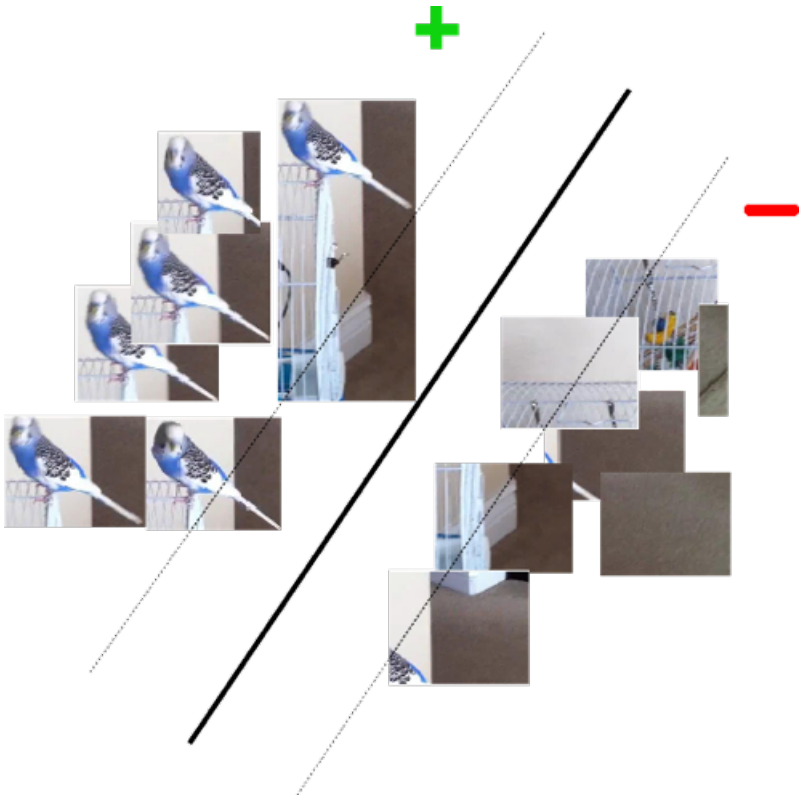


Figure 31. Positive and negative training examples are used to train a linear svm classifier for each class in the deep feature space.

which have an IoU overlap less than 0.3 (which is a generally accepted value in detector evaluation for selecting negative examples) with the bounding boxes of corresponding positive examples. One could fine-tune the whole VGG-16 net end-to-end optimizing the weights of the CNN for the feature representation and the weights for classifying each video object jointly. However, the number of positive examples from each video appears limited for effective fine-tuning a deep network like the one of VGG-16 net. We choose to simplify the problem by decoupling it. We warp all extracted training instances and forward propagate them through the VGG-16 net. As suggested by [16, 80], a 4096-dimensional feature vector is extracted from each training instance by reading off features from *fc6* layer as the input representation of each training sample. Once features are extracted, we train one linear SVM per class with training labels applied.

4.2.4 Proposal Reweighing

Taking the proposals $\mathcal{P}$ which are generated in Sec. 4.2.1, we extracted a 4096-dimensional feature vector of the proposals $(r_i, s_{r_i}) \in \mathcal{P}$ where s_{r_i} indicates the score of proposal r_i after the Watershed process. Our goal is to reweigh all proposals $\mathcal{P}$ with our learned dis-

criminative object-specific representation. We score proposals using the SVM trained for that class, illustrated in Fig. 31,

$$c_{r_i} = w_k \cdot x_{r_i} + b_k, \tag{14}$$

where w_k and b_k are the weights and bias for class k.

We apply a similar weighted average pooling strategy as in Sec. 3.3 to aggregate the region-wise confidence and their spatial extent. For each proposal r_i, we rescore it by multiplying its score s_{r_i} and SVM classification confidence c_{r_i}, which is denoted by $\tilde{s}_{r_i} = s_{r_i} \cdot c_{r_i}$. We then generate score map $\mathcal{S}_{r_i}$ of the size of image frame, which is composited as the binary map of current region proposal multiplied by its score $\tilde{s}_{r_i}$. We perform an average pooling over the score maps of all the proposals to compute a confidence map:

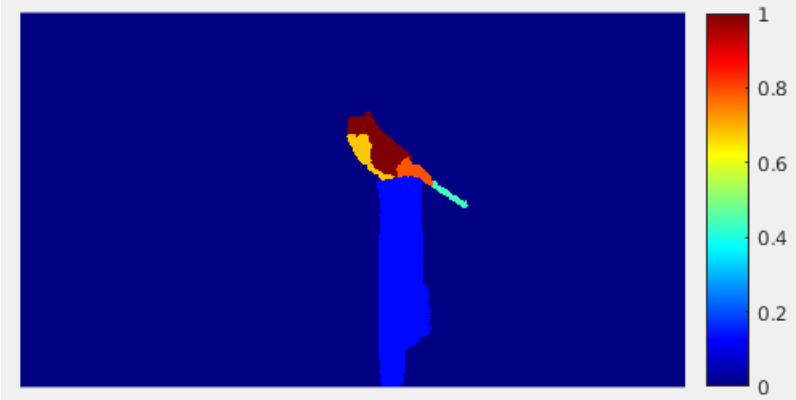
$$\mathcal{C}^t = \frac{\sum_{r_i \in \mathcal{R}^t} \mathcal{S}_{r_i}}{\sum_{r_i \in \mathcal{R}^t} \tilde{s}_{r_i}} \tag{15}$$

where $\sum_{r_i \in \mathcal{R}^t} \mathcal{S}_{r_i}$ performs element-wise operation and $\mathcal{R}^t$ represents the set of candidate proposals from frame t. The reweighing strategy is illustrated in Fig. 32. The reweighed proposals collectively form a confidence map per frame indicating the evidence of the presence of objects from certain category.

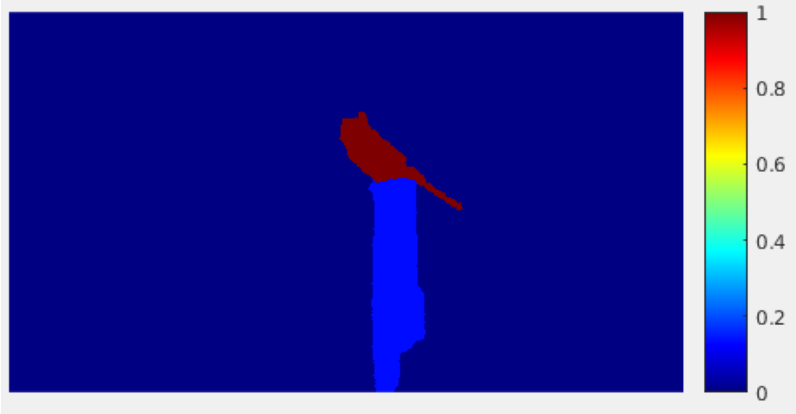
4.2.5 Semantic Confidence Diffusion

The estimation of semantic evidence, i.e., confidence map, is performed independently on each frame, regardless of the temporal information in the sequence. This frame-dependent semantic evidence might be stronger and more reliable on certain frames, whereas it could be weaker or spurious on some frames which would result in erroneous segmentation in the later stage. We adopt a semantic confidence diffusion model in this section, to propagate and accumulate the frame-dependent semantic evidence over the temporal domain to form spatio-temporal confidence maps. The key insight is that once the evidence can "flow" smoothly in the video data, its reliability is boosted.

Firstly, we convert the pixel based confidence map into superpixel based confidence, by averaging confidence of all pixels inside each superpixel. Each superpixel enforces a higher level of local smoothness. We start by diffusing the per-superpixel confidence among three consecutive frames $t-1$, t and $t+1$. For superpixel s_i^t on frame t, we update its confidence by combining the diffused confidence from superpixels s_j^{t-1} on frame $t-1$



(a)



(b)

Figure 32. (a) Confidence map before reweighing which does not have consistent predictions over the object. (b) Final confidence map after applying reweighing strategy which gives consistent predictions due to the object representation learning in deep feature space.

and superpixels s_k^{t+1} on frame $t+1$. During diffusion, the amount of diffused confidence from s_j^{t-1} and s_k^{t+1} to s_i^t is determined by two factors — the correlation between each pair considering motion "flows" and the reliability of motion estimation inside s_j^{t-1} and s_k^{t+1}.

Similar with the first approach, the correlation between each two superpixels is measured by the overlapping ratio between the warped version of s_j^{t-1} following motion vectors and s_i^t and the reliability of motion estimation inside s_j^{t-1} is measured by the motion

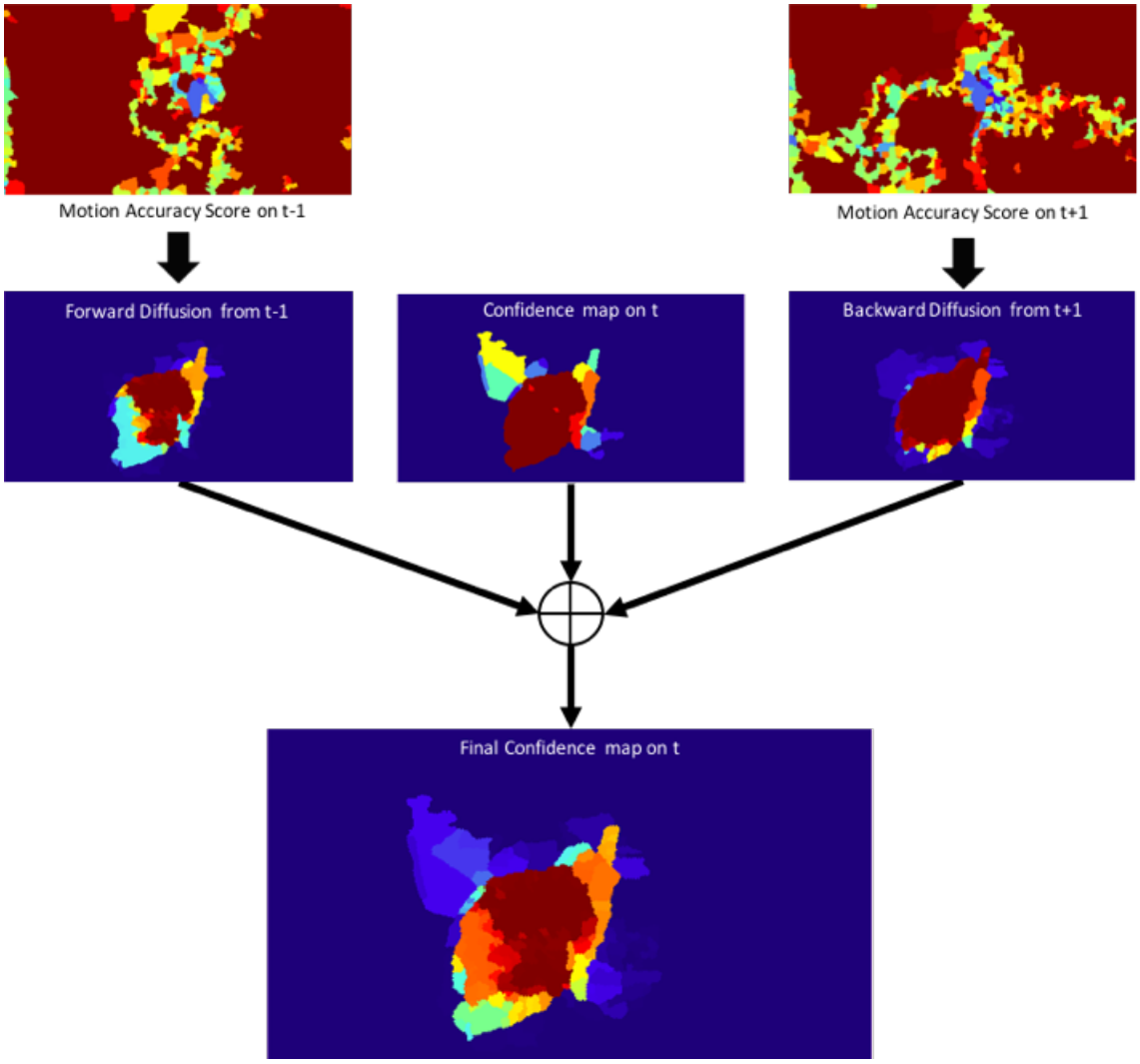


Figure 33. The amount of diffused confidence from neighbouring frames depends not only on the temporal correlation but also the per-superpixel motion accuracy.

non-coherence. We combine these two factors to update the confidence c_i^t of s_i^t,

$$\begin{aligned} c_i^t &= w_u c_i^t + 0.5 \cdot (1 - w_u)(\delta_i^{t-1} + \delta_i^{t+1}), \\ \delta_i^{t-1} &= \frac{\sum_{s_j \in \mathcal{S}^{t-1}} w_{j,i} \cdot c_j^{t-1}}{\sum_{s_j \in \mathcal{S}^{t-1}} w_{j,i}}, \\ \delta_i^{t+1} &= \frac{\sum_{s_k \in \mathcal{S}^{t+1}} w_{k,i} \cdot c_j^{t+1}}{\sum_{s_k \in \mathcal{S}^{t+1}} w_{k,i}}, \\ w_{j,i} &= \rho_{j,i} \cdot m_j, \quad w_{k,i} = \rho_{k,i} \cdot m_k. \end{aligned} \tag{16}$$

We iteratively update the confidence of superpixels of all frames. The diffusion process is shown in Fig. 33

5 Video Object Segmentation

We start describing our video object segmentation method by firstly introducing the fundamentals of conditional random fields (CRF), followed by the formulation of our solution based on CRF.

5.1 Preliminaries

We define a discrete random field consisting of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ without loop edges, a finite set of labels $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, and a probability distribution P on the space $\mathcal{X}$ of label assignments. $x \in \mathcal{X}$ is a map that assigns to each vertex v a label x_v in $\mathcal{L}$.

A clique c is a set of vertices in graph $\mathcal{G}$ where every vertex has an edge to every other vertex. A random field is said to be Markov if it satisfies the Markovian property:

$$P(x_v|x_{\mathcal{V}\setminus v}) = P(x_v|x_{N_v}), \tag{17}$$

where $P(\mathrm{x}) > 0 \ \forall \mathrm{x} \in \mathcal{L}$ and N_v denote the set of neighbouring vertex v, i.e., $\{u \in \mathcal{V}|(u, v) \in E\}$. This indicates the property that the assignment of a label to a vertex is only conditionally dependent on the assignment to other neighbouring vertices.

An energy function $E: \ \mathcal{L} \rightarrow \mathbb{R}$ maps any labelling $x \in \mathcal{L}$ to a real number $E(\mathrm{x})$ called its energy. The energy function is formulated as the negative logarithm of the posterior probability distribution of the labelling, and its minimisation is equivalent to maximising a posteriori probability (MAP) x^* of a random field which is defined as

$$\mathrm{x}^* = \mathrm{argmin}_{\mathrm{x}\in\mathcal{L}} E(\mathrm{x}). \tag{18}$$

The posterior distribution over the labellings of CRF is represented as Gibbs energy

$$E(\mathrm{x}) = \sum_{c\in\mathcal{C}} \psi_c(\mathrm{x}_c). \tag{19}$$

where $\mathcal{C}$ is the set of all cliques, $\psi_c(x_c)$ denotes the potential function of the clique c, and $\mathrm{x}_c = \{x_i, i \in c\}$.

5.2 Formulation

Video object segmentation is formulated as a superpixel-labelling problem of assigning each superpixel two classes: objects and background (not listed in the keywords). Similar to Sec. 4.1, we define a graph of superpixels $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ by connecting frames temporally with optical flow motion vectors, as illustrated in Fig. 34.

We achieve the optimal labelling by minimising the following energy function:

$$E(x) = \sum_{i \in \mathcal{V}} (\psi_i^c(x_i) + \lambda_o \psi_i^o(x_i)) + \lambda_s \sum_{i \in \mathcal{V}, j \in N_i^s} \psi_{i,j}^s(x_i, x_j) + \lambda_t \sum_{i \in \mathcal{V}, j \in N_i^t} \psi_{i,j}^t(x_i, x_j) \quad (20)$$

where N_i^s and N_i^t are the sets of superpixels adjacent to superpixel s_i spatially and temporally in the graph respectively; λ_o, λ_s and λ_t are parameters; $\psi_i^c(x_i)$ indicates the colour based unary potential and $\psi_i^o(x_i)$ is the unary potential of semantic object confidence which measures how likely the superpixel to be labelled by x_i given the semantic confidence map; $\psi_{i,j}^s(x_i, x_j)$ and $\psi_{i,j}^t(x_i, x_j)$ are spatial pairwise potential and temporal pairwise potential respectively. We set parameters $\lambda_o = 10$, $\lambda_s = 1000$ and $\lambda_t = 2000$ empirically. The definitions of these unary and pairwise terms are explained in detail next.

5.3 Unary Potentials

We define unary terms to measure how likely a superpixel is to be labelled as the background or the object of interest according to both the appearance model and semantic object confidence map.

Colour unary potential is defined similar to [86], which evaluates the fit of a colour distribution (of a label) to the colour of a superpixel,

$$\psi_i^c(x_i) = -\log U_i^c(x_i)$$

where $U_i^c(\cdot)$ is the colour likelihood from the colour model. We train two gaussian mixture models (GMMs) over the average RGB values of superpixels, for objects and background respectively. These GMMs are estimated by sampling the superpixel colours according to the semantic confidence map.

Semantic unary potential is defined to evaluate how likely the superpixel to be labelled by

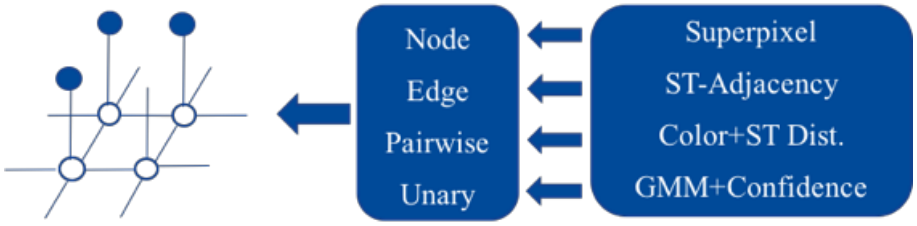


Figure 34. Graph construction for video object segmentation which takes into account both the spatial and temporal properties of video data. Colour model and semantic confidence maps are employed to model the object.

x_i given the semantic confidence map c_i^t

$$\psi_i^o(x_i) = -\log U_i^o(x_i)$$

where $U_i^o(\cdot)$ is the semantic likelihood, i.e., for an object labelling $U_i^o = c_i^t$ and $1 - c_i^t$ otherwise.

5.4 Pairwise Potentials

We define the pairwise potentials to encourage both spatial and temporal smoothness of labelling while preserving discontinuity in the data. These terms are defined similar to the affinity matrix in Sec. 4.1.

Superpixels in the same frame are spatially connected if they are adjacent. The spatial pairwise potential $\psi_{i,j}^s(x_i, x_j)$ penalises different labels assigned to spatially adjacent superpixels:

$$\psi_{i,j}^s(x_i, x_j) = \frac{[x_i \neq x_j]\exp(-d^c(s_i, s_j))}{d^s(s_i, s_j)}$$

where $[\cdot]$ denotes the indicator function.

The temporal pairwise potential is defined over edges where superpixels are temporally connected on consecutive frames. Superpixels s_i^{t-1} and s_j^t are deemed as temporally connected if there is at least one pixel of s_i^{t-1} is propagated to s_j^t following the optical flow motion vectors,

$$\psi_{i,j}^t(x_i, x_j) = \frac{[x_i \neq x_j]\exp(-d^c(s_i, s_j))}{d^t(s_i, s_j)}.$$

Taking advantage of the similar definitions in computing affinity matrix in Sec. 4.1, the

pairwise potentials can be efficiently computed by reusing the affinity in Eq. 5 and 8.

6 Experiments and Results

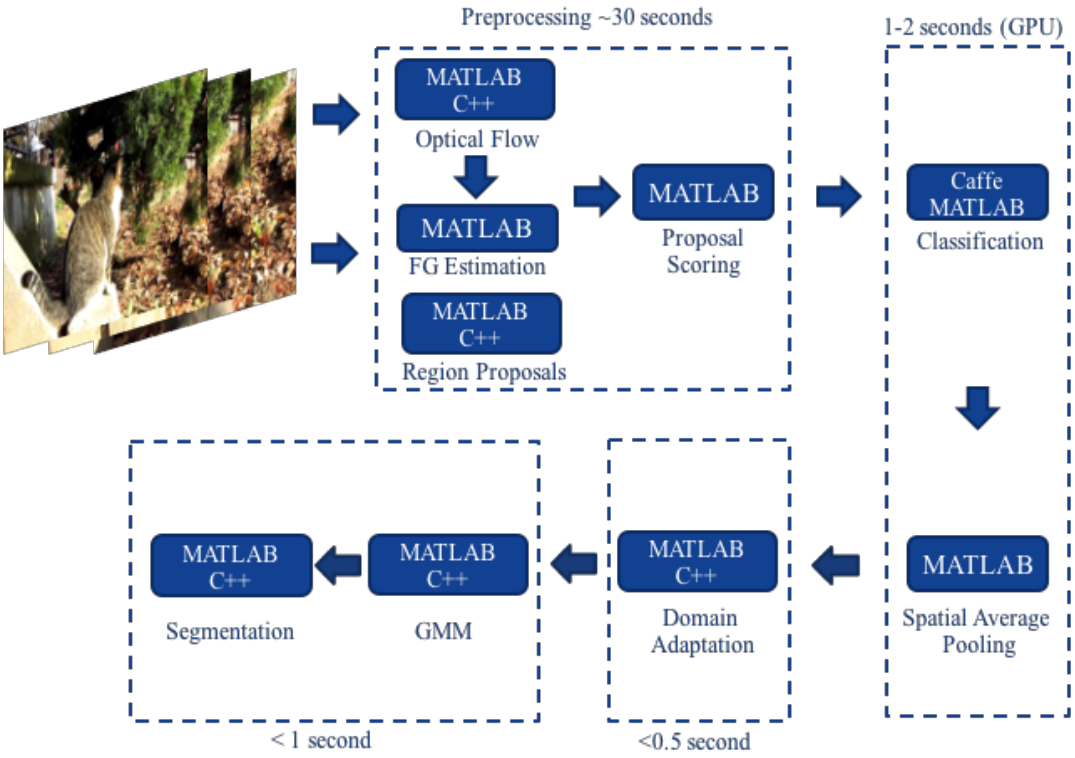
We adopt alpha expansion [87] which is a multi-label inference algorithm for CRF formulation to minimise Eq. 20 and the inferred labels gives the semantic object segmentation of the video. We implement our method using MATLAB and C/C++, with Caffe [88] implementation of VGG-16 net [10]. We reuse the superpixels returned from [13] which is produced by [89]. Large displacement optical flow algorithm [25] is adopted to cope with strong motion in natural videos. 5 components per GMM in RGB colour space are learned to model the colour distribution following [86], although unsupervised approach exists [90] for learning a finite mixture model from multivariate data. Our domain adaptation method performs efficient learning on superpixel graph with an unoptimised MATLAB/C++ implementation, which takes around 30 seconds over a video shot of 100 frames. The average time on segmenting one preprocessed frame is about three seconds on a commodity desktop with a Quad-Core 4.0 GHz processor, 16 GB of RAM, and GTX 980 GPU. Specific timing information of each module in two proposed domain adaptation approaches is shown in Fig. 35.

We set parameters by optimizing segmentation in multiple runs against labelling ground truth over a sampled set of 5 videos from publicly available *Freiburg-Berkeley Motion Segmentation Dataset* dataset [91] and these parameters are fixed for the evaluation.

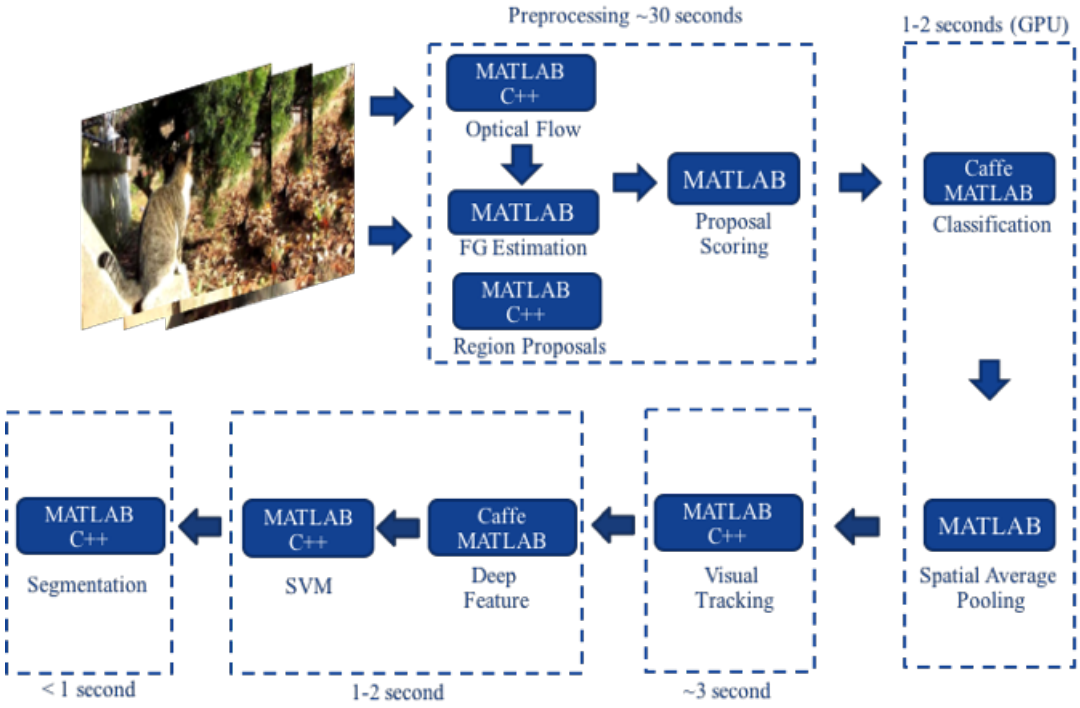
We evaluate our method on a large scale video dataset YouTube-Objects [92] and SegTrack [42]. YouTube-Objects consists of videos from 10 object classes with pixel-level ground truth for every 10 frames of 126 videos provided by [93]. These videos are very challenging and completely unconstrained. SegTrack consists of 5 videos with one or more objects presented in each video.

6.1 YouTube-Objects Dataset

We quantify the segmentation performance using the standard *intersection-over-union* (IoU) overlap as the accuracy metric. We compare our approach with 6 state-of-the-art automatic approaches on this dataset, including two motion driven segmentation [2, 6], three weakly supervised approaches [54, 56, 92], and state-of-the-art object-proposal based approach [3]. Among the compared approaches, [2, 3] reported their results by fitting a bounding box to the largest connected segment and overlapping with the ground truth bounding box; the result of [3] on this dataset is originally reported by [6] by testing



(a) Approach I



(b) Approach II

Figure 35. Timing information of two domain adaptation approaches.

on 50 videos (5/class). We include the result of [3] even it does not report per class performance, because there have been few state-of-the-art methods which reported pixel-

Table 1. IoU accuracies on YouTube-Objects Dataset

	Brox [2]	Lee [3]	Prest [92]	Papazoglou [6]	Tang [54]	Zhang [56]	Baseline	App. I	App. II
Plane	0.539	NA	0.517	0.674	0.178	0.758	0.693	0.757	**0.760**
Bird	0.196	NA	0.175	0.625	0.198	0.608	0.590	0.658	**0.747**
Boat	0.382	NA	0.344	0.378	0.225	0.437	0.564	**0.656**	0.588
Car	0.378	NA	0.347	0.670	0.383	**0.711**	0.594	0.650	0.659
Cat	0.322	NA	0.223	0.435	0.236	0.465	0.455	0.514	**0.557**
Cow	0.218	NA	0.179	0.327	0.268	0.546	0.647	**0.714**	0.675
Dog	0.270	NA	0.135	0.489	0.237	0.555	0.495	0.570	**0.574**
Horse	0.347	NA	0.267	0.313	0.140	0.549	0.486	0.567	**0.575**
Mbike	0.454	NA	0.412	0.331	0.125	0.424	0.480	0.560	**0.569**
Train	0.375	NA	0.250	0.434	0.404	0.358	0.353	0.392	**0.430**
Cls. Avg.	0.348	0.28	0.285	0.468	0.239	0.541	0.536	0.604	**0.613**
Vid. Avg.	NA	NA	NA	0.432	0.228	0.526	0.523	0.592	**0.600**

level evaluations on YouTube-Objects. The performance of [6] measured with respect to segmentation ground truth is reported by [56]. Zhang *et al.* [56] reported results in more than 5500 frames sampled in the dataset based on the segmentation ground truth. Wang *et al.* [51] reported the average results on 12 randomly sampled videos in terms of a different metric, i.e., per-frame pixel errors across all categories, and thus not listed here for comparison. We report both class and video average results which are the average accuracies over all classes and all videos respectively.

As shown in Table 1, our method surpasses the competing methods in 7 out of 10 classes, with gains up to 6.3%/6.6% and 7.2%/7.4% in category/video average accuracies over the best competing method [56] by the proposed two approaches respectively. This is remarkable considering that [56] employed strongly-supervised deformable part models (DPM) as object detector while our approach only leverages image recognition model which lacks the capability of localizing objects. [56] outperforms our method on *Car*, otherwise exhibiting varying performance across the categories — higher accuracy on more rigid objects, but lower accuracy on highly flexible and deformable objects such as *Cat* and *Dog*. We owe it to that, though based on object detection, [56] prunes noisy detections and regions by enforcing spatio-temporal constraints, rather than learning an adapted data-driven representation in our approach. It is also worth remarking on the improvement in classes, e.g., *Cow*, where the existing methods normally fail or underperform due to the heavy reliance on motion information. The main challenge of the *Cow* videos is that cows very frequently stand still or move with mild motion, which the existing approaches might fail to capture whereas our proposed method excels by leveraging the recognition and representation power of deep convolutional neural network, as well as the semi-supervised domain adaptation.

Interestingly, another weakly supervised method [54] slightly surpasses our first approach on *Train* although all methods do not perform very well on this category due to the slow motion and missed detections on partial views of trains. This is probably owing to that [54] uses a large number of similar training videos which may capture objects in rare view. Otherwise, our method doubles or triples the accuracy of [54]. Motion driven method [6] can better distinguish rigid moving foreground objects on videos exhibiting relatively clean backgrounds, such as *Plane* and *Car*.

Comparing with the baseline scheme which excludes the domain adaptation component, we can see the proposed two domain adaptation approaches are able to learn to successfully compensate the shift to the target with a gain of 6.8%/6.9% and 7.7%/7.7% in category/video average accuracies. The proposed Approach II slightly surpasses Approach I with 0.9%/0.8% in category/video average accuracies. One possible explanation might be the benefits of explicitly modelling the objects in deep feature space in the second approach. Representative qualitative segmentation results are shown in Fig. 36-45, where the segmentation results from our method are shown in green contours following the existing works. The qualitative segmentation results demonstrate the accurate localisation of objects and their boundaries, as well as the spatial-temporal stabilities of segmentations on challenging videos.

As a failure case, the motorbikes in Fig. 44 are under-segmented, forming a single segment with the rider. The reason which causes this failure is due to the under-segmentation by the initial object proposals on areas exhibiting similar colours, whereas our approaches are agnostic to the particular region proposal method — finding the better ones to further improve the quality of our methods is out of the scope of this thesis.

Figure 36. Exemplar qualitative segmentation results on the sequence from the “Plane” class.

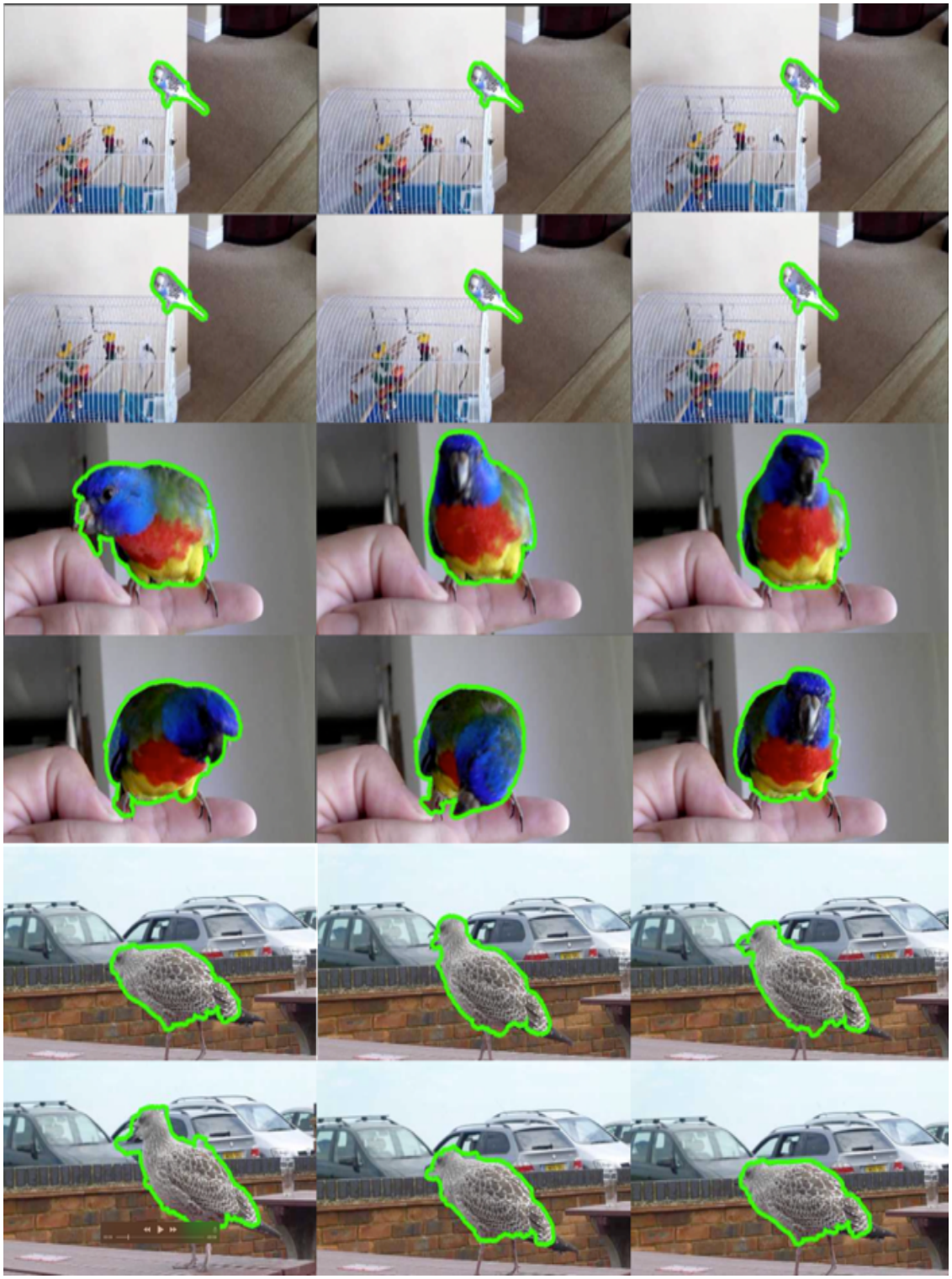


Figure 37. Exemplar qualitative segmentation results on the sequence from the “Bird” class.

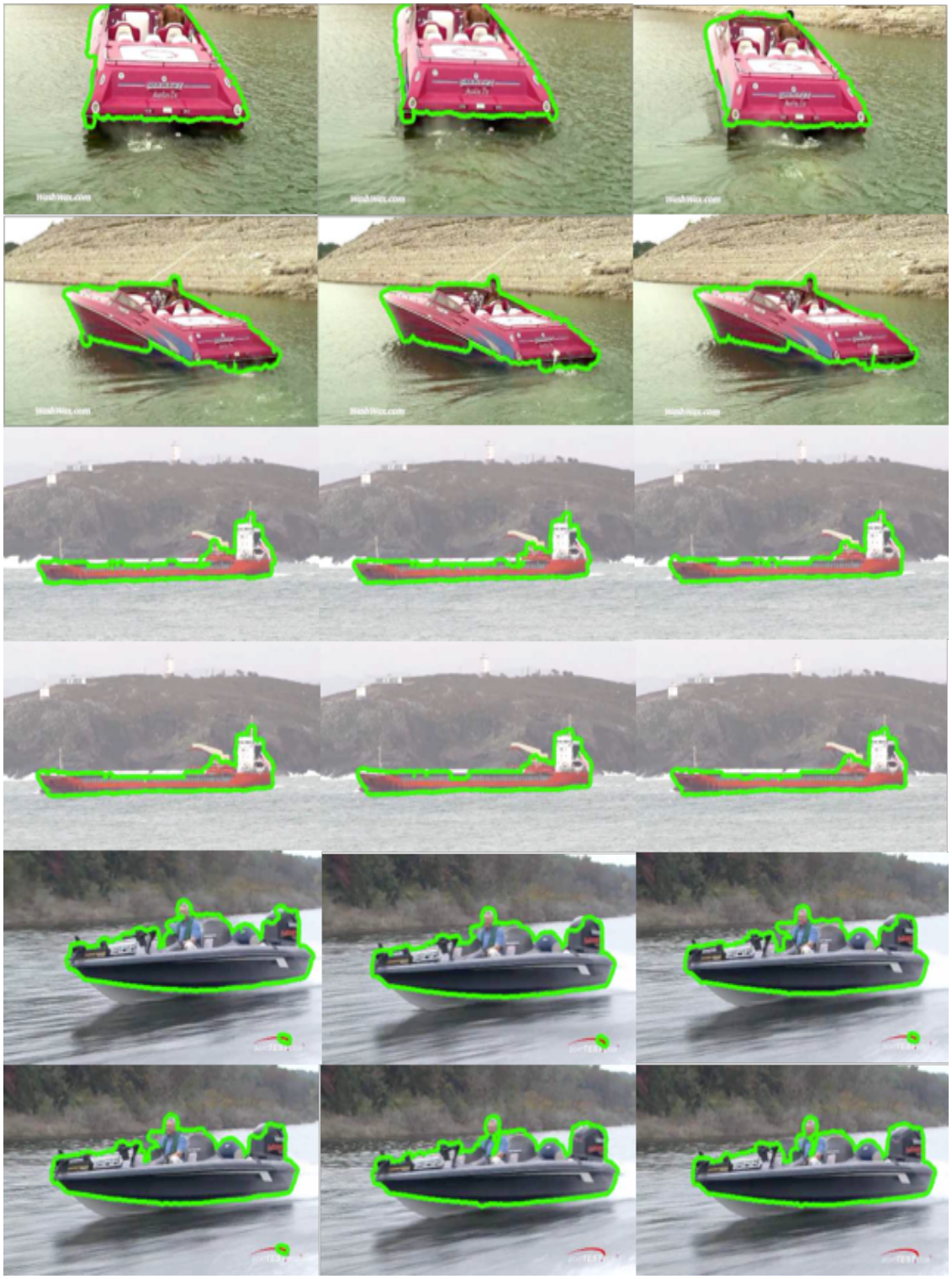


Figure 38. Exemplar qualitative segmentation results on the sequence from the “Boat” class.

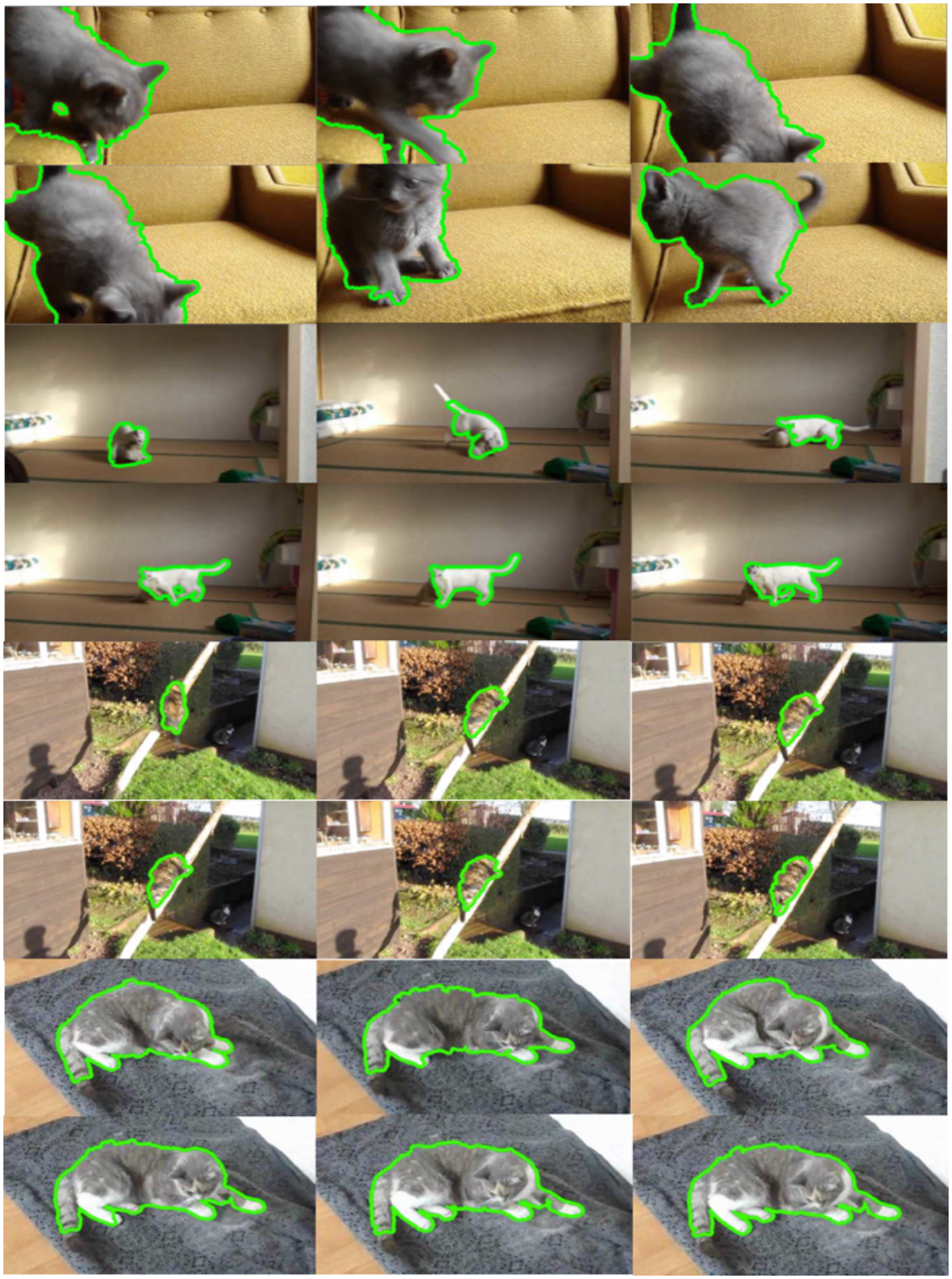


Figure 39. Exemplar qualitative segmentation results on the sequence from the “Cat” class.

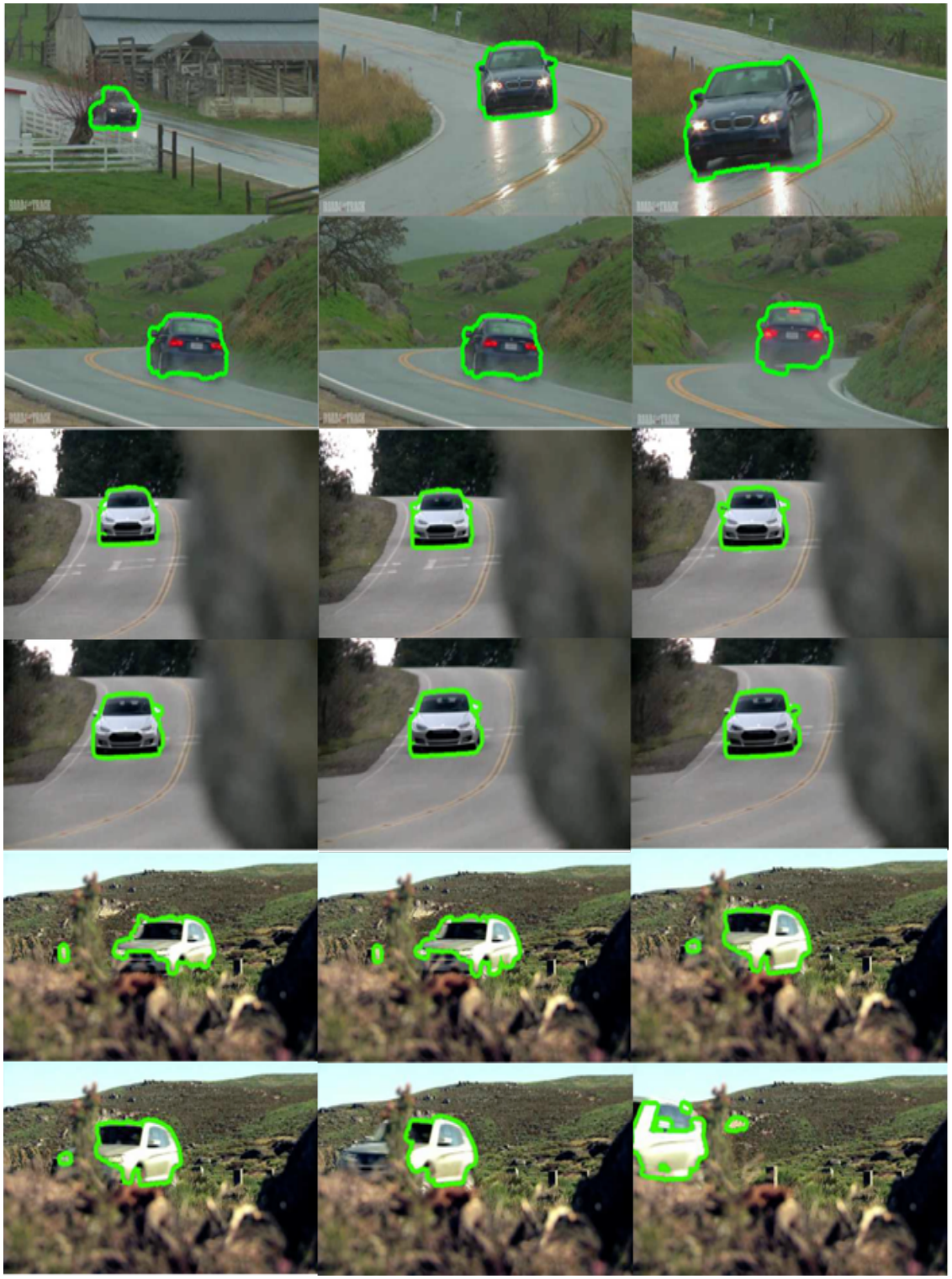


Figure 40. Exemplar qualitative segmentation results on the sequence from the “Car” class.

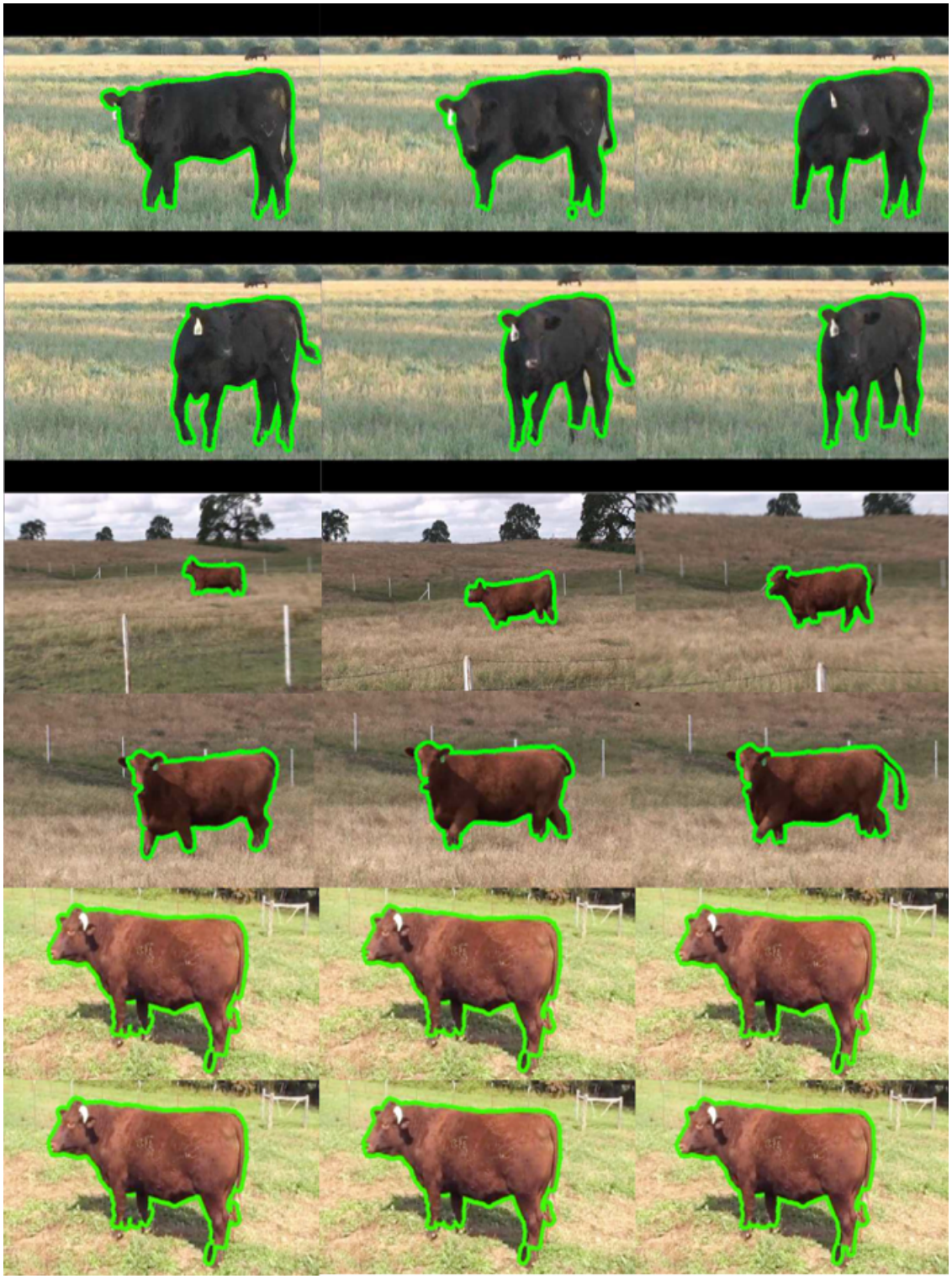


Figure 41. Exemplar qualitative segmentation results on the sequence from the “Cow” class.

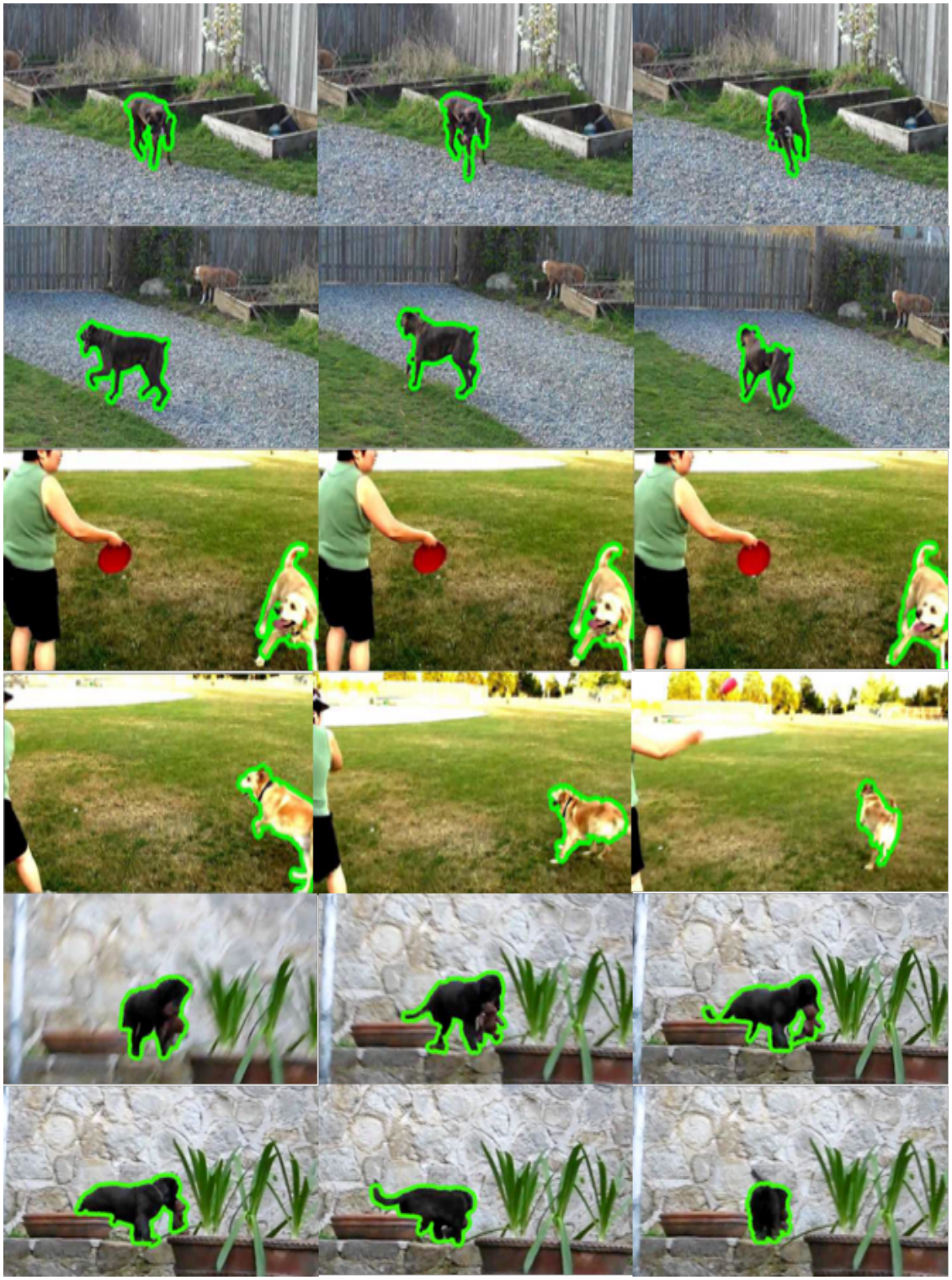


Figure 42. Exemplar qualitative segmentation results on the sequence from the “Dog” class.

Figure 43. Exemplar qualitative segmentation results on the sequence from the “Horse” class.

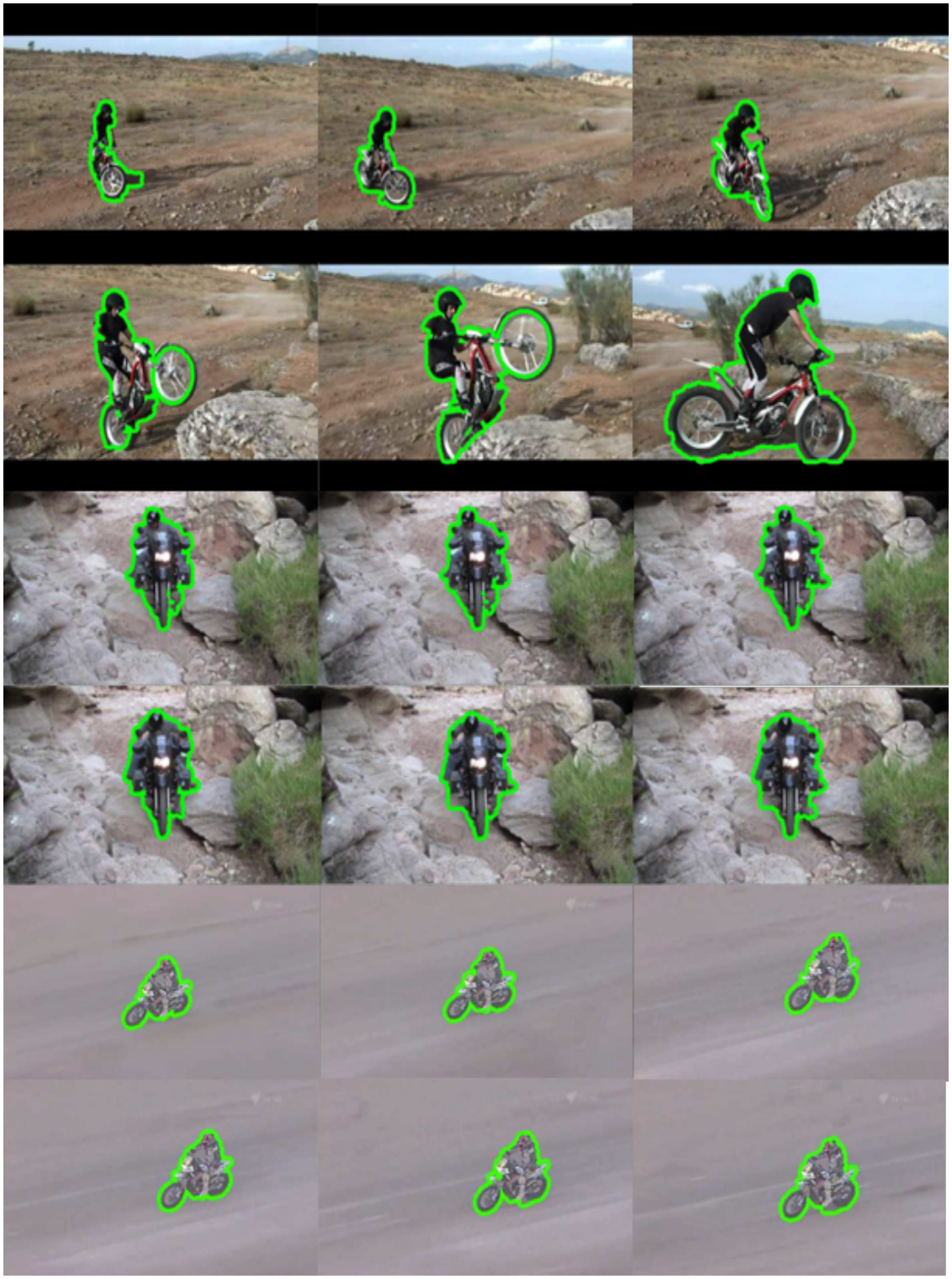


Figure 44. Exemplar qualitative segmentation results on the sequence from the “MBike” class.

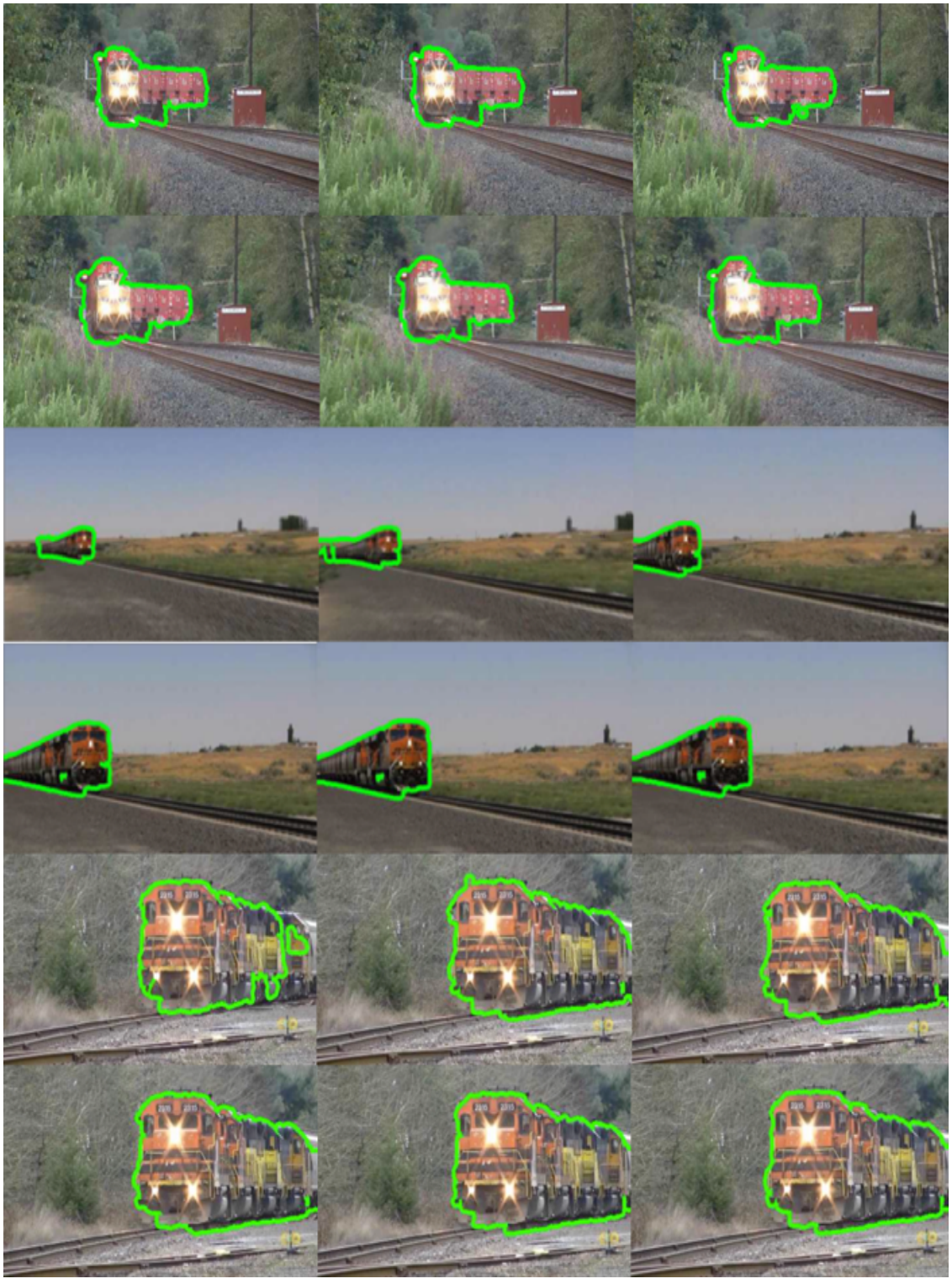


Figure 45. Exemplar qualitative segmentation results on the sequence from the “Train” class.

Table 2. Quantitative segmentation results on SegTrack measured by the average number of incorrect pixels per frame.

Video (No. frames)	App. I	App. II	[2]	[6]	[5]	[3]	[56]
birdfall (30)	170	224	468	217	**155**	288	339
cheetah (29)	826	858	1968	890	**633**	905	803
girl (21)	1647	1747	7595	3859	1488	1785	**1459**
monkeydog (71)	304	**282**	1434	284	472	521	365
parachute (51)	363	346	1113	855	220	201	**196**

6.2 SegTrack Dataset

We evaluate on SegTrack dataset to compare with the representative state-of-the-art unsupervised object segmentation algorithms [2, 3, 5, 6, 56]. Note that, most methods compared on SegTrack are Figure-Ground segmentation methods rather than semantic video object segmentation methods. We only compare with the most representative Figure-Ground segmentation methods following [56] as baseline. To avoid confusion of segmentation results, all the compared methods only consider the primary object.

As shown in Table 2, our two approaches outperform weakly supervised method [56] on *birdfall* and *monkeydog* videos, motion driven method [6] on four out of five videos, and proposal ranking method [3] on four videos. Clustering point tracks based method [2] results in highest error among all the methods. Overall, our performance is about on par with weakly supervised method [56]. The proposal merging method [5] obtains the best results on two videos, yet it is sensitive to motion accuracy as reported by [7] on the other dataset. We also observe that approach II performs better on longer videos, i.e., *monkeydog* and *parachute*, which facilitate good representation learning of video objects.

Note that, due to the nature of the Figure-Ground segmentation, i.e., segmenting all moving foreground object without assigning semantic labels, these methods [2, 3, 5, 6] are dealing with a much less challenging problem comparing with our goal. The performance of the Figure-Ground segmentation methods has been optimized to detect motion or saliency driven cues for segmenting single object. We believe that the progress on this dataset is plateaued due to the limited number of available video sequences and frames. Representative qualitative segmentations of our approaches are shown in Fig. 46.

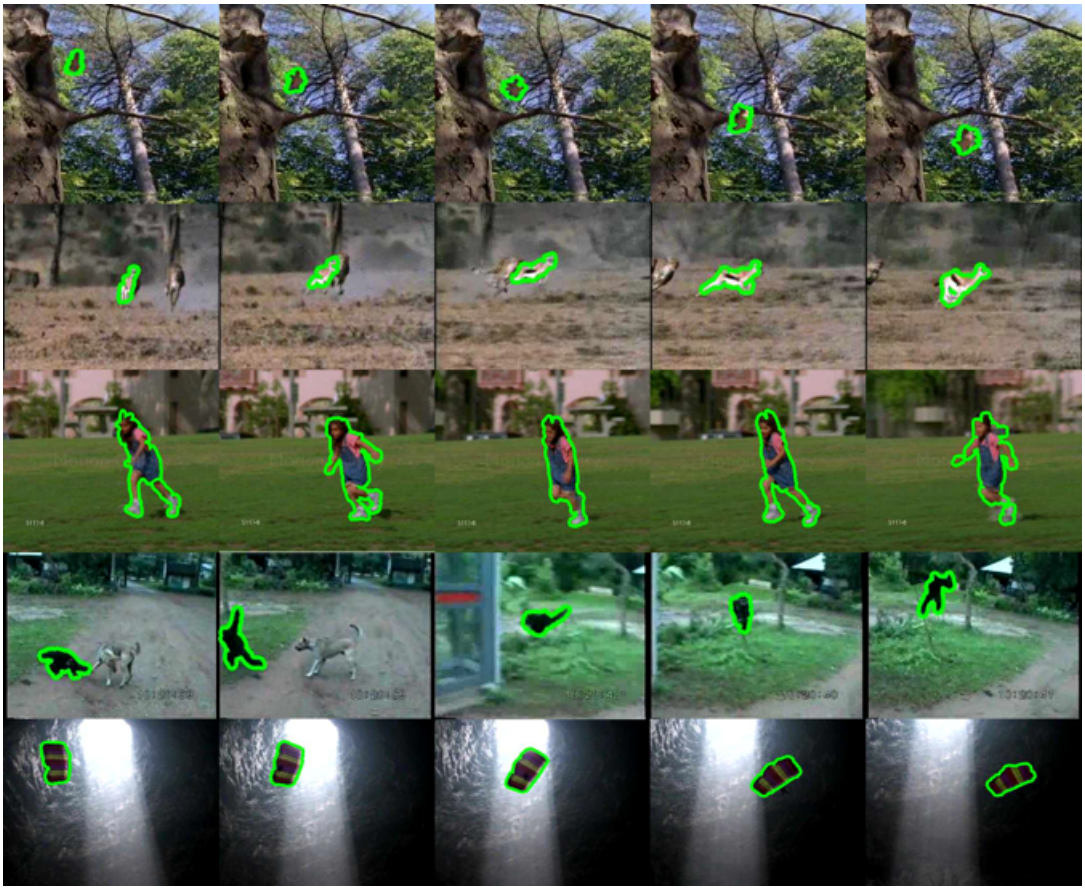


Figure 46. Qualitative results of our method on SegTrack dataset.

6.3 Future Work

In our first approach, the affinity matrix controls the confidence propagation in the linear system which is computed based on colour, spatial and temporal distances. As a future work, it would be interesting to incorporate representations learned from higher layers of CNN into the domain adaptation, which might potentially improve adaptation by propagating and combining higher level context. It would also be interesting to investigate how deep features would improve the unary and pairwise potentials, accounting for higher level contextual information and textures.

Due to the limited extracted training samples from a single video in our second approach, we decoupled the representation learning and classification in a CNN model. As a future work, we would like to investigate the possibility of retraining a fully convolutional network using extracted training samples from multiple weakly labelled videos, as shown in Fig. 47. The benefits are twofold: firstly, incorporating extracted training samples from multiple videos of the same class generalises our approach to a wider variety of objects; secondly, this approach is applicable to live streaming videos where future frames are

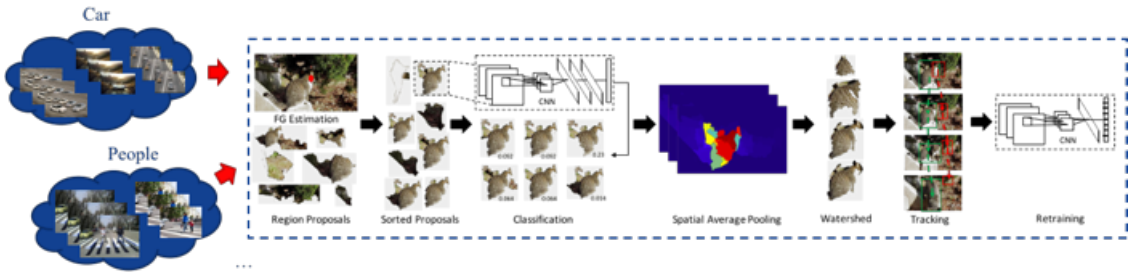


Figure 47. Illustration of applying approach II on multiple weakly labelled videos.

not available for learning a complete representation, since retrained fully convolutional network is capable of producing per-frame segmentation on incoming video frames.

7 Conclusion

We have proposed two semi-supervised frameworks to adapt CNN classifiers from image recognition domain to the target domain of semantic video object segmentation. These frameworks combine the recognition and representation power of the CNN with the intrinsic structure of unlabelled data in the target domain to improve inference performance, imposing spatio-temporal smoothness constraints on the semantic confidence over the unlabelled video data. This proposed domain adaptation framework enables learning a data-driven representation of video objects. We demonstrated that this representation underpins a robust semantic video object segmentation method which achieves the state-of-the-art performance comparing with the existing semantic video object segmentation methods on challenging datasets.

REFERENCES

[1] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1219–1225, 2009.

[2] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, pages 282–295, 2010.

[3] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1995–2002, 2011.

[4] Tianyang Ma and Longin Jan Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–677, 2012.

[5] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, 2013.

[6] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[7] Tinghuai Wang and Huiling Wang. Graph transduction learning of object proposals for video object segmentation. In *Asian Conference on Computer Vision*, pages 553–568, 2014.

[8] Tinghuai Wang and Huiling Wang. Primary object discovery and segmentation in videos via graph-based transductive inference. *Computer Vision and Image Understanding*, 143(2):159–172, 2016.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 1106–1114, 2012.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[12] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. In *Annual Conference on Neural Information Processing Systems*, 2015.

[13] Ian Endres and Derek Hoiem. Category independent object proposals. In *European Conference on Computer Vision*, pages 575–588, 2010.

[14] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1879–1886, 2011.

[15] Santiago Manen, Matthieu Guillaumin, and Luc J. Van Gool. Prime object proposals with randomized prim's algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2536–2543, 2013.

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[17] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2010.

[18] João Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3241–3248, 2010.

[19] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr. BING: binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014.

[20] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[21] Dan Oneata, Jérome Revaud, Jakob J. Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *European Conference on Computer Vision*, pages 737–752, 2014.

[22] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090, 2015.

[23] Gilad Sharir and Tinne Tuytelaars. Video object proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, 2012.

[24] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical symposium east*, pages 319–331. International Society for Optics and Photonics, 1981.

[25] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36, 2004.

[26] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[27] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[28] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[29] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[30] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550. IEEE, 2010.

[31] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*. BMVA Press, 2014.

[32] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.

[33] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[34] Hanxi Li, Yi Li, Fatih Porikli, et al. DeepTrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In *British Machine Vision Conference*, volume 1, page 3, 2014.

[35] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

[36] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. *arXiv preprint arXiv:1502.06796*, 2015.

[37] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. *arXiv preprint arXiv:1510.07945*, 2015.

[38] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[39] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. *ACM Transactions on Graphics*, 23(3):574–583, 2004.

[40] John P. Collomosse, David Rowntree, and Peter M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Transaction Visualization and Computer Graphics*, 11(5):540–549, 2005.

[41] Tinghuai Wang and John P. Collomosse. Probabilistic motion diffusion of labeling priors for coherent video segmentation. *IEEE Transactions on Multimedia*, 14(2):389–400, 2012.

[42] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M. Rehg. Motion coherent tracking using multi-label mrf optimization. *International Journal of Computer Vision*, 100(2):190–202, 2012.

[43] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision, Australia, December 1-8, 2013*, pages 2192–2199, 2013.

[44] Tinghuai Wang, Bo Han, and John P. Collomosse. TouchCut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding*, 120:14–30, 2014.

[45] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa. Efficient hierarchical graph-based video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2141–2148, 2010.

[46] Chenliang Xu, Caiming Xiong, and Jason J. Corso. Streaming hierarchical video segmentation. In *European Conference on Computer Vision (6)*, pages 626–639, 2012.

[47] Chaohui Wang, Martin de La Gorce, and Nikos Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 747–754, 2009.

[48] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbelaez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2233–2240, 2011.

[49] Daniela Giordano, Francesca Murabito, Simone Palazzo, and Concetto Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4814–4822, 2015.

[50] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015.

[51] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3402, 2015.

[52] Jiong Yang, Gangqiang Zhao, Junsong Yuan, Xiaohui Shen, Zhe Lin, Brian Price, and Jonathan Brandt. Discovering primary objects in videos by saliency fusion and iterative appearance estimation. *IEEE Transaction on Circuits and Systems for Video Technology*, 2015.

[53] Glenn Hartmann, Matthias Grundmann, Judy Hoffman, David Tsai, Vivek Kwatra, Omid Madani, Sudheendra Vijayanarasimhan, Irfan A. Essa, James M. Rehg, and Rahul Sukthankar. Weakly supervised learning of object segmentations from web-scale video. In *European Conference on Computer Vision Workshop*, pages 198–208, 2012.

[54] Kevin D. Tang, Rahul Sukthankar, Jay Yagnik, and Fei-Fei Li. Discriminative segment annotation in weakly labeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2483–2490, 2013.

[55] Xiao Liu, Dacheng Tao, Mingli Song, Ying Ruan, Chun Chen, and Jiajun Bu. Weakly supervised multiclass video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 57–64, 2014.

[56] Yu Zhang, Xiaowu Chen, Jia Li, Chen Wang, and Changqun Xia. Semantic object segmentation via detection in weakly labeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3641–3649, 2015.

[57] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[58] David H. Hubel and Torsten N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195:215–243, 1968.

[59] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[61] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–130, 1988.

[62] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[64] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[65] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Annual Conference on Neural Information Processing Systems*, pages 809–817. 2013.

[66] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Annual Conference on Neural Information Processing Systems*, pages 487–495, 2014.

[67] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014.

[68] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[70] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *MM*, pages 643–650. ACM, 2015.

[71] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Annual Conference on Neural Information Processing Systems*, pages 91–99. 2015.

[72] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[73] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[74] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *International Conference on Machine Learning*, pages 737–744. ACM, 2009.

[75] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *European Conference on Computer Vision*, pages 140–153. Springer, 2010.

[76] David Stavens and Sebastian Thrun. Unsupervised learning of invariant features using video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1649–1656. IEEE, 2010.

[77] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3361–3368. IEEE, 2011.

[78] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *Annual Conference on Neural Information Processing Systems*, pages 3212–3220, 2012.

[79] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

[80] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[81] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4086–4093, 2015.

[82] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[83] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2015.

[84] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013.

[85] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Sch. Learning with local and global consistency. In *Annual Conference on Neural Information Processing Systems*, pages 321–328, 2004.

[86] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[87] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[88] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[89] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301, 2009.

[90] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.

[91] Peter Ochs, Jagannath Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.

[92] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289, 2012.

[93] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*, pages 656–671. Springer, 2014.