Towards Computational Baby Learning: A Weakly-supervised Approach for Object Detection

Xiaodan Liang[1,2] Si Liu[1,4] Yunchao Wei[1,3] Luoqi Liu[1] Liang Lin[2] Shuicheng Yan [1*]
[1] National University of Singapore [2] Sun Yat-sen University [3] Beijing Jiaotong university
[4] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences

Abstract

Intuitive observations show that a baby may inherently possess the capability of recognizing a new visual concept (e.g., chair, dog) by learning from only very few positive instances taught by parent(s) or others, and this recognition capability can be gradually further improved by exploring and/or interacting with the real instances in the physical world. Inspired by these observations, we propose a computational model for weakly-supervised object detection, based on prior knowledge modelling, exemplar learning and learning with video contexts. The prior knowledge is modeled with a pre-trained Convolutional Neural Network (CNN). When very few instances of a new concept are given, an initial concept detector is built by exemplar learning over the deep features the pre-trained CNN. The well-designed tracking solution is then used to discover more diverse instances from the massive online weakly labeled videos. Once a positive instance is detected/identified with high score in each video, more instances possibly from different view-angles and/or different distances are tracked and accumulated. Then the concept detector can be fine-tuned based on these new instances. This process can be repeated again and again till we obtain a very mature concept detector. Extensive experiments on Pascal VOC-07/10/12 object detection datasets [9] well demonstrate the effectiveness of our framework. It can beat the state-of-the-art full-training based performances by learning from very few samples for each object category, along with about 20,000 weakly labeled videos.

1. Introduction

Empirically, we may have the following intuitive observations on how a baby learns: after the parent(s) or others teach the baby a few instances about a new concept, the initial recognition capability about the concept can be built[1]. During continuously exploring and/or interacting with diverse instances and scenes in real life, the baby can associate the initial simple instances with other variants by using various information linkages. Based on the accumulated instances about the concept, the baby can gradually improve its recognition capability and recognize diverse instances he/she never saw.

Recent successes in computer vision [33] [20] [41], however, largely on a large number of labeled instances of visual concepts, which may require considerable human efforts. The construction of an appearance-based object detector is costly and difficult because the number of training examples must be large enough to capture different variations in the object appearance. Some researchers have made efforts on improving the initial models by using very few labeled data, along with the detection/search results from web images [4] [8] [5] or weakly annotated videos [30] [2]. this paper, we build a computational model for weakly supervised object detection by drawing inspiration from the baby learning process. As illustrated in Figure 1, we propose a robust learning framework which can effectively model the prior knowledge, build the initial model by exemplar learning with very few positive instances for a new concept, and gradually learn a mature object detector by exploring more diverse instances in videos.

First, we model the prior knowledge (i.e. feature representation) with a pre-trained Convolutional Neural network (CNN) in two steps. We first train a generic CNN by the large image classification dataset. We then fine-tune the CNN with the instances of previously learned visual concepts for transferring object classification network into the detection network. Second, when very few positive instances of a new concept are given, the initial object detector is built by exemplar learning [25], which trains a separate linear classifier for every exemplar in the training set based

*This work was done when the first author worked as an intern in National University of Singapore.

[1]Note that it does not necessarily mean baby truly learns in this way from neuron-science perspective.

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.120

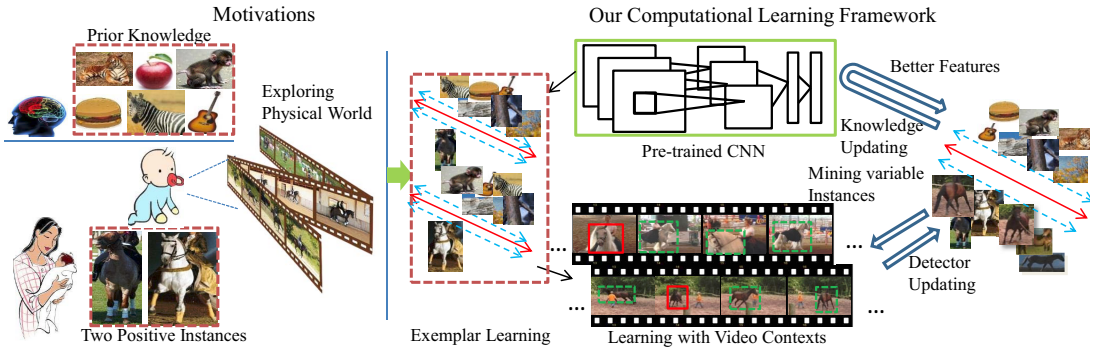

Figure 1. Illustration of our computational baby learning framework. Inspired by the baby learning process, we integrate prior knowledge modelling, exemplar learning, learning with video contexts for supervised object detection. The prior knowledge (i.e. feature representation) is modelled with a pre-trained CNN. When very few positive instances of a new concept (e.g., horse) are given, an initial concept detector can be built by exemplar learning. Once a positive instance in a frame is detected with the highest score, more instances (green dashed box) can be tracked by harnessing video contexts. The concept detector can be gradually improved with these new instances and we repeat this process again and again. In addition, the pre-trained CNN will be gradually fine-tuned if enough instances are collected, which leads to more informative features for training detectors.

on the deep features from the pre-trained CNN. Third, more instances can be mined by exploring the video clips from the online video sharing websites (e.g., YouTube.com). The positive instance with highest detection score in each clip is selected as the seed, and then region-based video tracking is performed to mine instances by considering the appearance consistency and spatial correspondence. The object detector can thus be progressively improved based on these newly tracked instances. After this process repeats again and again, a very mature object detector can be obtained. With enough instances for the new concept, the pre-trained CNN can also be further fine-tuned, which can provide better deep feature representation. The new object detector can be gradually improved in a never ending way as long as more videos are continuously explored.

Extensive experiments on three challenging object detection datasets (Pascal VOC 07/10/12) well demonstrate the superiority of our computational baby learning framework over other state-of-the-arts [14] [29] [38] [13]. For all three datasets, we only need to learn one detector for each concept, while all previous works train different models for different datasets. Our framework beats other state-of-the-arts by learning from very few positive instances along with about 20,000 videos for each object category.

The contributions of this paper can be summarized as the followings. 1) To the best of our knowledge, the proposed framework build an effective computational framework for weakly supervised object detection with inspiration from the baby learning process, where the prior knowledge modelling, exemplar learning and learning with video contexts are integrated. 2) Only two positive instances are required for learning a new concept detector and then the detector is refined with new diverse instances from YouTube videos crawled by key words. But all key words are not used in the computational baby learning process given that there is no guarantee that there must have a specific object if a corresponding key word is present. It makes our framework scalable and robust for learning concept detectors by utilizing large-scale videos. 3) The knowledge of learned concepts can be effectively retained in our model and conveniently utilized to learn new concepts.

2. Related Work

Supervised Learning. Recently, Convolutional Neural Networks (CNNs) have been shown to perform well in a variety of vision tasks with millions of annotated training images and thousands of categories, including classification [33], detection [14] and segmentation [10, 24]. Notably, Krizhevsky *et al.* [18] and Szegedy *et al.* [33] achieved great progress in the classification task with large and deep supervised CNN training. Girshick *et al.* [14] proposed to fine-tune the pre-trained Krizhevsky's network with the PASCAL VOC dataset and achieved the state-of-the-art object detection performance. However, the large performance increase achieved by these methods is only possible due to massive efforts on manually annotating millions of images.

Semi-Supervised Learning. To minimize human efforts, some attempts have been devoted to learning reliable models with very few labeled data. Those methods can be summarized into two categories: learning from unlabeled web images (image-based) or video data (video-based). For the first category, existing image-based approaches [4] [8] [31] [6] iteratively used image search and detection results to cover more variations. Also text-based [8] and semantic relationships [4] were further used

to provide more constraints on selecting instances. One problem with these approaches is that the data variations (e.g., different viewpoints or background clutters) cannot be effectively expanded when only with image-based visual similarities. Some other works proposed to transfer the annotated image-level labels [12] or ground-truth bounding boxes [36] from labeled images to unlabeled images for semantically related classes. However, still a lot of labeled images are required to build the adequate subspaces for knowledge transferring. For the second category, video-based approaches [28] [40] [37] [19] [34] [3] [15] [26] [23] utilized motion cues and appearance correlations within temporal adjacent frames to augment the model training. For example, [28] used videos with one class label while our method utilizes many unlabeled videos and very few seed instances. Yang *et al.* [40] used the pre-trained object detector to detect confident or hard samples. Different from [40], our method investigates how to utilize the video contexts to mine more informative instances. Our weakly supervised learning means that extremely scare annotated samples (e.g., one or two samples) are used, which is a special case of semi-supervised learning. A very recent paper [26] proposed a similar semi-supervised learning approach that iteratively learns and labels object instances from long videos. The main differences between our method with [26] lies in two aspects: first, our method used very few annotated seeds while [26] used sparsely annotated frames in videos; second, better feature representations are iteratively updated with more mined object instances, while [26] used the hand-crafted features.

One-shot Learning. Our learning framework is partially similar to the one-shot learning [11] which learns visual object classifiers by using very few samples. Most of the one-shot learning methods are based on the feature representation transfer [1], similar geometric context [17] or cross-modal knowledge transfer [32]. However, their performance is far from that of the state-of-the-art object classifiers. By continuously learning from video context, our framework can achieve the state-of-the-art detection results.

3. Computational Baby Learning Framework

Figure 1 shows our proposed framework. Inspired by intuitive observations of the baby learning process, our method integrates prior knowledge modelling, exemplar learning, learning with video contexts for weakly supervised object detection task. More specifically, the prior knowledge is modeled with a pre-trained CNN. Given very few instances for each new concept, an initial concept detector can be learned with exemplar learning over the deep features from the pre-trained CNN. More difficult instances can be obtained by exploring from real-world unlabelled videos. After that, the detector can be fine-tuned based on these new instances. This process is repeated again and again to obtain a mature detector. The pre-trained CNN can thus be fine-tuned to generate more informative features based on massive mined instances.

3.1. Prior Knowledge Modelling

We model the prior knowledge with two steps. First, we pre-train a general CNN on the ImageNet [7] with image-level annotations. Second, we fine-tune the previous pre-trained CNN with the previously learned concepts in ILSVRC2013 detection dataset for transferring the object classification network into detection network.

Network architectures. We explore two CNN architectures for pre-training: the 7-layer architecture by Krizhevsky *et al.* [18] and the Network in Network (NIN) proposed by Lin *et al.* [21]. We use the same parameter settings for these two network architectures as in [18] and [21]. The CNN fine-tuning starts SGD with a learning rate of 0.001 for both two networks. For the 7-layer architecture [18], we uniformly sample 32 positive windows (over all classes) and 96 background windows to construct a mini-batch of size 128. The fine-tuning is run for 70k SGD iterations and takes 9 hours on a single NVIDIA GeForce GTX TITAN GPU. For NIN [21], a mini-batch of size 80, consisting of 20 positive windows and 60 background windows, is used. The fine-tuning is run for 150k SGD iterations and takes 16 hours.

The usage of learned concepts. Since we validate our framework on the PASCAL VOC challenge, we thus use the 179 object classes on the ILSVRC2013 detection dataset as the learned concepts, which excludes the corresponding 21 classes related with the VOC 20 classes. During fine-tuning, we only replace the 1000-way classification layer of the pre-trained CNN with a randomly initialized (N+1)-way classification layer, where N is the number of learned concepts, plus one for background. In our setting, N = 179. We use the validation set (20,121 images) in the ILSVRC2013 detection dataset and only the images that contain at least one object of the 179 classes are used. All region proposals with ≥ 0.5 intersection-over-union (IoU) overlap with a ground-truth box are regarded as positives and the rest as negatives. Though our framework can use any category-independent region proposal method, we choose the selective search [35] to enable a controlled comparison with the previous work [14].

3.2. Exemplar Learning

The initial concept detector can be learned based on these deep features from pre-trained CNN and very few positive instances of a new concept.

Feature extraction. For all positive and negative instances, we enlarge the tight bounding boxes to contain 16 pixels of image contexts and then wrap it into a fixed 227×227 size as used in [14]. Deep features are computed as the

Figure 2. Some exemplar negative samples. Top row shows the collected general background images and bottom row shows the exemplar instances of previously learned concepts.

outputs from the penultimate fully-connected layer (4096-dimension) by forward propagating a mean-subtracted 227 $\times$ 227 image through the pre-trained CNN.

Selection of seed instances. Our selection strategy of the seeds (including the number of seeds) is optional and our framework can be bootstrapped with any number of seeds of the new concept. For most of our results, we select two common positive instances for each concept from the PASCAL VOC 2007 training set. Specifically, we cluster all positive instances into 10 clusters by k-means. For the top-2 largest clusters, the nearest two samples to the two centroids are selected as the seeds for each concept.

Negative set collection. To fairly justify our method, we do not use any annotations of PASCAL VOC Challenge to obtain the negative instances. The negative set used contains a batch of general background images (i.e., no specific object is included) and learned concept instances. As illustrated in the top row of Figure 2, we collect $4,000$ general scene images from Flickr and use the categories in the SUN scene dataset [39] as the search keywords. All region proposals in these background images are used as negative samples. For the learned concepts, the region proposals with ≥ 0.5 IoU overlap with the bounding box of 179 object class instances in the ILSVRC 2013 detection validation set are also treated as negative samples. Our initial experiment indicates that using general background images, versus our negative set, can result in about 4% drop in mAP. The possible reason may be that more hard negative samples are included in other object-level concept instances. After more instances of new concepts are collected, our negative set will be gradually enlarged.

Exemplar SVM training. Inspired by [25], we train a separate linear SVM classifier for each positive instance, and each SVM classifier is highly tuned according to the exemplar's appearance. The exemplar's decision boundary is thus decided, in large part, by the negative samples. For each test image, we thus independently run each exemplar detector and use the non-maximum suppression to create a final set of detections.

3.3. Learning with Video Contexts

We iteratively improve the concept detectors by mining more diverse instances from weakly labeled videos.

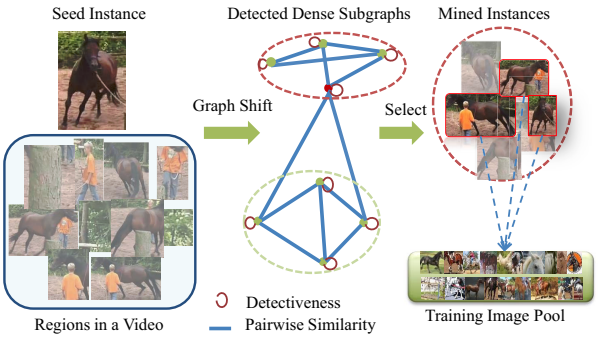

Figure 3. Region-based video tracking. Given the seed instance, we track other reliable instances from other regions. The affinity graph is built by combining both the pair-wise similarity and the detectiveness of each region. Then dense subgraphs are detected within the affinity graph by graph shift. The subgraph containing the seed instance (red point) is selected. Two instances with top-2 largest similarities with seed instances are placed into the training image pool for fine-tuning the detector in next iteration.

About 20,000 videos for each new concept are crawled from YouTube. Due to the computational limitation, we use the keywords from the VOC dataset collection to prune the videos unrelated to the new concept. The collected video set includes approximate 30% "noisy" videos that contain none of the instances of the concept. No manual annotation is performed. In each iteration, the region-based tracking is performed to accumulate more instances. The detector and knowledge updating are then performed.

Seed instance selection. In each video clip, there is much redundant information with few appearance differences in temporal adjacent frames. To guarantee appearance variance of tracked instances and limit computational complexity, only key frames of each video are analyzed. We select the image with ℓ_2 norm of the global GIST [27] feature difference larger than 0.01 as a key frame, compared with its temporal adjacent frames. For all key frames, we perform object detection with the initial detector. We only select the video containing the region with detection score larger than 1, and the region with the highest score in this video is selected as the seed positive instance.

Region-based video tracking. The region-based video tracking is performed on the selected videos and initialized with their seed instances as illustrated in Figure 3. In our framework, we treat the tracking task as a region-based cluster mining problem for both moving and static concepts. Specifically, we extract a batch of region proposals in all key frames using selective search [35] and represent each region r_i with both the deep feature $\mathbf{x_i}$ and the spatial coordinates $\mathbf{p_i} = (c_i^x, c_i^y, w_i, h_i)$ corresponding to the position, width and height. Since we wish to select the instances from different frames, which may capture more diverse visual patterns, the similarity of two regions from the same frame is thus set as zero. The similarity $A_{i,j}$ for each pair (r_i, r_j)

from different frames is thus defined by fusing the appearance similarity and the localization similarity,

$$A_{i,j} = \exp\{\frac{-||\mathbf{x_i} - \mathbf{x_j}||^2}{\delta_1^2}\} + \alpha(\exp\{\frac{-||\mathbf{p_i} - \mathbf{p_j}||^2}{\delta_2^2}\}), \quad (1)$$

where δ_1 and δ_2 are the empirical variances of $\mathbf{x}$ and $\mathbf{p}$, respectively, and we set $\alpha = 0.3$ because the appearance similarity is more important considering the camera moving and static objects. We normalize these two kinds of similarities by dividing their corresponding maximum to make them be comparable. In addition, to make our framework robust to outliers (i.e., noisy regions), we also constrain the detection score to enlarge the possibility of the region to contain the concept. thus use the detectiveness $O_i \in \{0, 1\}$ of each region r_i to indicate whether the region has high detection score. Specifically, by thresholding the detection score, O_i of the region with detection score larger than -3 is set as 1, otherwise as 0. Note that we do not directly use the detector scores because these diverse instances may not be detected by the current detector but can bring more data diversity for further improving the detectors.

To seamlessly integrate the detectiveness of the region and the pairwise similarity, we use the graph shift algorithm [22] to obtain more instances, which is efficient and robust for graph mode seeking. This algorithm is particularly suitable for our task as it directly operates on the affinity graph and leaves the outlier points ungrouped. Formally, we define an individual graph $G = (R, A)$ for each video. $R = \{r_1, \ldots, r_n\}$ represents all the regions and A is a symmetric matrix with non-negative elements. The diagonal elements of A represent the detectiveness of the regions while the non-diagonal elements measure the pairwise similarities between regions. The modes of a graph G are defined as local maximizers of the graph density function $g(y) = y^T Ay, y \in \Delta^n$, where $\Delta^n = \{y \in R^n : y \geq 0 \text{ and } ||y||_1 = 1\}$. The strongly connected subgraphs correspond to large local maxima of $g(y)$ over simplex which is an approximate measure of the average affinity score of these subgraphs. And these subgraphs can be found by solving the quadratic optimization problem, i.e., $\max g(y) = y^T Ay, y \in \Delta^n$, as in [22]. The graph shift algorithm starts from an individual sample and evolves towards the mode of G. The instances reaching the same mode are grouped as a cluster. We can thus select the target subgraph that contains the seed instance. To prevent the rapid semantic drift, we only select two instances in this subgraph, which appear in different frames and have highest similarities with the seed instance. We can thus accumulate much more instances to improve detectors iteratively. From our experiments, about 50 key frames are ultimately selected on average for each video and about 90% frames are discarded for the video containing the class.

Detector Updating. After accumulating more instances from weakly labeled videos, a large set of positive instances of the new concept is collected, which can help improve the concept detector in the next iteration. The frames selected in the previous iterations will not be considered in later iterations, which makes the model equipped with different instances in every iteration. In order to update the concept detector, these newly mined instances are added into the positive set. The regions from general background images and learned concepts are treated as negatives. We retrain one linear SVM classifier for each new concept and the hard negative mining method is also used. After this process repeats again and again, we can achieve a very mature concept detector with a considerable number of mined instances. For fair comparison, we use the same detection strategies as the previous work [14] in testing phase.

Knowledge Updating. Once enough instances of each new concept (about 10,000 instances) are obtained, the pre-trained CNN can be further improved to generate more informative features by fine-tuning it with these new instances. During fine-tuning, we replace the (N+1)-way output layer of the pre-trained CNN in Section 3.1 with a randomly initialized (M+1)-way classification layer (including M new concepts and one for background). We set $M = 20$ in our experiments. Since these mined instances may contain some noisy data (e.g., inaccurate bounding box of the object), we only use the original set of mined instances during fine-tuning and no additional data augmentation (e.g., ≥ 0.5 IoU overlap) is performed. The negatives for training concept detectors are used as background. The fine-tuning is run for 50K SGD iterations for the 7-lay architecture [18] and 100K iterations for NIN [21], respectively.

Finally, a bounding box regression model is learned to fix many localization errors in the testing as in [14]. From the mined instances, we select $2,000$ detected instances with highest detection scores in the later iterations as ground-truth boxes for training the regression model. The concept detector in the later iterations will be very mature and these top detection boxes have high possibilities to locate the precise object locations. We only consider a region proposal if it has an IoU with ground-truth box greater than 0.8.

4. Experiments

We evaluate our framework on the PASCAL Visual Object Classes (VOC) datasets [9], which are widely used as the benchmark for object detection. PASCAL VOC 2007, VOC 2010 and VOC 2012 are all tested. For each object class, we train the object detector by using two seeds and about $20,000$ weakly labeled videos. Note that our method is independent of any specific test set and only one object detector is used for testing the three datasets. For VOC 2010 and VOC 2012, we evaluate test results on the online evaluation server. We compare our method with the

Table 1. Detection average precision (%) on PASCAL VOC2007 test. Rows 1-4 show the baselines. Rows 5-7 are the results of R-CNN based on the NIN [21]. Rows 8-9 show R-CNN results fine-tuned with 179 extra classes. Rows 10-14 show our results in different iterations, with/without fine-tuning and bounding box regression. "B_initial" and "B_I15" represent the results in the beginning with two seeds and after the 15th iteration, respectively. "B_FT" and "B_FT_I2" are the results after fine-tuning with the mined instances and running 2 more iterations, respectively. "B_FT_I2_no179_BB" represents the results after directly using the classification network, and no fine-tuning with 179 detection classes is performed. Rows 16-19 show the results based on NIN [21].

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM HSC[29]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3
R-CNN[14]	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN BB[14]	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
Ngrams [8]	14.0	36.2	12.5	10.3	9.2	35.0	35.9	8.4	10.0	17.5	6.5	12.9	30.6	27.5	6.0	1.5	18.8	10.3	23.5	16.4	17.2
R-CNN_NIN	72.2	74.5	58.6	47.4	38.7	68.1	75.4	72.1	38.3	69.9	57.2	69.5	67.5	72.4	59.4	34.8	61.5	60.3	67.4	69.9	61.8
R-CNN_NIN BB	72.1	78.2	64.3	49.8	42.2	71.6	77.1	77.8	41.7	72.7	61.3	73.6	77.3	73.6	64.2	37.2	64.9	64.5	70.2	72.8	65.4
R-CNN_NIN 179 BB	74.2	79.3	66.7	**50.1**	39.1	71.1	72.8	76.1	46.3	72.8	64.1	74.9	76.8	74.1	**65.9**	39.1	65.1	**65.1**	**71.1**	**73.7**	65.9
R-CNN 179	62.5	70.2	54.4	42.7	35.4	63.1	71.9	61.5	34.0	61.0	47.1	60.7	64.1	67.9	56.8	32.6	58.2	45.7	59.2	64.5	55.7
R-CNN 179 BB	69.8	73.2	60.2	43.8	38.7	66.2	75.2	65.3	36.1	66.8	56.1	65.0	70.7	70.8	60.6	33.7	64.2	49.1	64.2	65.2	59.7
B_Initial	26.3	11.9	3.2	12.9	9.3	16.0	2.5	6.4	0.9	14.3	4.1	9.7	13.2	21.6	13.7	6.0	15.9	3.5	11.1	31.4	11.7
B_I15	61.1	65.7	51.1	38.8	29.8	57.4	63.8	57.8	26.8	57.1	44.3	57.2	55.7	61.3	45.5	27.2	57.1	38.0	50.4	58.0	50.3
B_FT	65.2	71.6	53.8	39.5	32.2	64.1	70.4	63.0	33.9	60.9	50.2	58.5	64.8	65.9	54.0	27.4	60.6	45.8	59.3	60.7	55.1
B_FT_I2	68.9	70.5	55.6	42.7	37.0	64.1	71.1	66.1	34.5	63.1	51.8	60.9	63.0	67.1	52.8	31.6	62.1	45.8	57.6	64.2	56.5
B_FT_I2_BB	72.2	72.8	61.8	46.7	42.0	66.1	74.2	74.6	37.3	68.3	56.8	65.7	71.3	68.4	58.0	35.1	66.3	47.2	64.0	65.7	60.7
B_FT_I2_no179_BB	72.0	70.2	56.5	40.7	37.2	61.7	60.9	75.8	33.7	66.6	44.4	**76.5**	69.7	**76.9**	58.5	29.3	66.9	49.7	61.1	57.6	58.3
B_NIN_I15	69.0	69.4	52.3	42.4	36.3	65.6	68.9	67.5	33.2	70.7	50.3	68.1	68.8	68.9	40.1	26.3	67.3	57.1	61.5	67.6	57.6
B_NIN_FT	71.1	71.5	59.0	43.7	37.1	68.1	73.1	72.8	39.8	72.1	55.3	68.3	67.6	70.7	54.8	35.4	68.4	58.2	64.9	66.2	60.9
B_NIN_FT_I2	71.0	73.6	61.3	46.3	40.6	70.3	73.8	74.0	43.7	72.9	55.6	68.5	69.2	70.7	57.6	37.7	69.3	59.6	65.3	68.7	62.5
B_NIN_FT_I2_BB	**75.9**	76.8	**66.9**	49.0	**47.9**	**72.1**	**77.2**	**77.9**	**48.6**	**78.5**	**65.0**	73.9	**77.3**	73.6	62.7	**40.4**	**73.5**	64.4	69.2	70.6	**67.1**

Table 2. Detection average precision (%) on PASCAL VOC2010 test.

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Regionlets[38]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM[13]	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN BB[14]	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	53.7
R-CNN 179 BB	73.1	66.2	55.2	38.7	36.5	59.2	60.2	71.8	27.1	51.6	42.0	69.5	63.9	71.0	59.2	29.7	60.0	38.5	61.8	51.2	54.3
B_FT_I2_BB	75.7	68.0	59.2	42.6	40.0	62.4	62.0	72.3	29.5	58.2	40.8	72.0	66.3	72.8	59.9	30.9	62.6	39.9	59.0	55.7	56.5
B_NIN_FT_I2_BB	**77.7**	**73.8**	**62.3**	**48.7**	**45.4**	**67.3**	**67.0**	**80.3**	**41.3**	**70.8**	**49.7**	**79.5**	**74.7**	**78.6**	**64.5**	**36.0**	**69.9**	**55.7**	**70.4**	**61.7**	**63.8**

state-of-the-art baselines, including DPM HSC [29], Regionlets [38], SegDPM [13] and R-CNN [14]. They used all data in the VOC *train* and *val* set for training detectors. We also compare our method with the recent weakly supervised method [8], which discovers instances from web images. We implement two versions of R-CNN (i.e., "R-CNN 179" and "R-CNN 179 BB" with bounding-box regression), which firstly fine-tune the classification CNN with 179 extra classes and then fine-tune the CNN with VOC 20 classes following the settings in [14]. We also report the performances of two version of R-CNN (i.e., "R-CNN_NIN" and "R-CNN_NIN BB") using the Network-in-Network (NIN) [21]. We follow the published code of R-CNN [14] and only replace their original network with NIN architecture detailed in Section 3.1.

4.1. Comparison with the state-of-the-arts

Table 1, 2 and 3 shows the results on the VOC 2007, 2010 and 2012, respectively. All our variants strongly outperform the methods [29] [38] [13] based on hand-crafted features learning and deformable part models. Based on the 7-layer network [18], our method ("B_FT_I2_BB") achieves 60.7% in mAP, which is significantly superior to 58.5% of "R-CNN" [14] and 34.3% of "DPM HSC" [14]. Compared to R-CNN, our method increases the performance by 2.8% and 2.7% on VOC2010 and VOC2012, respectively. When fine-tuning the CNN based on the Network in Network (NIN) [21], our method ("B_NIN_FT_I2_BB") can achieve 67.1% on VOC2007, 63.8% on VOC 2010, and 63.2% on VOC2012, which outperforms the "R-CNN BB [14]" by a large margin of more than 8% on all three test sets and significantly outperforms the "R-CNN_NIN BB" by 1.7% on VOC2007. The bounding box regression can further fix a large number of mislocalized detections. Note that our method only uses two positive instances and trains one single detector for all three datasets, while the baselines use different large training sets and carefully tune the model parameters for different test sets. This superiority well verifies the effectiveness and generality of our framework that automatically learns a significantly better detector than the fully supervised methods. The recent weakly supervised method [8] only obtained 17.2% in mAP on VOC 2007, which is much worse than the proposed method.

4.2. Discussions on Different Components

Different network architectures. "R-CNN_NIN BB" can significantly increase the mAP on VOC 2007 achieved by [14] from 58.5% to 65.4% and mAP on VOC 2012 of [14] from 53.3% to 62.4%, respectively. Its superiority largely benefits from the better neural network architec-

Table 3. Detection average precision (%) on PASCAL VOC2012 test.

VOC 2012 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
SDS[16]	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7	50.7
R-CNN BB[14]	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1	53.3
R-CNN_NIN BB	77.9	73.1	62.6	39.5	**43.3**	**69.1**	66.4	78.9	39.1	68.1	**50.0**	77.2	71.3	76.1	64.7	38.4	66.9	**56.2**	66.9	**62.7**	62.4
R-CNN_NIN 179 BB	77.6	73.3	**65.7**	39.3	42.8	68.8	65.9	80.1	41.2	69.4	49.7	78.5	72.5	76.2	**64.9**	**39.8**	66.7	54.5	65.5	61.8	62.7
R-CNN 179 BB	72.7	66.0	55.0	34.0	32.0	59.5	60.5	70.9	29.2	51.5	40.2	70.0	62.3	68.4	58.9	30.8	58.2	37.7	58.3	53.9	53.5
B_FT_I2_BB	75.8	68.2	58.2	39.6	37.0	63.2	62.2	72.3	29.3	59.0	40.8	71.4	66.2	71.3	59.4	30.9	61.2	41.1	57.3	56.5	56.0
B_NIN_FT_I2_BB	**78.0**	**74.2**	61.3	**45.7**	42.7	68.2	**66.8**	**80.2**	**40.6**	**70.0**	49.8	**79.0**	**74.5**	**77.9**	64.0	35.3	**67.9**	55.7	**68.7**	62.6	**63.2**

Table 4. Detection average precision (%) on PASCAL VOC2007 test by the version initialized with more training data.

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN BB[14]	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN_NIN BB	72.1	**78.2**	64.3	49.8	42.2	71.6	77.1	77.8	41.7	72.7	61.3	73.6	77.3	73.6	64.2	37.2	64.9	64.5	**70.2**	72.8	65.4
B_VOC_I2	69.2	70.3	58.5	42.7	38.3	64.6	71.7	67.3	37.2	64.7	52.1	62.6	64.6	69.1	54.1	33.3	62.7	46.6	58.8	66.8	57.8
B_VOC_I2_BB	72.3	72.7	64.0	46.7	43.5	67.5	74.8	74.5	39.4	70.1	57.7	68.3	71.5	70.2	58.5	36.9	68.1	49.1	65.5	67.9	62.0
B_NIN_VOC_I2	73.6	74.8	65.5	48.4	46.4	71.2	74.6	76.0	46.1	75.3	56.8	72.6	73.6	73.8	60.1	40.5	71.6	68.3	67.4	73.1	65.5
B_NIN_VOC_I2_BB	**76.2**	77.8	**69.9**	**51.0**	**52.0**	**73.6**	**77.5**	**78.0**	**49.4**	**82.1**	**65.2**	**76.8**	**79.1**	**74.9**	**64.5**	**41.6**	**74.7**	**70.8**	69.8	**73.6**	**68.9**

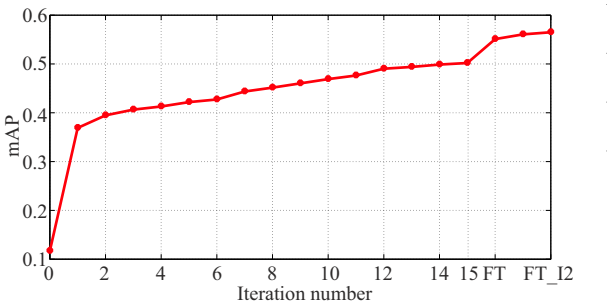

Figure 4. Performances in different iterations of our framework on VOC 2007. We run 15 iterations for mining more instances. Then we fine-tune the CNN with mined instances ("FT") and 2 iterations are further performed to improve the detectors ("FT_I2").

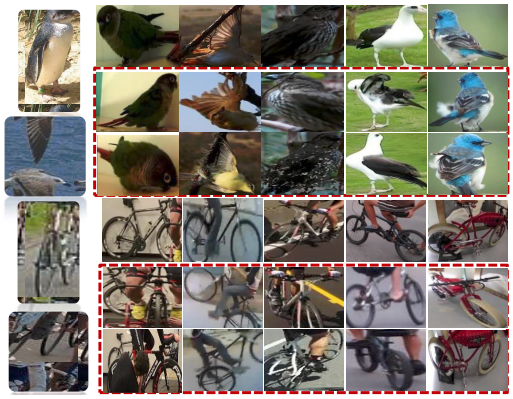

Figure 5. Visualization of our tracking results for bird and bicycle. For each class, the left column shows the initialized two seeds. The top row shows the detected seeds in each video and two tracked instances are presented within the dashed box.

ture. And the large improvement of "B_NIN_FT_I2_BB" over "B_FT_I2_BB" also demonstrates that more informative CNNs can lead to better initialization and learning capabilities during the repetition of the process.

Usage of learned concepts. The "R-CNN 179 BB" only performs slightly better than "R-CNN BB" (e.g., smaller than 0.6% increase on VOC 2010 and VOC 2012). The main reason is that the samples from 179 extra classes provide limited additional information when enough instances of 20 classes are already used in [14]. Similar slight improvement can be observed when comparing "R-CNN_NIN 179 BB" and "R-CNN_NIN BB". However, when only two instances of a new concept are given, our method can benefit from these instances for domain-specific fine-tuning. After fine-tuning the classification network with 179 extra classes, the performance of our method ("B_FT_I2_BB") can increase by 2.4% over "B_FT_I2_no179_BB".

Detector updating by learning with video contexts. Figure 4 shows our performances in different iterations for updating object detectors by our framework. We show the results based on the 7-layer network on VOC 2007 and the corresponding AP for each class is presented in Table 1. In the beginning, we only obtain 11.7% in mAP with only two seeds. After the first round of learning with video context, we can substantially improve the mAP to 36.1%, which is even higher than mAP of DPM HSC [29]. Most of the easy test samples can be detected by our updated model. After 15 iterations are performed, we can achieve 50.3% in mAP ("B_I15") and collect about 10,000 instances for each class.

Knowledge updating by learning with video contexts. Figure 4 also reports the performances of fine-tuning the pre-trained CNN when more diverse instances are mined. After further fine-tuning the pretrained CNN with these mined instances, 4.8% improvement is achieved by comparing "B_FT" with "B_I15"), as reported in Table 1. Using NIN as the CNN architecture, we achieve 57.6% after 15 iterations ("B_NIN_I15") and obtain 60.9% after fine-tuning ("B_NIN_FT"). It proves that more informative features can be learned by fine-tuning. We then further improve the detectors by running 2 more iterations based the fine-tuned CNNs and better detection performances can be achieved, shown by "B_FT_I2", "B_NIN_FT_I2". Limited by the computational cost, our experiments only report the current results at two iterations after fine-tuning CNN. With

Table 5. Detection average precision (%) on PASCAL VOC2012 test by the version initialized with more training data.

VOC 2012 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
SDS[16]	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7	50.7
R-CNN BB[14]	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1	53.3
R-CNN_NIN BB	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	**38.4**	66.9	56.2	66.9	62.7	62.4
B_NIN_VOC_I2	77.6	71.7	60.9	41.3	38.2	65.5	64.5	80.0	38.1	69.9	47.1	79.3	74.7	76.1	61.9	33.5	67.7	54.8	62.6	63.2	61.4
B_NIN_VOC_I2_BB	**80.2**	**75.0**	**64.9**	**45.8**	**44.0**	**70.1**	**67.6**	**81.4**	**40.8**	**71.4**	**51.9**	**81.0**	**75.6**	**78.2**	**66.1**	37.6	**68.5**	**59.4**	**68.0**	**65.2**	**64.6**

Table 6. Detection average precision (%) on aeroplane class of PASCAL VOC2007 by different seed selections. We run our version "B_FT_I2" with the same setting by using different numbers of randomized seed instances. We report the results based on different numbers of seeds, i.e., one, two and five, as well as different seed instances randomly ten times for each number of seed.

seed number	1	2	3	4	5	6	7	8	9	10	mean
1	65.1	66.1	68.1	67.3	67.2	68.2	65.4	66.1	67.8	66.7	66.8
2	67.0	66.8	70.4	69.7	68.3	69.2	66.9	67.9	69.2	69.8	68.5
5	**69.7**	**69.3**	**71.2**	**72.3**	**70.1**	**70.8**	**71.5**	**70.9**	**70.2**	**71.3**	**70.7**

better CNN architectures (e.g., googleLeNet [33]), it is predictable that our detectors can be further improved.

Different seed selections. We extensively evaluate how our framework performs with different seed selections. Due to the computational limitation, we only test on one specific object class, i.e., aeroplane, as reported in Table 6. We test three numbers of seeds during the initialization. For each number, we generate different seeds randomly ten times to evaluate the robustness on seed selections. It can be seen that our method can archive better performance with more initial seeds. With different randomized seeds of each number, we obtain slightly different results and their mean 68.5% is only slightly worse than 68.9% by our version with two selected instances in Table 1. The CNN fine-tuning only with the aeroplane class may lead to this slightly decrease. The main reason for the robustness may be the usage of the large number of videos. By mining various instances with different views or appearance changes, we can easily introduce greater data diversity into the model training.

4.3. Initialization with More Training Data

We evaluate the state-of-the-arts (e.g., R-CNN) trained with all training sets can be further improved by using our framework. The results of "B_VOC_I2" and "B_VOC_I2_BB" are shown in Table 4, in which all the images in VOC 2007 are used. The detectors are trained over the deep features from the 7-layer CNN and two more iterations are performed to mine more instances from videos. We obtain 62.0% mAP on VOC 2007, 3.5% higher than the original "R-CNN BB" (58.5% in mAP). Based on the Network-in-Network, we also achieve superior performances over "R-CNN_NIN BB" (68.9% vs 65.4% in mAP on VOC 2007 and 64.6% vs 62.4% in mAP on VOC 2012 in Table 4 and Table 5, respectively).

4.4. Visualizations

We show the two selected instances and some mined instances for four classes in Figure 5. We randomly select some mined instances from all iterations. It can be observed that our method successfully tracks more instances with different view-angles, occlusions or appearance variance. Many qualitative detection results on the VOC 2007 test set are presented in Figure 6, which are obtained from our best model "B_NIN_FT_I2_BB", each image is selected due to it is impressive and accurate.

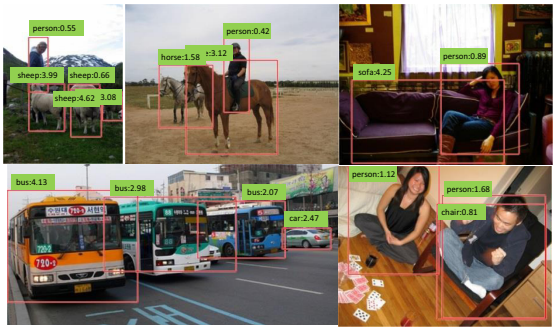

Figure 6. Some exemplar detection results. All detections with precision greater than 0.5 are shown. Each detection is labeled with the predicted class and the detection score from the detector.

5. Conclusion and Future Work

In this paper, inspired by the intuitive observation of the baby learning process, we presented a novel computational weakly supervised learning framework for object detection by combining prior knowledge modeling, exemplar learning, and learning with video contexts. Significant improvements over fully-training based methods were achieved by our framework on PASCAL VOC 07/10/12 with only two positive instances along with about 20,000 weakly labeled real-world videos. In the future, we will explore how to adequately utilize more contextual information (e.g. scene, human actions, other objects) to mine more accurate and diverse instances. Our framework can also be easily extended to improve various vision tasks, such as face age recognition, people identification and scene classification.

Acknowledgement

This work was supported in part by the Guangdong Natural Science Foundation under Grant S2013050014548 and Grant 2014A030313201, in part by the Program of Guangzhou Zhujiang Star of Science and Technology under Grant 2013J2200067, and in part by Fundamental Research Funds for the Central Universities (no. 13lgjc26).

References

[1] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *CVPR*, 2005. 3

[2] C.-Y. Chen and K. Grauman. Watching unlabeled video helps learn new human actions from very few labeled snapshots. In *CVPR*, pages 572–579, 2013. 1

[3] D.-J. Chen, H.-T. Chen, and L.-W. Chang. Video object cosegmentation. In *ACM Multimedia*, 2012. 3

[4] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013. 1, 2

[5] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 1

[6] J. Choi, M. Rastegari, A. Farhadi, and L. S. Davis. Adding unlabeled samples to categories by learned attributes. In *CVPR*, pages 875–882, 2013. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[8] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 1, 2, 6

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1, 5

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 35(8):1915–1929, 2013. 2

[11] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006. 3

[12] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *NIPS*, 2009. 3

[13] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 2, 6

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 2, 3, 5, 6, 7, 8

[15] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 3

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*. 2014. 7, 8

[17] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 3

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2, 3, 5, 6

[19] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 1346–1353, 2012. 3

[20] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. lin, and S. Yan. Deep human parsing with active template regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99), 2015. 1

[21] M. Lin, Q. Cheng, and S. Yan. Network in network. In *ICLR*, 2014. 3, 5, 6

[22] H. Liu and S. Yan. Robust graph mode seeking by graph shift. *ICML*, pages 671–678, 2010. 5

[23] S. Liu, X. Liang, L. Liu, K. Lu, L. Lin, and S. Yan. Fashion parsing with video context. In *ACM Multimedia*, pages 467–476, 2014. 3

[24] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-cnn meets knn: Quasi-parametric human parsing. *CVPR*, 2015. 2

[25] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1, 4

[26] I. Misra, A. Shrivastava, and M. Hebert. Watch and learn: Semi-supervised learning of object detectors from videos. *CVPR*, 2015. 3

[27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 4

[28] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 3

[29] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013. 2, 6, 7

[30] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Computer Vision and Pattern Recognition*, pages 29–36, 2005. 1

[31] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, pages 369–383, 2012. 2

[32] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 3

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 1, 2, 8

[34] K. D. Tang, R. Sukthankar, J. Yagnik, and F.-F. Li. Discriminative segment annotation in weakly labeled video. In *CVPR*, pages 2483–2490, 2013. 3

[35] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, pages 1879–1886, 2011. 3, 4

[36] A. Vezhnevets and V. Ferrari. Associative embeddings for large-scale knowledge transfer with self-assessment. *arXiv preprint arXiv:1312.3240*, 2013. 3

[37] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014. 3

[38] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, pages 17–24, 2013. 2, 6

[39] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010. 4

[40] Y. Yang, G. Shu, and M. Shah. Semi-supervised learning of feature hierarchies for object detection in a video. In *CVPR*, 2013. 3

[41] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 2015. 1