Temporal Dynamic Graph LSTM for Action-driven Video Object Detection

Yuan Yuan[1] Xiaodan Liang[2] Xiaolong Wang[2] Dit-Yan Yeung[1] Abhinav Gupta[2]
[1]The Hong Kong University of Science and Technology [2] Carneige Mellon University
yyuanad@ust.hk, xiaodan1@cs.cmu.edu, xiaolonw@cs.cmu.edu, dyyeung@cse.ust.hk, abhinavg@cs.cmu.edu

Abstract

In this paper, we investigate a weakly-supervised object detection framework. Most existing frameworks focus on using static images to learn object detectors. However, these detectors often fail to generalize to videos because of the existing domain shift. Therefore, we investigate learning these detectors directly from boring videos of daily activities. Instead of using bounding boxes, we explore the use of action descriptions as supervision since they are relatively easy to gather. A common issue, however, is that objects of interest that are not involved in human actions are often absent in global action descriptions known as "missing label". To tackle this problem, we propose a novel temporal dynamic graph Long Short-Term Memory network (TD-Graph LSTM). TD-Graph LSTM enables global temporal reasoning by constructing a dynamic graph that is based on temporal correlations of object proposals and spans the entire video. The missing label issue for each individual frame can thus be significantly alleviated by transferring knowledge across correlated objects proposals in the whole video. Extensive evaluations on a large-scale daily-life action dataset (i.e., Charades) demonstrates the superiority of our proposed method. We also release object bounding-box annotations for more than 5,000 frames in Charades. We believe this annotated data can also benefit other research on video-based object recognition in the future.

1. Introduction

With the recent success of data-driven approaches in recognition, there has been a growing interest in scaling up object detection systems [38]. However, unlike classification, exhaustively annotating object instances with diverse classes and bounding boxes is hardly scalable. Therefore, there has been a surge in exploring in unsupervised and weakly-supervised approaches for object detection. However, fully unsupervised approaches [30, 17] without any annotations currently give considerably inferior performance on similar tasks, while conventional weakly-supervised methods [2, 16, 42] use static images to learn the detectors. These object detectors, however, fail to generalize to videos due to shift in domain. One alternative is to use these weakly-supervised approaches but using video frames themselves. However, current approaches rely heavily on the accuracy of image-level labels and are vulnerable to missing labels (as shown in Figure 1). Can we design a learning framework that is robust to these missing labels ?

In this paper, we explore a novel slightly-supervised video object detection pipeline that uses human action labels as supervision for object detection. As illustrated in Figure 1, the coarse human action labels spanning multiple frames (e.g., *watching a laptop* or *sitting in a chair*) help indicate the presence of participating object instances (e.g., *laptop* and *chair*). Compared to prior works, our investigated setting has two major merits: 1) the textual action descriptions for videos are much cheaper to collect, e.g., through text tags, search queries and action recognition datasets [32, 10, 36]; and 2) the intrinsic temporal coherence in video domain provides more cues to facilitate the recognition of each object instance and help overcome the missing label problem.

Action-driven supervision for object detection is much more challenging since it can only access object labels for some specific frames, while a considerable number of uninvolved object labels are unknown. As shown in the right column of Figure 1, four action categories are labeled for different periods in the given video. In each period, the action label (e.g., *tidying a shelf*) only points out the *shelf* category and misses the rest of the categories such as *laptop*, *table*, *chair* and *refrigerator*. On the other hand, the missed categories (e.g., *laptop*) may appear in other labeled actions in the same video. Inspired by this observation, we propose to alleviate the missing label issue by exploiting the rich temporal correlations of object instances in the video. The core idea is that action labels in a different period may help to infer the presence of some objects in this current period. Specifically, a novel temporal dynamic graph LSTM (TD-Graph LSTM) framework is introduced to model the complex and dynamic temporal graph structure for object proposals in the whole video and thus enable the joint reasoning for all frames. The knowledge of all action labels in

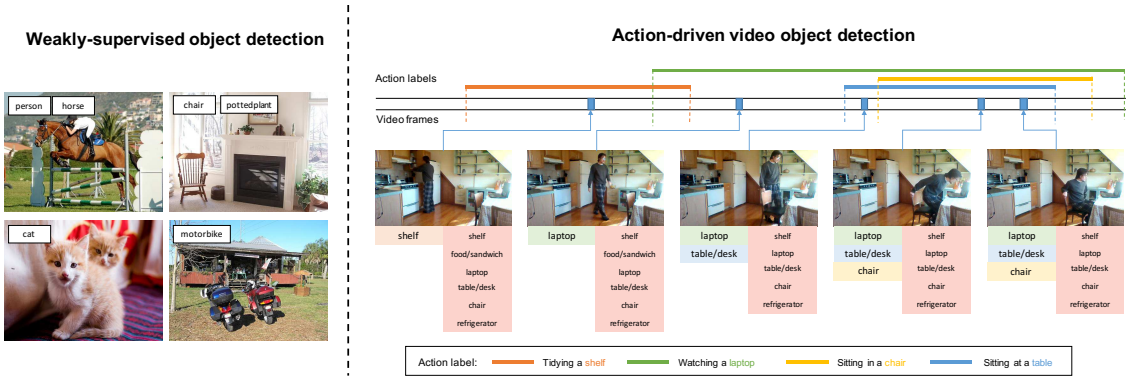

Figure 1. **(Left)** shows the traditional weakly-supervised object detection setting. Each training image has an accurate image-level annotation about object categories. **(Right)** shows our action-driven weakly-supervised video detection setting. Video-level action labels are provided for each video, indicating what and when (the start and end) the action happened in the video. For each frame, the object categories in its left-below are the participating objects in the action label, while those in its right-below are all objects appearing in the frame.

the video can thus be effectively transfered into all frames to enhance their frame-level categorizations.

To incorporate the temporal correlation of object proposals for global reasoning, we resort to the family of recurrent neural networks [11] due to their good sequential modeling capability. However, existing recurrent networks are largely limited in the constrained information propagation on fixed nodes following predefined routes such as tree-LSTM [39], graph-LSTM [20] and structural-RNN [12, 18]. In contrast, due to the unknown object localizations and temporal motion, it is difficult to find an optimal structure that connects object proposals for routed information propagation to achieve weakly-supervised video object detection. The proposed TD-Graph LSTM, posed as a general dynamic recurrent structure, overcomes these limitations by performing the dynamic information propagation based on an adaptive temporal graph that varies over both time periods in the video and model status in each updating step.

Specifically, the dynamic temporal graph is constructed based on the visual correlation of object proposals across neighboring frames. The set of graph nodes denotes the entire collection of object proposals in all the frames, while graph edges are adaptively specified for consecutive frames in distinct learning steps. At each iteration, given the updated feature representation of object proposals, we only activate the edge connections with object proposals that have highest similarities with each current proposal. The adaptive graph topology can thus be constructed where different proposals are connected with different temporal correlated neighbors. TD-Graph LSTM alternatively performs the information propagation through each temporal graph topology and updates the graph topology at each iteration. In this way, our model enables the joint optimization of feature learning and temporal inference towards a robust slightly-supervised detection framework.

The contributions of this paper are summarized as 1) We explore a new slightly-supervised video object detection pipeline that leverages convenient action descriptions as the supervision; 2) A novel TD-Graph LSTM framework alleviates the missing label issue by enabling global reasoning over the whole video; 3) TD-Graph LSTM is posed as a general dynamic recurrent structure that performs temporal information propagation on an adaptively updated graph topology at each iteration; 4) We collect and release 5,000 frame annotations with object-level bounding boxes on daily-life videos, with the goal of evaluating our model and also helping advance the object detection community.

2. Related Works

Weakly-Supervised Object Detection. Though recent state-of-the-art fully-supervised detection pipelines [9, 28, 8, 27, 23] have achieved great progress, they heavily rely on large-scale bounding-box annotations. To alleviate this expensive annotation labor, weakly-supervised methods [6, 34, 1, 35, 41, 4, 13, 46] have recently attracted a lot of interest. These approaches use cheaper image-level object labels rather than bounding boxes. Beyond the image domain, another line of research [43, 19, 33, 26, 25, 14, 17, 45] attempts to exploit the temporal information embedded in videos to facilitate the weakly-supervised object detection. Different from all the existing pipelines, we investigate a much cheaper action-driven object detection setting that aims to detect all object instances given only action descriptions. In addition, instead of employing multiple separate steps (e.g., detection and tracking) [15, 17, 43, 19, 33] to capture motion patterns, our TD-graph LSTM is an end-to-end framework that incorporates the intrinsic temporal coherence with a designed dynamic recurrent network structure into the action-driven slightly-supervised detection.

Sequential Modeling. Recurrent neural networks, espe-

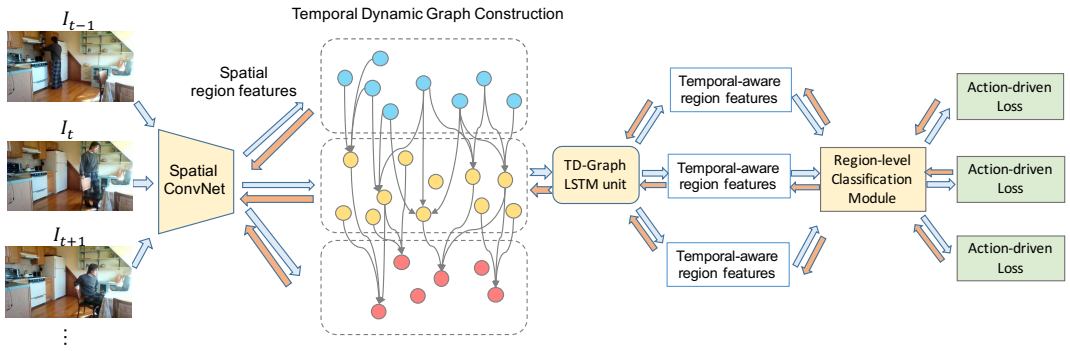

Figure 2. Our TD-Graph LSTM. Each frame is first passed into a spatial ConvNet to extract region-level features. A temporal graph structure is then constructed by dynamic edge connections between regions in two consecutive frames. TD-Graph LSTM then recurrently propagates information over the updated graph to generate temporal-aware feature representations for all regions. A region-level classification module is then adopted to produce category confidences of all regions in each frame, which are aggregated to obtain frame-level action predictions. The final action-driven loss for each frame is used to feedback signals into the whole model. After each gradient updating, the temporal graph is dynamically updated based on new visual features. For clarity, some edges in the graph are omitted.

cially Long Short-Term Memory (LSTM) [11], have been adopted to address many video processing tasks such as action recognition [24], action detection [44], video prediction [37, 31], and video summarization [47]. However, limited by the fixed propagation route of existing LSTM structures [11], most of the previous works [24, 44, 37] can only learn the temporal interdependency between the holistic frames rather than more fine-grained object-level motion patterns. Some recent approaches develop more complicated recurrent network structures. For instance, structural-RNN [12] develops a scalable method for casting an arbitrary spatio-temporal graph as a rich RNN mixture. A more recent Graph LSTM [21] defined over a pre-defined graph topology enables the inference for more complex structured data. However, both of them require a pre-fixed network structure for information propagation, which is impractical for weakly-supervised/slightly-supervised object detection without the knowledge of object localizations and precise object class labels. To handle the propagation over dynamically specified graph structures, we thus propose a new temporal dynamic network structure that supports the inference over the constantly changing graph topologies in different training steps.

3. The proposed TD-Graph LSTM

Overview. We establish a fully-differentiable temporal dynamic graph LSTM (TD-Graph LSTM) framework for the action-driven video object detection task. For each video, the provided annotations are a set of action labels $\mathbf{Y} = \{y_1, \ldots, y_N\}$, each of which describes the action $y_i =< a_i, c_i >$ appearing within a consecutive sequence of frames $\{I_{d_i^s}, \ldots, I_{d_i^e}\}$, where d_i^s and d_i^e indicate the action starting and ending frame index. a_i denotes the corresponding action noun while c_i denotes the object noun. For example, the action *tidying a shelf* is comprised of the action *Tidying* and object *a shelf*. To achieve weakly-supervised object detection, we only extract the object nouns $\{c_i\}$ of action labels in all videos and eliminate the prepositions (e.g., *a*, *the*) to produce an object category corpus (e.g., *shelf*, *door*, *cup*) with C classes. Each frame I can be thus assigned with several participating object classes. For example, frames with two actions will be assigned with more than one participating object class, as shown in Figure 1. The action-driven object detection is thus posed as a multi-class weakly-supervised video object detection problem. For simplicity, we eliminate the subscript i of action labels in the following.

Figure 2 gives an overview of our TD-Graph LSTM. Each frame in the input video is first passed through a spatial ConvNet to obtain spatial visual features for region proposals. Based on visual features, similar regions in two consecutive frames are discovered and associated to indicate the same object across the temporal domain. A temporal graph structure is constructed by connecting all of the semantically similar regions in two consecutive frames, where graph nodes are represented by region proposals. The TD-Graph LSTM unit is then employed to recurrently propagate information over the whole temporal graph, where LSTM units take the spatial visual features as the input states. Benefiting from the graph topology, TD-Graph LSTM is capable of incorporating temporal motion patterns for participating objects in the action in a more efficient and meaningful way. TD-Graph LSTM outputs the enhanced temporal-aware features of all regions. Region-level classification is then employed to produce classification confidences. These region-level predictions can finally be aggregated to generate frame-level object class prediction, supervised by the object classes from action labels. The action-driven object categorization loss thus enables the holistic back-propagation into all regions in the video,

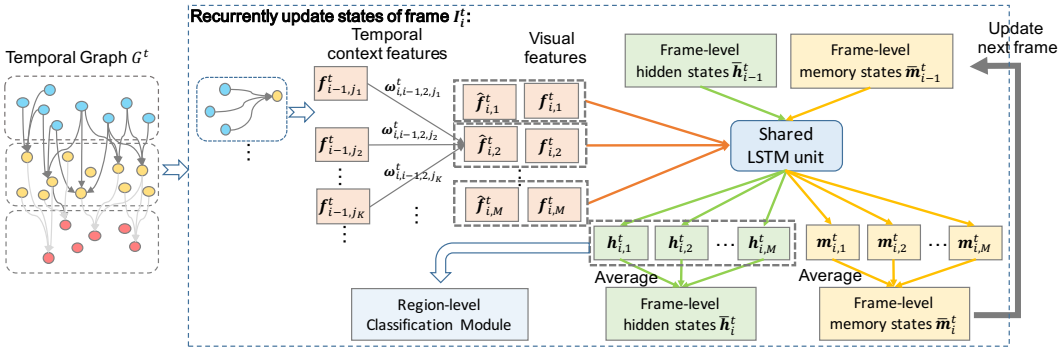

Figure 3. Illustration of the TD-Graph LSTM layer at t-th gradient updating. Given the constructed temporal graph $\mathcal{G}^t$, the TD-Graph LSTM recurrently updates the hidden states of each frame $I_i, i \in \{1, \ldots, N\}$ as the enhanced temporal-aware visual feature, and then feeds these features into a region-level classification module to compute final category confidences of all regions. Specially, each LSTM unit takes the shared frame-level hidden states $\bar{\mathbf{h}}_{i-1}^t$ and memory states $\bar{\mathbf{m}}_{i-1}^t$, and input features for all regions as the inputs. Then the updated hidden states and memory states for all regions are produced, which are then averaged to generate the new frame-level hidden states $\bar{\mathbf{h}}_i^t$ and memory states $\bar{\mathbf{m}}_i^t$ for updating next frame I_{i+1}. The input features of each region consist of the visual features $\mathbf{f}_{i,j}^t$ and temporal context features $\hat{\mathbf{f}}_{i,j}^t$ that are aggregated by its connected regions with edge weights in the preceding frame.

where the prediction of each frame can mutually benefit from each other.

3.1. TD-Graph LSTM Optimization

The proposed TD-Graph LSTM is comprised by three parametrized modules: spatial ConvNet $\Phi(\cdot)$ for visual feature extraction, TD-Graph LSTM unit $\Psi(\cdot)$ for recurrent temporal information propagation, and region-level classification module $\varphi(\cdot)$. These three modules are iteratively updated, targeted at the action-driven object detection.

At each model updating step t, a temporal graph structure $\mathcal{G}^t =< \mathbf{V}, \mathcal{E}^t >$ for each video is constructed based on the updated spatial visual features $\mathbf{f}^t$ of all regions $\mathbf{r}$ in the videos, defined as $\mathcal{G}^t = \beta(\Phi^t(\mathbf{r}))$. $\beta(\cdot)$ is a function to calculate the dynamic edge connections $\mathcal{E}^t$ conditioning on the updated visual features $\mathbf{f}^t = \Phi^t(\mathbf{r})$. The TD-Graph LSTM unit Ψ^t recurrently functions on the visual features $\mathbf{f}^t$ of all frames and propagates temporal information over the graph $\mathcal{G}^t$ to obtain the enhanced temporal-aware features $\hat{\mathbf{f}}^t = \Psi^t(\mathbf{f}^t|\mathcal{G}^t)$ of all regions in the video. Based on the enhanced $\hat{\mathbf{f}}^t$, the region-level classification module φ produces classification confidences $\mathbf{rc}^t$ for all regions, as $\mathbf{rc}^t = \varphi(\hat{\mathbf{f}}^t)$. These region-level category confidences $\mathbf{rc}^t$ can be aggregated to produce frame-level category confidences $\mathbf{pc}^t = \gamma(\mathbf{rc}^t)$ of all frames by summing the category confidences of all regions of each frame.

During training, we define the action-driven loss for each frame as a hinge loss function and train a multi-label image classification objective for all frames in the videos:

$$\begin{aligned}\mathcal{L}(\Phi, \Psi, \varphi) &= \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}\max(0, 1 - \mathbf{y}_{c,i}\mathbf{pc}_{c,i}) \\ &= \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}\max(0, 1 - \mathbf{y}_{c,i}\gamma(\varphi(\Psi(\mathbf{f}_i|\mathcal{G})))),\end{aligned} \tag{1}$$

where C is the number of classes and $\mathbf{y}_{c,i}, i \in \{1, \ldots, N\}$ represents action-driven object labels for each frame. For each frame I_i, $y_{c,i} = 1$ only if the action-driven object label c is assigned to the frame I_i, otherwise as -1. The objective function defined in Eq. 1 can be optimized by the Stochastic Gradient Descent (SGD) back-propagation. At each t-th gradient updating, the temporal graph structure $\mathcal{G}^t$ is accordingly updated by $\beta(\Phi^t(\mathbf{r}))$ for each video. Thus, the TD-Graph LSTM unit optimizes over a dynamically updated graph structure $\mathcal{G}^t$. In the following sections, we introduce the above-defined parametrized modules.

3.2. Spatial ConvNet

Given each frame I_i, we first extract category-agnostic region proposals and then extract their visual features by passing them into a spatial ConvNet $\Phi(\cdot)$ following [8]. To provide a fair comparison on action-driven object detection, we adopt the EdgeBoxes [40] proposal generation method which does not require any object annotations for pretraining. We select the top $M = 500$ proposals $\mathbf{r}_i = \{r_{i,1}, r_{i,2}, ..., r_{i,M}\}$ for the frame I_i with the highest objectness scores, considering the computation efficiency. At the t-th updating step, visual features $\mathbf{f}_i^t = \{f_{i,1}^t, f_{i,2}^t, ..., f_{i,M}^t\} \in \mathbb{R}^{M \times D}$ of all regions $\mathbf{r}_i$ are extracted using the updated spatial ConvNet model, i.e., $\mathbf{f}_i^t = \Phi^t(\mathbf{r}_i)$. The spatial ConvNet $\Phi(\cdot)$ consists of several convolutional layers from the base net and one ROI-pooling layer [8], and two fully-connected layers.

3.3. TD-Graph LSTM Unit

Dynamic Graph Updating. Given the updated visual features $\mathbf{f}_i^t$ of each frame I_i, the temporal graph structure $\mathcal{G}^t =< \mathbf{V}, \mathcal{E}^t >$ can be accordingly constructed by learning the dynamic edge connections $\mathcal{E}^t$. The graph node

$\mathbf{V} = \{v_{i,j}\}, j = \{1, \ldots, M\}$ is represented by visual features $\{\mathbf{f}_{i,j}^t\}$ of all regions in all frames; that is, $M \times N$ nodes for M region proposals of N frames. Each node $v_{i,j}$ is connected with nodes in the preceding frame I_{i-1} and the nodes in subsequent frame I_{i+1}. To incorporate the motion dependency in consecutive frames, the edge connections $\mathcal{E}_{i,i-1}^t$ between nodes in I_i and I_{i-1} are mined by considering their appearance similarities in visual features. Specifically, the edge weight between each pair of nodes $(v_{i,j}, v_{i-1,j'})$ is first calculated as $\frac{1}{2}\exp(-||\mathbf{f}_{i,j}^t - \mathbf{f}_{i-1,j'}^t||_2)$. To make the model inference efficient and alleviate the missing issue, each node $v_{i,j}$ is only connected to K nodes $v_{i-1,j'}$ with the top-K highest edge weights in preceding frame I_{i-1}, and these activated edge weights are normalized to be summed as 1. We denote the normalized edge weight as $\omega_{i,i-1,j,j'}^t$. Thus, the updated temporal graph structure $\mathcal{G}^t$ can be regarded as an undirected K-neighbor graph where each node $v_{i,j}$ is connected with at most K nodes in previous frames.

TD-Graph LSTM. TD-Graph LSTM layer propagates temporal context over graph and recurrently updates the hidden states $\{\mathbf{h}_{i,j}^t\}$ of all regions in each frame I_i to construct enhanced temporal-aware feature representations. These features are fed into the region-level classification module to compute the category-level confidences of each region. TD-Graph LSTM updates hidden state of frame i by incorporating information from frame-level hidden state $\bar{\mathbf{h}}_{i-1}^t$ and memory state $\bar{\mathbf{m}}_{i-1}^t$. The usage of the shared frame-level hidden state and memory state enables the provision of a compact memorization of temporal patterns in the previous frame and is more suitable for massive and possibly missing graph nodes (e.g., 500 in our setting) in a large temporal graph. After performing N updating steps for all frames, our model effectively embeds the rich temporal dependency to obtain the enhanced temporal-aware feature representations of all regions in all frames. For updating the features of each node $v_{i,j}$ in the frame I_i, the TD-Graph LSTM unit takes as the input its own visual features $\mathbf{f}_{i,j}^t$, temporal context features $\hat{\mathbf{f}}_{i,j}^t$, frame-level hidden states $\bar{\mathbf{h}}_{i-1}^t$ and memory states $\bar{\mathbf{m}}_{i-1}^t$, and outputs the new hidden states $\mathbf{h}_{i,j}^t$. Given the dynamic edge connections $e_{i,j} = \{< v_{i,j}, v_{i-1,j'} >\}, j' \in \mathcal{N}_{\mathcal{G}}(v_{i,j})$, the temporal context features $\hat{\mathbf{f}}_{i,j}^t$ can be calculated by performing a weighted summation of features of connected regions:

$$\hat{\mathbf{f}}_{i,j}^t = \sum_{j' \in \mathcal{N}_{\mathcal{G}}(v_{i,j})} \omega_{i,i-1,j,j'}^t \mathbf{f}_{i-1,j'}^t. \tag{2}$$

And the shared frame-level hidden states $\bar{\mathbf{h}}_{i-1}^t$ and memory states $\bar{\mathbf{m}}_{i-1}^t$ can be computed as

$$\bar{\mathbf{h}}_{i-1}^t = \frac{1}{M}\sum_{j=1}^{M} \mathbf{h}_{i-1,j}^t, \quad \bar{\mathbf{m}}_{i-1}^t = \frac{1}{M}\sum_{j=1}^{M} \mathbf{m}_{i-1,j}^t. \tag{3}$$

The TD-Graph LSTM unit consists of four gates for each node $v_{i,j}$: the input gate $\mathbf{gu}_{i,j}^t$, the forget gate $\mathbf{gf}_{i,j}^t$, the memory gate $\mathbf{gc}_{i,j}^t$, and the output gate $\mathbf{go}_{i,j}^t$. The $W_t^u, W_t^f, W_t^c, W_t^o$ are the recurrent gate weight matrices specified for input visual features and $W_t^{ut}, W_t^{ft}, W_t^{ct}, W_t^{ot}$ are those for temporal context features. $U_t^u, U_t^f, U_t^c, U_t^o$ are the weight parameters specified for frame-level hidden states. The new hidden states and memory states in the graph $\mathcal{G}^t$ can be calculated as follows:

$$\begin{aligned}
\mathbf{gu}_{i,j}^t &= \delta(W_t^u \mathbf{f}_{i,j}^t + W_t^{ut} \hat{\mathbf{f}}_{i,j}^t + U_t^u \bar{\mathbf{h}}_{i-1}^t + b_t^u), \\
\mathbf{gf}_{i,j}^t &= \delta(W_t^f \mathbf{f}_{i,j}^t + W_t^{ft} \hat{\mathbf{f}}_{i,j}^t + U_t^f \bar{\mathbf{h}}_{i-1}^t + b_t^f), \\
\mathbf{go}_{i,j}^t &= \delta(W_t^o \mathbf{f}_{i,j}^t + W_t^{ot} \hat{\mathbf{f}}_{i,j}^t + U_t^o \bar{\mathbf{h}}_{i-1}^t + b_t^o), \\
\mathbf{gc}_{i,j}^t &= \tanh(W_t^c \mathbf{f}_{i,j}^t + W_t^{ct} \hat{\mathbf{f}}_{i,j}^t + U_t^c \bar{\mathbf{h}}_{i-1}^t + b_t^c), \\
\mathbf{m}_{i,j}^t &= \mathbf{gf}_{i,j}^t \odot \bar{\mathbf{m}}_{i-1}^t + \mathbf{gu}_{i,j}^t \odot \mathbf{gc}_{i,j}^t, \\
\mathbf{h}_{i,j}^t &= \mathbf{go}_{i,j}^t \odot \tanh(\mathbf{m}_{i,j}^t).
\end{aligned} \tag{4}$$

Here δ is a logistic sigmoid function, and $\odot$ indicates a point-wise product. Given the updated hidden states $\{\mathbf{h}_{i,j}^t\}$ and memory states $\{\mathbf{m}_{i,j}^t\}$ of all regions in frame I_i, we can obtain new frame-level hidden states $\bar{\mathbf{h}}_i^t$ and memory states $\bar{\mathbf{m}}_i^t$ for updating the states of regions in frame I_{i+1}. The TD-LSTM unit recurrently updates the states of all regions in each frame, and thus the past temporal information in preceding frames can be utilized for updating each frame. The TD-Graph LSTM layer is illustrated in Figure 3.

3.4. Region-level Classification Module

Given the updated hidden states $\mathbf{h}_{i,j}^t$ for each node $v_{i,j}$, we use a region-level classification module to obtain the category confidences of all regions, that is, $\mathbf{rc}_i^t = \varphi(\mathbf{h}_i^t)$ of all M regions. Following the two-stream architecture of WSDDN [2], the region-level classification module contains a *detection stream* and a *classification stream*, and produces final classification scores by performing element-wise multiplication between them. The *classification stream* takes the region-level feature vectors $\mathbf{h}_i^t$ of all regions as the input and feeds it to a linear layer that outputs a set of class scores $\mathbf{S}_i^t \in \mathbb{R}^{M \times C}$ for C classes of all M regions. Here, we use the reproduced WSDDN in [16] that does not employ an additional softmax in the *classification stream*. These differences have a minor effect on the detection accuracy as has been discussed in [16]. The *detection stream* also takes $\mathbf{h}_i^t$ as the input and feeds it to another linear layer that outputs a set of class scores, giving a matrix of scores $\mathbf{L}_i^t \in \mathbb{R}^{M \times C}$. $\mathbf{L}_i^t$ is then fed to another softmax layer to normalize the scores over the regions in the frame. The final scores of all regions $\mathbf{rc}_i^t$ are obtained by taking the element-wise multiplication of the two scoring matrices $\mathbf{S}_i^t$ and $\mathbf{L}_i^t$. We sum all the region-level class scores $\mathbf{rc}_i^t$ to obtain the frame-level class prediction scores $\mathbf{pc}_i^t$.

4. Experiments

4.1. Dataset and Evaluation Measures

Dataset Analysis. We evaluate the action-drive weakly-supervised object detection performance on the *Charades* dataset [32]. The *Charades* video dataset is composed of daily indoor activities collected through Amazon Mechanical Turk. There are 157 action classes and on average 6.8 actions in each video, which occur in various orders and contexts. In order to detect objects in videos by using action labels, we only consider the action labels that are related to objects for training. Therefore, there are 66 action labels that are related to 17 object classes in our experiments. We show distribution of object classes (in a random subset of videos) in Figure 5 (a). The training set contains 7,542 videos. Videos are down-sampled to 1 fps and we only sample the frames assigned with action labels in each video. During training, only frame-level action labels are provided for each video.

In order to evaluate the video object detection performance over 17 daily object classes, we collect the bounding box annotations for 5,000 test frames from 200 videos in the Charades test set. The bounding box number distribution in each frame is shown in Figure 5 (b), ranging from 1 to 23 boxes appearing in the frame. More than 60% frames have more than 4 bounding boxes and most video frames exhibit severe motion blurs and low resolution. This poses more challenges for the object detection model compared to an image-based object detection dataset, such as the most popular PASCAL VOC [7] that is widely used in existing weakly-based object detection methods. Figure 4 further shows example frames with action labels on the *Charades* dataset. It can be seen that each action label only provides one piece of object class information for the frame that may contain several object classes, which can be regarded as the missing label issue for training a model under this action-driven setting. Moreover, the video frames often appear with a very cluttered background, blurry objects and diverse viewpoints, which are more challenging and realistic compared to existing image datasets (e.g., MS COCO[22] and ImageNet[29]) and video datasets (e.g., UCF101[36]).

Evaluation Measures. We evaluate the performance of both object detection and image classification tasks on *Charades*. For detection, we report the average precision (AP) at 50% intersection-over-union (IOU) of the detected boxes with the ground truth boxes. For classification, we also report the AP on frame-level object classification.

4.2. Implementation Details

Our TD-Graph LSTM adopts the VGG-CNN-F model [3] pre-trained on ImageNet ILSVRC 2012 challenge data [29] as the base model, and replaces the last pooling layer *pool*5 with an SPP layer [9] to be compatible with the first fully connected layer. We use the EdgeBoxes algorithm [48] to generate the top 500 regions that have width and height larger than 20 pixels as candidate regions for each frame. To balance the performance and time cost, we set the number of edges linked to each node K to 100. For training, we use stochastic gradient descent with momentum 0.9 and weight decay 5×10^{-4}. All weight matrices used in the TD-Graph LSTM units are randomly initialized from a uniform distribution of $[-0.1, 0.1]$. TD-Graph LSTM predicts the hidden and memory states with the same dimension as the previous region-level CNN features. Each mini-batch contains at most 6 consecutive sampled frames in a video. The network is trained on the *Charades* training set by using fine-tuning on all layers, including those of the pre-trained base CNN model. The experiments are run for 30 epochs for the model convergence. The learning rates are set to 10^{-5} for the first ten epochs, then decreased to 10^{-6}. All our models are implemented on the public Torch [5] platform, and all experiments are conducted on a single NVIDIA GeForce GTX TITAN X GPU with 12 GB memory. The runtime is 2.5 fps and 3.9 fps for training and testing respectively.

Figure 4. Several samples of key frames from videos in *Charades*. The action labels are given at the bottom of the image and the related objects are listed at the top of the image.

4.3. Results and Comparisons

We compare the proposed TD-Graph LSTM model with two state-of-the-art weakly-supervised learning methods on the *Charades* dataset, WSDDN [2] and ContextLocNet [16]. As both of the two methods were proposed for image-based weakly-supervised image object detection, here we run the source code of ContextLocNet [16] and their reproduced WSDDN[1] on the *Charades* dataset to make a fair comparison with our method. Their models are trained by treating the action-related object labels in each frame as the supervision information and are evaluated on each video frame. The difference between our model and WSDDN [2] is our usage of TD-Graph LSTM layers to leverage rich temporal correlations in the whole video. Similar to WSDDN, ContextLocNet is also a two stream model with an enhanced localization module using various

[1] https://github.com/vadimkantorov/contextlocnet

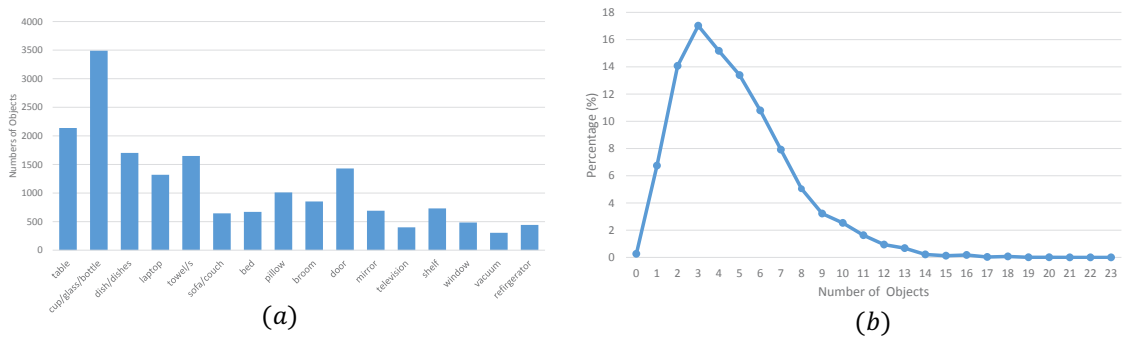

Figure 5. (a) The distribution of object classes appearing in the action labels of the training set. (b) The distribution of the ground truth bounding box numbers in each image of the test set.

Table 1. Per-class performance comparison of our proposed models with two state-of-the-art weakly-supervised learning methods when evaluating on the *Charades* dataset[32], test **classification** average precision (%).

Method	bed	broom	chair	cup	dish	door	laptop	mirror	pillow	refri	shelf	sofa	table	tv	towel	vacuum	window	mAP
WSDDN [2]	39.8	5.85	36.1	21	16.3	11.6	30.5	4.7	2.8	6.5	8.1	14.8	37.8	5	12.5	8.2	4.8	15.67
ContextLocNet [16]	43.37	5.65	38.95	16.62	12.46	8.67	27.75	4.5	3.51	**11.12**	**9.79**	15.67	37.44	**14.39**	9.72	16.36	3.97	16.47
TD-Graph LSTM w/o LSTM	32.54	5.875	31.69	**27.9**	15.79	14.19	18.81	**6.15**	8.35	4.5	9.3	24.33	33	8.26	14.7	7.68	**6.72**	15.89
TD-Graph LSTM w/o graph	25.04	6.51	43.79	21.54	15.6	**15.86**	19.57	5.61	9.32	6.2	9.02	**25.95**	39.2	8.85	**15.27**	**18.18**	5.63	17.13
TD-Graph LSTM	**47.62**	**12.26**	**45.07**	23.55	**16.7**	15.6	**30.9**	5.05	**17.64**	7.43	9.53	19.52	**43.29**	4.23	12.47	15.03	5.91	**19.52**

Table 2. Per-class performance comparison of our proposed models with two state-of-the-art weakly-supervised learning methods when evaluating on the *Charades* dataset[32], test **detection** average precision (%).

Method	bed	broom	chair	cup	dish	door	laptop	mirror	pillow	refri	shelf	sofa	table	tv	towel	vacuum	window	mAP
WSDDN [2]	2.38	0.04	1.17	0.03	**0.13**	0.31	2.81	0.28	0.02	0.12	0.03	0.41	1.74	1.18	0.07	0.08	0.22	0.65
ContextLocNet [16]	7.4	0.03	0.55	0.02	0.01	0.17	1.11	0.66	0	0.07	1.75	4.12	0.63	0.99	0.03	**0.75**	0.78	1.12
TD-Graph LSTM w/o LSTM	7.41	**0.05**	3	0.05	0.02	0.56	0.11	0.65	0.04	0.16	0.25	1.67	2.46	1.24	**0.11**	0.46	**1.46**	1.16
TD-Graph LSTM w/o graph	**9.69**	0.02	2.85	0.34	0.05	0.87	1.95	**0.69**	0.05	**0.44**	2.11	3.34	1.91	1.05	0.05	0.29	0.69	1.55
TD-Graph LSTM	9.19	0.04	**4.18**	**0.49**	0.11	**1.17**	**2.91**	0.3	**0.08**	0.29	**3.21**	**5.86**	**3.35**	**1.27**	0.09	0.6	0.47	**1.98**

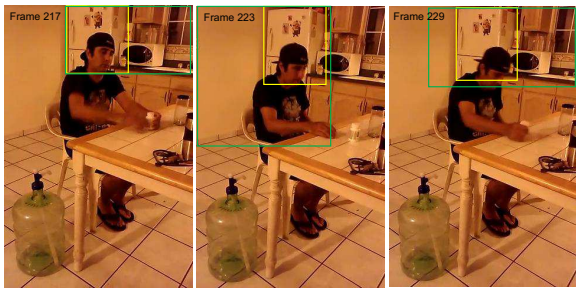

Figure 6. Our TD-Graph LSTM addresses well the missing label issue. It can successfully detect the refrigerator that is not referred to by any action labels (A green box shows the detection result and yellow box the ground truth.)

surrounding context. Specifically, we use the contrastive-S setup of ContextLocNet. All of these models use the same base model and region proposal method, i.e., VGG-CNN-F model [3] and EdgeBoxes [48].

We report the comparisons with two state-of-the-art on classification mAP and detection mAP in Table 1 and Table 2, respectively. It can be observed that our TD-Graph LSTM model substantially outperforms two baselines on both classification mAP and detection mAP, particularly, 3.05% higher than ContextLocNet [16] and 3.85% than WSDDN [2] in terms of classification mAP. Especially, our TD-Graph LSTM surpasses two baselines in small objects, e.g., over 14.13% for pillow class and 6.93% for cup class. Although our model and two baselines all obtain low detection mAP under this challenging setting, our TD-Graph LSTM still surpasses two baselines on detecting crowded and small objects in the video. The superiority of our TD-Graph LSTM clearly demonstrates its effectiveness in challenging action-driven weakly-supervised object detection where the missing label issue is quite severe and a considerable number of bounding boxes appear in each frame with very low quality. We further show the qualitative comparison with two state-of-the-arts in Figure 7. Our model is able to produce more precise object detection for even very small objects (e.g., the cup in the middle row) and objects with heavy occlusion (e.g., the sofa in the bottom row). Our TD-Graph LSTM takes the advantage of exploiting complex temporal correlations between region proposals by propagating knowledge into a whole dynamic temporal graph, which effectively alleviates the critical missing label issue, as shown in Figure 6.

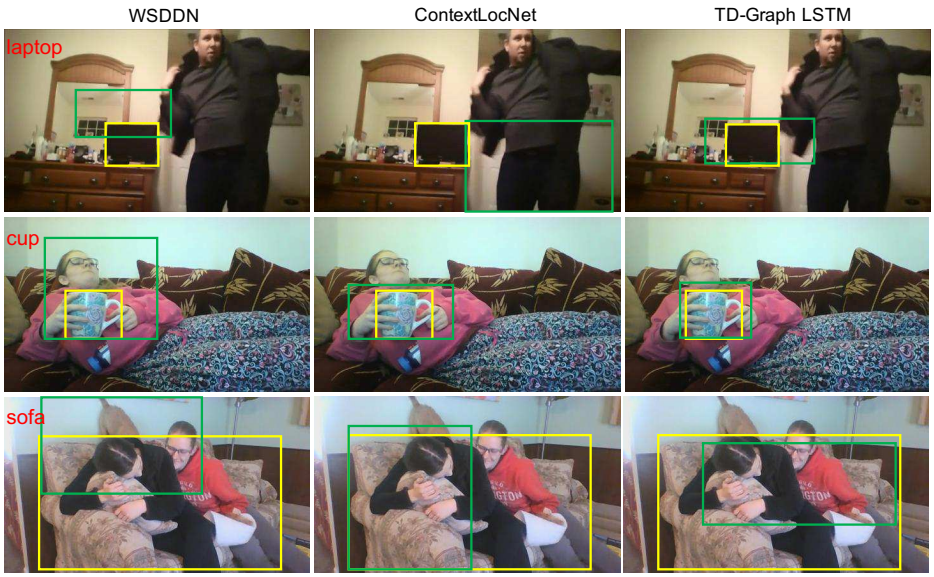

Figure 7. Qualitative comparisons with two state-of-the-arts on video object detection. The green boxes indicate detection results and yellow ones are the ground truth.

Table 3. Performance comparison of using different graph topologies when evaluating on the *Charades* dataset, test detection mAP (%) and classification mAP (%).

Method	det mAP	cls mAP
Ours w/o Graph	1.55	17.13
Ours w/ Mean Graph	1.41	16.92
Ours w/ Static Graph	1.89	17.97
Ours	**1.98**	**19.52**

4.4. Ablation Study

The results of model variants are reported in Table 1, Table 2 and Table 3.

The effectiveness of incorporating graph. The main difference between our TD-Graph with a conventional LSTM structure for sequential modeling is in propagating information over a dynamic graph structure. To verify its effectiveness, we thus compare our full model with the variant "TD-Graph LSTM w/o graph" that eliminates the edge connections between regions in consecutive frames, and updates the frame-level hidden and memory states with the original region-level features. Our TD-Graph LSTM consistently obtains better results over "TD-Graph LSTM w/o graph", which speaks to the advantage of incorporating a graph for the challenging action-driven object detection.

The effectiveness of temporal LSTM. We further verify that recurrent sequential modeling by the LSTM units over the temporal graph is beneficial for exploiting complex object motion patterns in daily videos. "TD-Graph LSTM w/o LSTM" indicates removing the LSTM units and directly aggregating the temporal context features to enhance features of each region. The performance gap between our full model and "TD-Graph LSTM w/o LSTM" verifies the benefits of adopting LSTM.

Dynamic graph vs Static graph vs Mean graph. Besides the proposed dynamic graph, another commonly used alternative is the fully-connected graph where each region is densely connected with all regions in the preceding frame; that is, "Ours w/ Static Graph" and "Ours w/ Mean Graph". "Ours w/ Static Graph" uses the adaptive edge weights similar to TD-Graph LSTM while "Ours w/ Mean Graph" uses the same weights for all edge connections. It can be seen that applying a dynamic graph structure can help significantly boost both detection and classification performance over other fully-connected graphs. The reason is that meaningful temporal correlations between regions can be discovered by the dynamic graph and leveraged to transfer motion context into the whole video.

5. Conclusion

In this paper, we propose a novel temporal dynamic graph LSTM architecture to address action-driven weakly-supervised object detection. It recurrently propagates the temporal context on a constructed dynamic graph structure for each frame. The global action knowledge in the whole video can be effectively leveraged for object detection in each frame, which helps alleviate the missing label problem. Extensive experiments on a large-scale daily-life action dataset *Charades* demonstrate the superiority of our model over the state-of-the-arts.

Acknowledgements: This work was supported by ONR MURI N000141612007 and Sloan Fellowship to AG. XL was supported by the Department of Defense under Contract No. FA8702-15-D-0002 with CMU for the operation of the Software Engineering Institute, a federally funded research and development center.

References

[1] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised detection with posterior regularization. In *BMVC*, 2014. 2

[2] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 5, 6, 7

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 6, 7

[4] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI*, 2016. 2

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 6

[6] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 2

[7] 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. 6

[8] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 2, 4

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 2, 6

[10] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2, 3

[12] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *CVPR*, 2016. 2, 3

[13] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu. Deep self-taught learning for weakly supervised object localization. In *CVPR*, 2017. 2

[14] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014. 2

[15] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016. 2

[16] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. ContextLocNet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016. 1, 5, 6, 7

[17] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015. 1, 2

[18] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan, and E. Xing. Interpretable structure-evolving LSTM. In *CVPR*, 2017. 2

[19] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *ICCV*, 2015. 2

[20] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph LSTM. In *ECCV*, 2016. 2

[21] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *CVPR*, 2016. 3

[22] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[24] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 3

[25] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2

[26] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 6

[30] S. Schulter, C. Leistner, P. M. Roth, and H. Bischof. Unsupervised object discovery and segmentation in videos. In *BMVC*, 2013. 1

[31] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015. 3

[32] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 1, 6, 7

[33] K. K. Singh, F. Xiao, and Y. J. Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *CVPR*, 2016. 2

[34] P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *ECCV*, 2012. 2

[35] H. O. Song, R. B. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, T. Darrell, et al. On learning to localize objects with minimal supervision. In *ICML*, 2014. 2

[36] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 1, 6

[37] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICLR*, 2015. 3

[38] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *arXiv:1707.02968*, 2017. 1

[39] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015. 2

[40] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 4

[41] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV*, 2014. 2

[42] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. *TPAMI*, 2016. 1

[43] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 2

[44] S. Yeung, O. Russakovsky, G. Mori, and F. Li. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 3

[45] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang. Revealing event saliency in unconstrained video collection. *IEEE Transactions on Image Processing*, 26(4):1746–1758, 2017. 2

[46] D. Zhang, D. Meng, and J. Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):865–878, 2017. 2

[47] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *ECCV*, 2016. 3

[48] C. L. Zitnick and P. Dollár. Edge Boxes: Locating object proposals from edges. In *ECCV*, 2014. 6, 7