Asynchronous Temporal Fields for Action Recognition

Gunnar A. Sigurdsson[1*] Santosh Divvala[2,3] Ali Farhadi[2,3] Abhinav Gupta[1,3]
[1]Carnegie Mellon University [2]University of Washington [3]Allen Institute for Artificial Intelligence
github.com/gsig/temporal-fields/

Abstract

Actions are more than just movements and trajectories: we cook to eat and we hold a cup to drink from it. A thorough understanding of videos requires going beyond appearance modeling and necessitates reasoning about the sequence of activities, as well as the higher-level constructs such as intentions. But how do we model and reason about these? We propose a fully-connected temporal CRF model for reasoning over various aspects of activities that includes objects, actions, and intentions, where the potentials are predicted by a deep network. End-to-end training of such structured models is a challenging endeavor: For inference and learning we need to construct mini-batches consisting of whole videos, leading to mini-batches with only a few videos. This causes high-correlation between data points leading to breakdown of the backprop algorithm. To address this challenge, we present an asynchronous variational inference method that allows efficient end-to-end training. Our method achieves a classification mAP of 22.4% on the Charades *[42] benchmark, outperforming the state-of-the-art (17.2% mAP), and offers equal gains on the task of temporal localization.*

1. Introduction

Consider the video shown in Figure 1: A man walks through a doorway, stands at a table, holds a cup, pours something into it, drinks it, puts the cup on the table, and finally walks away. Despite depicting a simple activity, the video involves a rich interplay of a sequence of actions with underlying goals and intentions. For example, the man stands at the table 'to take a cup', he holds the cup 'to drink from it', etc. Thorough understanding of videos requires us to model such interplay between activities as well as to reason over extensive time scales and multiple aspects of actions (objects, scenes, etc).

Most contemporary deep learning based methods have treated the problem of video understanding as that of only appearance and motion (trajectory) modeling [43, 53, 7, 27]. While this has fostered interesting progress in this domain, these methods still struggle to outperform models based on hand-crafted features, such as Dense Trajectories [56]. Why such a disconnect? We argue that video understanding requires going beyond appearance modeling, and necessitates reasoning about the activity sequence as well as higher-level constructs such as intentions. The recent emergence of large-scale datasets containing rich sequences of realistic activities [42, 63, 60] comes at a perfect time facilitating us to explore such complex reasoning.

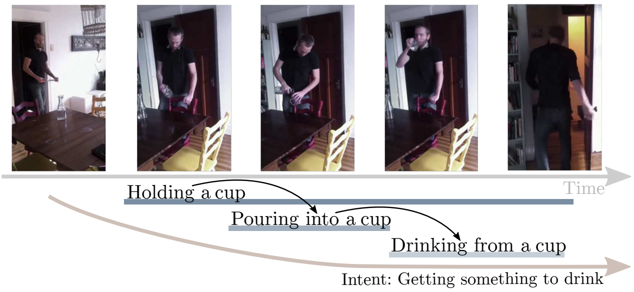

Figure 1. Understanding human activities in videos requires jointly reasoning about multiple aspects of activities, such as 'what is happening', 'how', and 'why'. In this paper, we present an end-to-end deep structured model over time trained in a stochastic fashion. The model captures rich semantic aspects of activities, including *Intent* (why), *Category* (what), *Object* (how). The figure shows video frames and annotations used in training from the *Charades* [42] dataset.

But what is the right way to model and reason about temporal relations and goal-driven behaviour? Over the last couple of decades, graphical models such as Conditional Random Fields (CRFs) have been the prime vehicles for structured reasoning. Therefore, one possible alternative is to use ConvNet-based approaches [19] to provide features for a CRF training algorithm. Alternatively, it has been shown that integrating CRFs with ConvNet architectures and training them in an end-to-end manner provides substantial improvements in tasks such as segmentation and situation recognition [66, 1, 62].

Inspired by these advances, we present a deep-structured model that can reason temporally about multiple aspects of activities. For each frame, our model infers the activity cate-

[*] Work was done while Gunnar was an intern at AI2.

gory, object, action, progress, and scene using a CRF, where the potentials are predicted by a jointly end-to-end trained ConvNet over all predictions in all frames. This CRF has a latent node for the intent of the actor in the video and pairwise relationships between all individual frame predictions.

While our model is intuitive, training it in an end-to-end manner is a non-trivial task. Particularly, end-to-end learning requires computing likelihoods for individual frames and doing joint inference about all connected frames with a CRF training algorithm. This is in stark contrast with the standard stochastic gradient descent (SGD) training algorithm (backprop) for deep networks, where we require mini-batches with a large number of independent and uncorrelated samples, not just a few whole videos. In order to handle this effectively: (1) we relax the Markov assumption and choose a fully-connected temporal model, such that each frame's prediction is influenced by all other frames, and (2) we propose an asynchronous method for training fully-connected structured models for videos. Specifically, this structure allows for an implementation where the influence (messages) from other frames are approximated by emphasizing influence from frames computed in recent iterations. They are more accurate, and show advantage over being limited to only neighboring frames. In addition to being more suitable for stochastic training, fully-connected models have shown increased performance on various tasks [18, 66].

In summary, our key contributions are: (a) a deep CRF based model for structured understanding and comprehensive reasoning of videos in terms of multiple aspects, such as action sequences, objects, and even intentions; (b) an asynchronous training framework for expressive temporal CRFs that is suitable for end-to-end training of deep networks; and, (c) substantial improvements over state-of-the-art, increasing performance from 17.2% mAP to 22.4% mAP on the challenging Charades [42] benchmark.

2. Related Work

Understanding activities and actions has an extensive history [32, 59, 22, 17, 23, 2, 26, 56, 29, 21]. Interestingly, analyzing actions by their appearance has gone through multiple iterations. Early success was with handcrafted representations such as Space Time Interest Points (STIP) [22], 3D Histogram of Gradient (HOG3D) [17], Histogram of Optical Flow (HOF) [23], and Motion Boundary Histogram [2]. These methods capture and analyze local properties of the visual-temporal datastream. In the past years, the most prominent hand-crafted representations have been from the family of trajectory based approaches [26, 56, 29, 21], where the Improved Dense Trajectories (IDT) [56] representation is in fact on par with state-of-the-art on multiple recent datasets [8, 42].

Recently there has been a push towards mid-level representations of video [37, 46, 13, 20], that capture beyond local properties. However, these approaches still used handcrafted features. With the advent of deep learning, learning representations from data has been extensively studied [14, 15, 44, 57, 52, 53, 24, 7, 61, 55, 40, 3]. Of these, one of the most popular frameworks has been the approach of Simonyan et al. [44], who introduced the idea of training separate color and optical flow networks to capture local properties of the video.

Many of those approaches were designed for short clips of individual activities and hence do not generalize well to realistic sequences of activities. Capturing the whole information of the video in terms of temporal evolution of the video stream has been the focus of some recent approaches [51, 6, 12, 35, 49, 30]. Moving towards more expressive deep networks such as LSTM has become a popular method for encoding such temporal information [48, 4, 65, 50, 58, 41, 64]. Interestingly, while those models move towards more complete understanding of the full video stream, they have yet to significantly outperform local methods [44] on standard benchmarks.

A different direction in understanding comes from reasoning about the complete video stream in a complementary direction — Structure. Understanding activities in a human-centric fashion encodes our particular experiences with the visual world. Understanding activities with emphasis on objects has been a particularly fruitful direction [25, 36, 9, 34, 54]. In a similar vein, some works have also tried modeling activities as transformations [58] or state changes [5]. Recently, there has been significant progress in modelling the complete human-centric aspect, where image recognition is phrased in terms of objects and their roles [62, 10]. Moving beyond appearance and reasoning about the state of *agents* in the images requires understanding human intentions [16, 31]. This ability to understand people in terms of beliefs and intents has been traditionally studied in psychology as the Theory of mind [33].

How to exactly model structure of the visual and temporal world has been the pursuit of numerous fields. Of particular interest is work that combines the representative power of deep networks with structured modelling. Training such models is often cumbersome due to the differences in jointly training deep networks (stochastic sampling) and sequential models (consecutive samples) [28, 66]. In this work, we focus on fully-connected random fields, that have been popular in image segmentation [18], where image filtering was used for efficient message passing, and later extended to use CNN potentials [39].

3. Proposed Method

Given a video with multiple activities, our goal is to understand the video in terms of activities. Understanding activities requires reasoning about objects being interacted

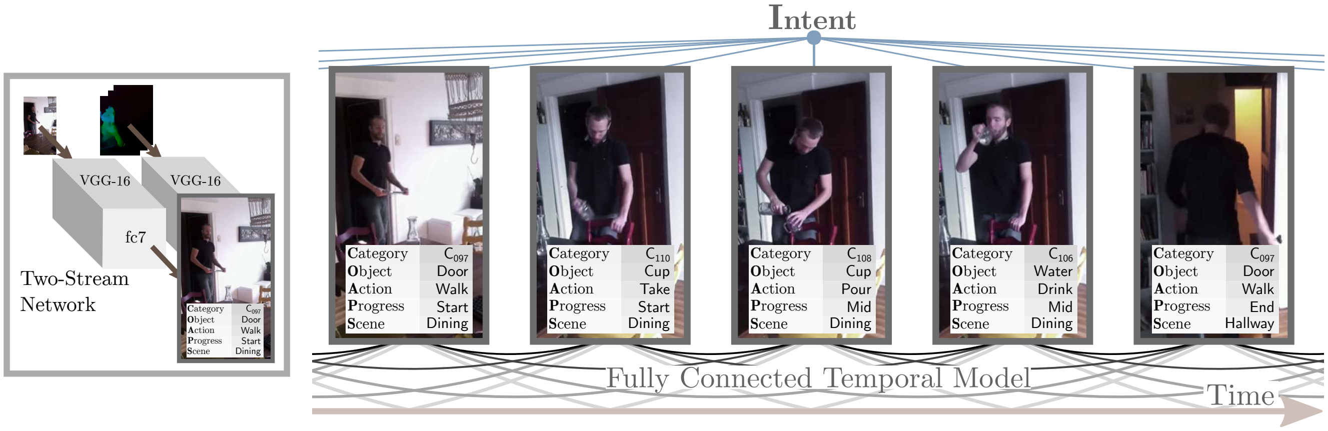
Intent
VGG-16
VGG-16
fc7
Two-Stream Network
Category C097 Object Door Action Walk Progress Start Scene Dining
Category C097 Object Door Action Walk Progress Start Scene Dining
Category C110 Object Cup Action Take Progress Start Scene Dining
Category C108 Object Cup Action Pour Progress Mid Scene Dining
Category C106 Object Water Action Drink Progress Mid Scene Dining
Category C097 Object Door Action Walk Progress End Scene Hallway
Fully Connected Temporal Model
Time

Figure 2. An overview of our structured model. The semantic part captures *object*, *action*, etc. at each frame, and temporal aspects captures those over time. On the left side, we show how for each timepoint in the video, a Two-Stream Network predicts the potentials. Our model jointly reasons about multiple aspects of activities in all video frames. The *Intent* captures groups of activities of the person throughout the whole sequence of activities, and fine-grained temporal reasoning is through fully-connected temporal connections.

with, the place where the interaction is happening, what happened before and what happens after this current action and even the intent of the actor in the video. We incorporate all these by formulating a deep Conditional Random Field (CRF) over different aspects of the activity over time. That is, a video can be interpreted as a graphical model, where the components of the activity in each frame are nodes in the graph, and the model potentials are the edges in the graph.

In particular, we create a CRF which predicts activity, object, etc., for every frame in the video. For reasoning about time, we create a *fully-connected temporal CRF*, referred as Asynchronous Temporal Field in the text. That is, unlike a linear-chain CRF for temporal modelling (the discriminative counterpart to Hidden Markov Models), each node depends on the state of every other node in the graph. We incorporate intention as another latent variable which is connected to all the action nodes. This is an unobserved variable that influences the sequence of activities. This variable is the common underlying factor that guides and better explains the sequence of actions an agent takes. Analysis of what structure this latent variable learns is presented in the experiments. Our model has three advantages: (1) it addresses the problem of long-term interactions; (2) it incorporates reasoning about multiple parts of the activity, such as objects and intent; and (3) more interestingly, as we will see, it allows for efficient end-to-end training in an asynchronous stochastic fashion.

3.1. Architecture

In this work we encode multiple components of an activity. Each video with T frames is represented as $\{X_1, \ldots, X_T, I\}$ where X_t is a set of frame-level random variables for time step t and I is an unobserved random variable that represent global intent in the entire video. We can further write $X_t = \{C_t, O_t, A_t, P_t, S_t\}$, where C is the activity category (e.g., 'drinking from cup'), O corresponds to the object (e.g., 'cup'), A represents the action (e.g., 'drink'), P represents the progress of the activity $\{\text{start, middle, end}\}$, and S represents the scene (e.g. 'Dining Room'). For clarity in the following derivation we will refer to all the associated variables of X_t as a single random variable X_t. A more detailed description of the CRF is presented in the appendix.

Mathematically we consider a random field $\{X, I\}$ over all the random variables in our model ($\{X_1, \ldots, X_T, I\}$). Given an input video $V{=}\{V_1, \ldots, V_T\}$, where V_t is a video frame, our goal is to estimate the maximum a posteriori labeling of the random field by marginalizing over the intent I. This can be written as:

$$\mathbf{x}^* = \arg\max_x \sum_I P(x, I|V). \tag{1}$$

For clarity in notation, we will drop the conditioning on V and write $P(X, I)$. We can define $P(X, I)$ using Gibbs distribution as: $P(X, I){=}\frac{1}{Z(\mathbf{V})}\exp\left(-E(x, I)\right)$ where $E(x, I)$ is the Gibbs energy over x. In our CRF, we model all unary and pairwise cliques between all frames $\{X_1, \ldots, X_T\}$ and the intent I. The Gibbs energy is:

$$E(\mathbf{x}, I) = \underbrace{\sum_i \phi_{\mathcal{X}}(x_i)}_{\text{Semantic}} + \underbrace{\sum_i \phi_{\mathcal{XI}}(x_i, I) + \sum_{\substack{i,j \\ i \neq j}} \phi_{\mathcal{XX}}(x_i, x_j)}_{\text{Temporal}}, \tag{2}$$

where $\phi_{\mathcal{XX}}(x_i, x_j)$ is the potential between frame i and frame j, and $\phi_{\mathcal{XI}}(x_i, I)$ is the potential between frame i and the intent. For notational clarity $\phi_{\mathcal{X}}(x_i)$ incorporates all unary and pairwise potentials for C_t, O_t, A_t, P_t, S_t. The model is best understood in terms of two aspects: Semantic

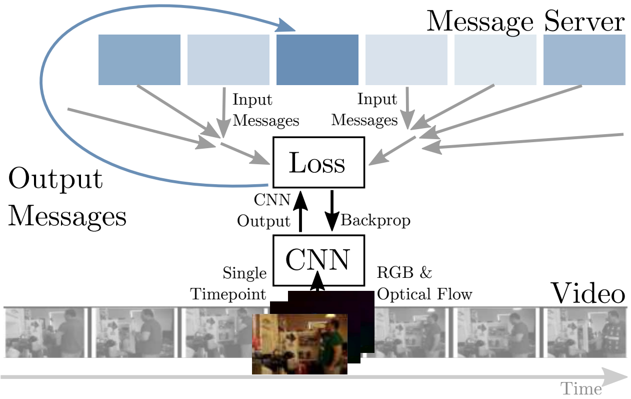

Figure 3. Illustration of the learning algorithm, and the message passing structure. Each timepoint that has been processed has a message (Blue highlights messages that have recently been computed). The loss receives a combination of those messages, uses those to construct new messages, and updates the network.

aspect, which incorporates the local variables in each frame $(C_t, O_t, A_t, P_t, S_t)$; and Temporal aspect, which incorporates interactions among frames and the intent I. This is visualized in Figure 2. We will now explain the semantic, and temporal potentials.

Semantic aspect The frame potential $\phi_\mathcal{X}(x_i)$ incorporates the interplay between activity category, object, action, progress and scene, and could be written explicitly as $\phi_\mathcal{X}(C_t, O_t, A_t, P_t, S_t)$. In practice this potential is composed of unary, pairwise, and tertiary potentials directly predicted by a CNN. We found predicting only the following terms to be sufficient without introducing too many additional parameters: $\phi_\mathcal{X}(C_t, O_t, A_t, P_t, S_t) {=} \phi(O_t, P_t) {+} \phi(A_t, P_t) {+} \phi(O_t, S_t) {+} \phi(C_t, O_t, A_t, P_t)$ where we only model the assignments seen in the training set, and assume others are not possible.

Temporal aspect The temporal aspect of the model is both in terms of the frame-intent potentials $\phi_{\mathcal{X}\mathcal{I}}(x_i, I)$ and frame-frame potentials $\phi_{\mathcal{X}\mathcal{X}}(x_i, x_j)$. The frame-intent potentials are predicted with a CNN from video frames (pixels and motion). The pairwise potentials $\phi_{\mathcal{X}\mathcal{X}}(x_i, x_j)$ for two time points i and j in our model have the form:

$$\phi_{\mathcal{X}\mathcal{X}}(x_i, x_j) = \mu(x_i, x_j) \sum_m w^{(m)} k^{(m)}(v_i, v_j), \tag{3}$$

where μ models the asymmetric affinity between frames, w are kernel weights, and each $k^{(m)}$ is a Gaussian kernel that depends on the videoframes v_i and v_j. In this work we use a single kernel that prioritises short-term interactions:

$$k(v_i, v_j) = \exp\left(-\frac{(j-i)^2}{2\sigma^2}\right) \tag{4}$$

The parameters of the general asymmetric compatibility function $\mu(x_i, x_j)$ are learned from the data, and σ is a hyper-parameter chosen by cross-validation.

3.2. Inference

While it is possible to enumerate all variable configurations in a single frame, doing so for multiple frames and their interactions is intractable. Our algorithm uses a structured variational approximation to approximate the full probability distribution. In particular, we use a mean-field approximation to make inference and learning tractable. With this approximation, we can do inference by keeping track of message between frames, and asynchronously train one frame at a time (in a mini-batch fashion).

More formally, instead of computing the exact distribution $P(X, I)$ presented above, the structured variational approximation finds the distribution $Q(X, I)$ among a given family of distributions that best fits the exact distribution in terms of KL-divergence. By choosing a family of tractable distributions, it is possible to make inference involving the ideal distribution tractable. Here we use $Q(X, I) = Q_\mathcal{I}(I) \prod_i Q_i(x_i)$, the structured mean-field approximation. Minimizing the KL-divergence between those two distributions yields the following iterative update equation:

$$\begin{aligned} Q_i(x_i) \propto \exp\Big\{ & \phi_\mathcal{X}(x_i) + \mathrm{E}_{U \sim Q_\mathcal{I}}\left[\phi_{\mathcal{X}\mathcal{I}}(x_i, U)\right] \\ & + \sum_{j>i} \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{X}\mathcal{X}}(x_i, U_j)\right] \Big\} \\ & + \sum_{j<i} \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{X}\mathcal{X}}(U_j, x_i)\right] \Big\} \qquad (5) \\ Q_\mathcal{I}(I) \propto \exp\Big\{ & \sum_j \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{X}\mathcal{I}}(U_j, I)\right] \Big\} \qquad (6) \end{aligned}$$

where Q_i is marginal distribution with respect to each of the frames, and $Q_\mathcal{I}$ is the marginal with respect to the intent. An algorithmic implementation of this equation is as presented in Algorithm 1.

Algorithm 1 Inference for Asynchronous Temporal Fields

```
Initialize Q                                        ▷ Uniform distribution
while not converged do
    Visit frame $i$
    Get $\sum_{j>i} \mathrm{E}_{U_j \sim Q_j}[\phi_{\mathcal{X}\mathcal{X}}(x_i, U_j)]$
    Get $\sum_{j<i} \mathrm{E}_{U_j \sim Q_j}[\phi_{\mathcal{X}\mathcal{X}}(U_j, x_i)]$
    Get $\sum_{j} \mathrm{E}_{U_j \sim Q_j}[\phi_{\mathcal{X}\mathcal{I}}(U_j, I)]$
    while not converged do
        Update $Q_i$ and $Q_\mathcal{I}$ using Eq. 6
    Send $\mathrm{E}_{U \sim Q_i}[\phi_{\mathcal{X}\mathcal{X}}(x, U)]$
    Send $\mathrm{E}_{U \sim Q_i}[\phi_{\mathcal{X}\mathcal{X}}(U, x)]$
    Send $\mathrm{E}_{U \sim Q_i}[\phi_{\mathcal{X}\mathcal{I}}(U, I)]$
```

Here 'Get' and 'Send' refer to the message server, and $f(x)$ is a message used later by frames in the same video. The term message server is used for a central process that keeps track of what node in what video sent what message, and

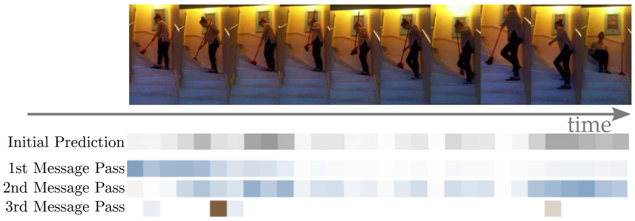

Figure 4. Evolution of prediction with increasing messages passes. The first row shows the initial prediction for the category *tidying with a broom* without any message passing, where darker colors correspond to higher likelihood, blue is then an increase in likelihood, and brown decrease. In the first message pass, the confidence of high predictions gets spread around, and eventually increases the confidence of the whole prediction.

distributes them accordingly when requested. In practice, this could be implemented in a multi-machine setup.

3.3. Learning

Training a deep CRF model requires calculating derivatives of the objective in terms of each of the potentials in the model, which in turn requires inference of $P(X,I|V)$. The network is trained to maximize the log-likelihood of the data $l(X) = \log \sum_I P(x,I|V)$. The goal is to update the parameters of the model, for which we need gradients with respect to the parameters. Similar to SGD, we find the gradient with respect to one part of the parameters at a time, specifically with respect to one potential in one frame. That is, $\phi^i_{\mathcal{X}}(\hat{x})$ instead of $\phi_{\mathcal{X}}(\hat{x})$. The partial derivatives of this loss with respect to each of the potentials are as follows:

$$\frac{\partial l(X)}{\partial \phi^i_{\mathcal{X}}(\hat{x})} = \mathbf{1}_{x=\hat{x}} - Q_i(\hat{x}) \tag{7}$$

$$\frac{\partial l(X)}{\partial \phi^i_{\mathcal{XI}}(\hat{x},\hat{I})} = \frac{\exp\sum_j \phi_{\mathcal{XI}}(x_j,\hat{I})}{\sum_I \exp\sum_j \phi_{\mathcal{XI}}(x_j,I)}\mathbf{1}_{x=\hat{x}} - Q_i(\hat{x})Q_{\mathcal{I}}(\hat{I}) \tag{8}$$

$$\begin{aligned}\frac{\partial l(X)}{\partial \mu^i(a,b)} &= \sum_{j>i} \mathbf{1}_{x=a} k(v_i,v_j) - Q_i(\hat{x}) \sum_{j>i} Q_{\mathcal{I}}(b) k(v_i,v_j) \\ &+ \sum_{j<i} \mathbf{1}_{x=b} k(v_j,v_i) - Q_i(\hat{x}) \sum_{j<i} Q_{\mathcal{I}}(a) k(v_i,v_j) \end{aligned} \tag{9}$$

where $\phi^i_{\mathcal{X}}(\hat{x})$ and $\phi^i_{\mathcal{XI}}(\hat{x},\hat{I})$ is the frame and frame-intent potentials of frame i, and we use $\hat{x}$ to distinguish between the labels and variables the derivative is taken with respect to. $\mu^i(a,b)$ are the parameters of the asymmetric affinity kernel with respect to frame i, and $\mathbf{1}_{x=\hat{x}}$ is a indicator variable that has the value one if the ground truth label corresponds to the variable. Complete derivation is presented in the appendix. These gradients are used to update the underlying CNN model. These update equations lead to the learning procedure presented in Algorithm 2.

Figure 3 graphically illustrates the learning procedure. Since the videos are repeatedly visited throughout the training process, we do not have to run multiple message passes to calculate each partial gradient. This shares ideas with contrastive divergence [11, 38]. Given a single video at test time, we visualize in Figure 4 how the predictions changes as the distribution converges with multiple messages passes.

Algorithm 2 Learning for Asynchronous Temporal Fields

```
Given videos $\mathcal{V}$
while not converged do
    for each example in mini-batch do
        Sample frame $v \in \mathbf{V} \subseteq \mathcal{V}$
        Get incoming messages
        Update $Q_i$ and $Q_{\mathcal{I}}$
        Find gradients with Eq. 7-9
        Backprop gradients through CNN
        Send outgoing messages
```

Message Passing The key thing to note is all the incoming messages are of the form $M(z){=}\sum_j f_j(z)$ where f_j is some function from node j; for e.g., $M(z) = \sum_j \mathrm{E}_{U_j \sim Q_j}[\phi_{\mathcal{XI}}(U_j,z)] = \sum_j f_j(z)$ from Algorithm 1. We use the following approximation during training:

$$M(z) {\approx} \frac{h}{\sum_j d^j} \sum_j d^j f_{J(j)}(z), \tag{10}$$

where $d \in [0,1]$ is a discount factor, h is a hyperparameter, and $J(\cdot)$ is an ordering of the messages in that video based on the iteration in which the message was computed. The messages are a weighted combination of stored messages.

4. Experimental Results and Analysis

We analyzed the efficacy of our model on the challenging tasks of video activity classification and temporal localization. In addition, we investigated the different parts of the model, and will demonstrate how they operate together.

Dataset Recent years have witnessed an emergence of large-scale datasets containing sequences of common daily activities [42, 63, 60]. For our evaluation, we chose the *Charades* dataset [42]. This dataset is a challenging benchmark containing 9,848 videos across 157 action classes with 66,500 annotated activities, including nouns (objects), verbs (actions), and scenes. A unique feature of this dataset is the presence of complex co-occurrences of realistic human-generated activities making it a perfect test-bed for our analysis. We evaluate video classification using the evaluation criteria and code from [42]. Temporal localization is evaluated in terms of per-frame classification using the provided temporal annotations.

Implementation details We use a VGG16 network [45] with additional layers to predict the model potentials (Figure 5). We train both a network on RGB frames, and stacks of optical flow images, following the two-stream architecture [44]. The main challenge in training the network is the increase in the output layer size. For the larger potentials,

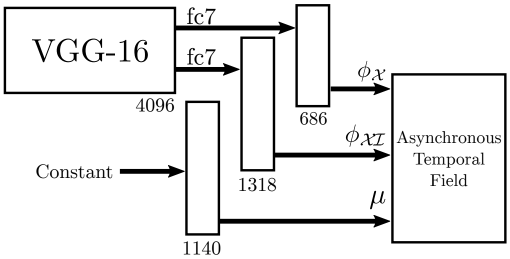

Figure 5. The VGG-16 variant predicts the potentials for both RGB and Flow. The network predicts the values of all potentials except one (in this figure we group the frame potentials $\phi_{\mathcal{X}}$ into one layer for clarity). The model is trained end-to-end by passing gradients from the Asynchronous Temporal Field through the network.

we used the following structure to go from fc7 to $\phi_{\mathcal{XI}}$: Linear layer (4096 to 100), ReLU, Dropout, Linear layer (100 to the potential values).

The input to the RGB network is an image of size 224×224×3 where we crop random location, size, and aspect ratio. We use data augmentation with color jitter and PCA lighting noise. The RGB network was pretrained on ImageNet. The input to the Flow network is a stack of 10 consecutive optical flow frames at 24 FPS starting with the current frame. Since each optical flow has two channels, the input size is 224×224×20 as in [44]. The Flow network was pretrained on UCF101 [47] as in Sigurdsson et al. [42], and random cropped in the same way as RGB.

We follow the training setup in Charades [42] and consider a frame to have one activity label at a time. Even so, our method is still able to reason about other activities in the video. Convergence of the model is evaluated using the approximate distribution $Q_i(X)$ at each frame. The Charades dataset has the property that scenes were chosen at random for each sequence of activities. For this reason, we found reasoning about scenes to reduce the performance, and the weight of that term was lowered in the model.

To obtain annotations for action progress p_t, we split each activity annotation into three equally sized parts. All layers of the network are trained with a batch size of 240 and a learning rate of 10^{-3} (RGB), 10^{-5} (Flow). Learning rate was reduced by a factor of 10 every 30k iterations for RGB, and every 140k iterations for Flow. The value of the message decay parameter d was set to $d = 0.9$, and the standard deviation σ in (4) was set to 6.25 sec (150 frames).

For testing, we sampled 25 equally spaced frames from the video and synchronously pass messages between the frames until convergence (10 message passes). The predictions of the RGB and Flow networks are combined in a probabilistic fashion by multiplying their probabilistic predictions for each class. More implementation details may be found in the appendix. The networks were implemented in Torch, and the code is available on project page.

Diverse batches As highlighted in Section 1, the standard way of sampling batches for temporal models results in high correlation between data points leading to a breakdown of the SGD. To understand the importance of having many diverse examples from multiple videos, we compare the convergence of our method to two alternatives using homogeneous batches: CNN+LSTM from Ng et al. [65], and a synchronous version of our method, where each batch contains full videos (only three videos fit into each mini-batch). We do synchronous message passing until convergence before calculating gradients for backprop. Figure 6 shows that our asynchronous training method, containing more diverse training batches, has faster and more stable convergence.

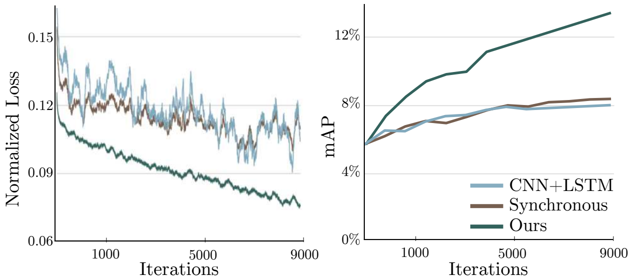

Figure 6. Convergence of our method compared to other methods that capture temporal structure. Our asynchronous training method contains more diverse batches, has faster and more stable convergence, and reaches higher accuracy on the test set.

4.1. Video Classification

Given a video, the task here is to verify whether it contains one or several of the 157 activity categories. Classification accuracy is measured with the standard mean average precision (mAP) criterion, where a prediction is given for each video. This task has been shown to be highly challenging, with the state-of-the-art non-ensemble methods reaching an mAP of only 17.2%, particularly as each video in this dataset has a sequence of multiple fine-grained activities with a real-world long-tailed activity distribution.

We trained our models using the provided training split following the procedure outlined in Section 3. To make predictions for the whole video, we marginalize out everything except the activity category for 25 equidistant frames in the video. The score for each activity category is the maximum across all frames following the setup from [42]. In our analysis, we include the provided non-ensemble baselines from [42] as well as the following additional baselines:

Two-Stream++. We reimplemented the network described in [42], which follows Simonyan et al. [45], with the same parameters. We added data augmentation and finetuned all layers of the network. The performance of only the RGB stream is included (*RGB++*). We also consider *Two-Stream Extended* which is the same network, but the Flow network was trained for 25 times more iterations than the RGB network (two weeks of computation on a Titan X

Approach	mAP	Approach	mAP
Random [42]	5.9	RGB++	15.6
C3D [53]	10.9	Two-Stream++	16.8
AlexNet [19]	11.3	Two-Stream+LSTM	17.8
IDT [56]	17.2	Two-Stream Extended	18.6
Two-Stream [43]	14.3	Ours (RGB Only)	18.3
		Ours	**22.4**

Table 1. Video classification results on Charades [42]. The left shows the published baselines from [42] and the right show additional new baselines. Our proposed approach outperforms all competing methods on this dataset.

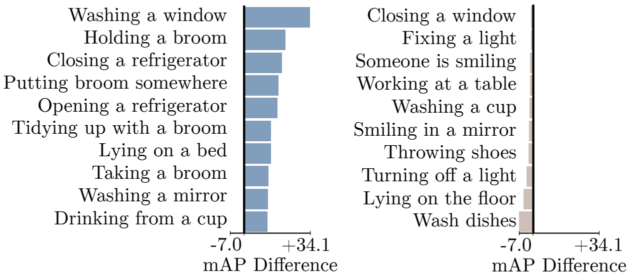

Figure 7. The classes with the highest positive and negative difference between our method and Two-Stream (no structure). Our method does better on many classes, without doing much worse on any. In particular, activities that have temporal structure, such as Opening/Closing a refrigerator have significantly higher performance, since our model can reason jointly about those.

GPU). Combined with the augmentation, we found this to non-trivially increase the accuracy.

Two-Stream+LSTM. We followed the method outlined in [65] to jointly train a LSTM on top of the two-stream network. We trained both an RGB and an Optical Flow network using the same setup from [42]. The trained networks from Two-Stream++ were used to initialize the models.

Table 1 displays the accuracy obtained by our method along with the baselines. Our proposed approach obtains an mAP of 22.4% substantially outperforming the Two-stream Extended baseline at 18.6% mAP, and the IDT baseline at 17.2%. Our method reasons over significantly larger timescales and multiple aspects of the activities. To ascertain this, we highlight in Figure 7, the activity classes with the highest positive and negative difference between our method and the Two-Stream network. It is interesting to note that two of those activities are *opening* and *closing* a refrigerator, that arguably have a significant causal structure (an *open* refrigerator was *opened* at some point), which our model harnesses to significantly increase the accuracy.

Ablation studies To study the contribution of different model parts, we also train ablated versions of our model separately choosing the best hyperparameters for each version. In addition to our model with only RGB or Flow, we also consider dropping $\phi_{\mathcal{X}\mathcal{X}}$ (i.e., no sequential information), $\phi_{\mathcal{X}\mathcal{I}}$ (i.e., no intent), both (i.e., only semantic information), and further dropping $\phi_{\mathcal{X}}$ (i.e., dropping all structure). Figure 8 shows that semantic reasoning improves over the baseline. Further, while both $\phi_{\mathcal{X}\mathcal{I}}$ and $\phi_{\mathcal{X}\mathcal{X}}$ capture temporal information, they are complementary.

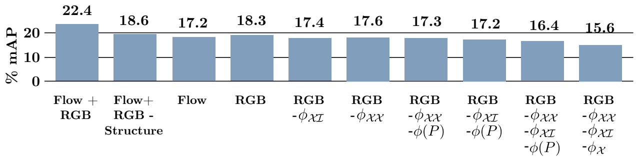

Figure 8. Ablation analysis for our proposed model. Y-axis is video classification mAP %. Each factor helps in improving the overall model performance. $\phi(P)$ indicates dropping the 'progress' term within the semantic factor $\phi_{\mathcal{X}}$.

	Random	RGB	Two-Stream++	Two-Stream +LSTM	Two-Stream Extended	Ours
Standard	2.42	7.89	8.94	9.60	9.37	**9.69**
Post-processing	2.42	9.05	10.9	10.4	11.6	**12.8**

Table 2. Temporal localization results (mAP %) on the Charades [42] dataset. Our proposed method outperforms the LSTM model, and is also more tractable to train at a large-scale.

4.2. Temporal Localization

To measure the ability of the methods to temporally localize and understand when exactly activities happen, we adapt the benchmark of [42] to evaluate with the same mAP metric but on individual frames. That is, instead of having a single prediction per video, evaluation is now split into 25 equidistant timepoints having zero or more activities, and the models make a prediction for each of those*. We find this way of evaluating localization robust to annotation ambiguity, and informative for challenging datasets. All hyperparameters were kept equal between localization and classification experiments. All baselines are run on 75 frames across the video, and then every third frame selected for a total of 25 frames. We also considered methods with *post-processing* where the model predictions for the 75 frames are averaged across 30 frames to obtain more spatial consistency, and then 25 frames selected as before.

Table 2 shows that our method outperforms the alternatives, including the LSTM model which has been shown to be a powerful temporal modeling tool, but challenging to train on top of a two-stream network due to correlations between consecutive samples. These results demonstrate the our method is tractable way of training end-to-end structured models to understand activities. Interestingly, our method still benefits from adding post-processing, significantly more than the LSTM baseline, likely since our method is reasoning on larger time-scales. This suggests

*This evaluation code has been included as a part of the Charades dataset (`allenai.org/plato/charades/`).

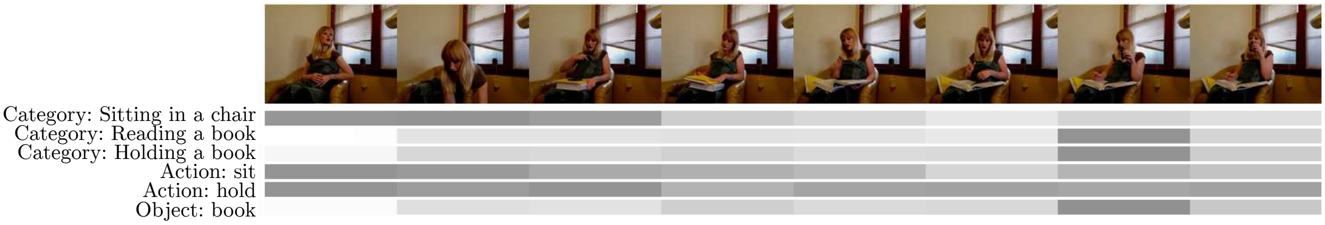

Figure 9. Model predictions for a sample video. We see the interplay between categories, objects and actions over time. For example, model becomes confident about the action *sit* early, which aids the understanding of *Sitting in a chair* once the chair becomes visible, and helps predicting *Reading a book*. Darker colors represent higher likelihood, and we average predictions to correspond to each frame.

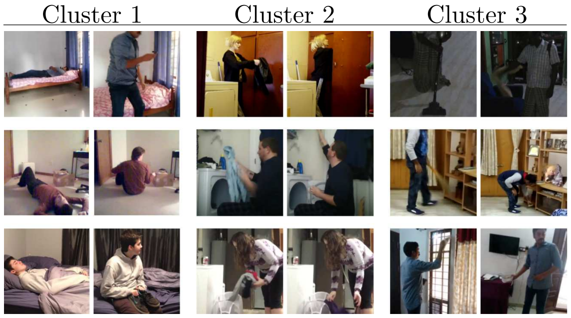

Figure 10. To visualize the learned intent, we cluster videos based on intent. In *Cluster 1*, the model captures the intent of get up from lying down. In *Cluster 2*, folding clothes is followed by putting them away, and *Cluster 3* shows cleaning with a broom/vacuum/towel, followed by picking up things.

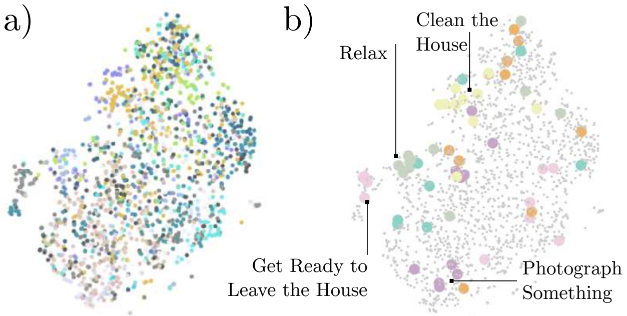

Figure 11. t-SNE visualization for the learned intent. Each point corresponds to a video. In a) it is colored based on its activity shared by the most of the 10 nearest neighbors (each video has multiple actions). In b) videos with 6 annotated intent types are emphasized with larger points colored by the type.

that our model could further benefit from joint training with additional kernels in the temporal term.

Qualitative visualization A key advantage of our model is the structured understanding of videos in terms of multiple aspects, such as action sequences, objects, and even intentions. To visualize this, we display predictions over time in Figure 9 for the three most confident activity categories, two most confident actions, and the most confident object. More examples are presented in the Appendix.

Interpretation of Intent In our model, the intent I is a continuous distribution over the latent variables. To get an insight into how our model learns the intent, we ran a simple experiment that clustered videos in the dataset that have the most similar inferred intent distributions. The first cluster in Figure 10 shows the model captures the simple intent that the person intends to get up from lying down. In the videos, these actions are 10-20 seconds apart, demonstrating that the intent helps reason over large time scales.

In order to further analyze the 'intent' variable, we plot the t-SNE embedding of the intent variable for the videos in the test set. We see that there is clear clustering of similar videos in Fig. 11a. We also annotated 10 types of intent (100 videos total). More details are presented in the Appendix. We observe that the intent representation preserves some of the intent types in Fig. 11b. Quantitatively, even without mitigating outliers, the average distance (in 10^{-3}) between pairs of videos within an intent type was 6.02 compared to 7.25 (σ=1.06) for any points, and the difference is significant for 5 of 10 intent types (p=0.1). This tentatively suggest that the intent captures interesting structure in the data, and we hope this will encourage future work.

5. Conclusion

In this paper, we have presented a deep-structured model using a fully-connected temporal CRF that not only models semantic aspects of activities but also reasons about long-term temporal relations. We also presented an asynchronous stochastic inference algorithm that circumvents a key bottleneck in the large-scale end-to-end model learning. Using our proposed method, we have demonstrated impressive activity classification and temporal localization results on a challenging dataset of realistic activities.

Acknowledgements: This work was partly supported by ONR MURI N00014-16-1-2007, ONR N00014-13-1-0720, NSF IIS-1338054, NSF-1652052, NRI-1637479, Intel via the Intel Science and Technology Center for Visual Cloud Systems, Allen Distinguished Investigator Award, gifts from Google, and the Allen Institute for Artificial Intelligence. The authors would like to thank Mark Yatskar for lending his expertise on deep CRFs, and Olga Russakovsky, Christoph Dann, and the anonymous reviewers for their invaluable suggestions and advice.

References

[1] L.-C. Chen*, A. G. Schwing*, A. L. Yuille, and R. Urtasun. Learning Deep Structured Models. In *Proc. ICML*, 2015. * equal contribution. 1

[2] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2

[3] C. R. de Souza, A. Gaidon, E. Vig, and A. M. Lpez. Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition. In *ECCV*, 2016. 2

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[5] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *ICCV*, 2013. 2

[6] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 2

[7] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 1, 2

[8] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`, 2015. 2

[9] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009. 2

[10] S. Gupta and J. Malik. Visual semantic role labeling. *CoRR*, /abs/1505.04474, 2015. 2

[11] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 5

[12] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. *ECCV*, 2012. 2

[13] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013. 2

[14] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013. 2

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[16] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 2

[17] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[18] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 2

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 7

[20] T. Lan, Y. Zhu, A. R. Zamir, and S. Savarese. Action recognition by hierarchical mid-level action elements. In *ICCV*, 2015. 2

[21] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, 2015. 2

[22] I. Laptev. On space-time interest points. *IJCV*, 64, 2005. 2

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[24] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 2

[25] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007. 2

[26] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshops*, 2009. 2

[27] P. Mettes, J. C. van Gemert, and C. G. Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, 2016. 1

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[29] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 2

[30] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 2

[31] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the why in images. *arXiv preprint arXiv:1406.5472*, 2014. 2

[32] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 2

[33] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(04):515–526, 1978. 2

[34] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *TPAMI*, 2012. 2

[35] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele. Script data for attribute-based recognition of composite activities. *ECCV*, 2012. 2

[36] M. S. Ryoo and J. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *CVPR*, 2007. 2

[37] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 2

[38] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, 2009. 5

[39] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 2

[40] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[41] G. A. Sigurdsson, X. Chen, and A. Gupta. Learning visual storylines with skipping recurrent neural networks. In *ECCV*, 2016. 2

[42] 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, 2, 5, 6, 7

[43] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, /abs/1312.6034, 2013. 1, 7

[44] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 5, 6

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 5, 6

[46] Y. Song, L.-P. Morency, and R. Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, 2013. 2

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

[48] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR*, /abs/1502.04681, 2015. 2

[49] C. Sun and R. Nevatia. Active: Activity concept transitions in video event classification. *ICCV*, 2013. 2

[50] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015. 2

[51] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 2

[52] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 2

[53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2, 7

[54] C. Vondrick, D. Oktay, H. Pirsiavash, and A. Torralba. Predicting motivations of actions by leveraging text. In *CVPR*, 2016. 2

[55] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 2

[56] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 2, 7

[57] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015. 2

[58] X. Wang, A. Farhadi, and A. Gupta. Actions ˜ transformations. In *CVPR*, 2016. 2

[59] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011. 2

[60] P. Weinzaepfel, X. Martin, and C. Schmid. Towards weakly-supervised action localization. *arXiv preprint arXiv:1605.05197*, 2016. 1, 5

[61] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *CVPR*, 2015. 2

[62] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. *CVPR*, 2016. 1, 2

[63] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015. 1, 5

[64] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015. 2

[65] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 2, 6, 7

[66] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation with deep densely connected mrfs. In *CVPR*, 2016. 1, 2