Can Humans Fly? Action Understanding with Multiple Classes of Actors

Chenliang Xu[1], Shao-Hang Hsieh[1], Caiming Xiong[2] and Jason J. Corso[1]

[1] Electrical Engineering and Computer Science, University of Michigan, Ann Arbor
[2] Statistics, University of California, Los Angeles
{cliangxu,shaohang,jjcorso}@umich.edu caimingxiong@ucla.edu

Abstract

Can humans fly? Emphatically no. Can cars eat? Again, absolutely not. Yet, these absurd inferences result from the current disregard for particular types of actors in action understanding. There is no work we know of on simultaneously inferring actors and actions in the video, not to mention a dataset to experiment with. Our paper hence marks the first effort in the computer vision community to jointly consider various types of actors undergoing various actions. To start with the problem, we collect a dataset of 3782 videos from YouTube and label both pixel-level actors and actions in each video. We formulate the general actor-action understanding problem and instantiate it at various granularities: both video-level single- and multiple-label actor-action recognition and pixel-level actor-action semantic segmentation. Our experiments demonstrate that inference jointly over actors and actions outperforms inference independently over them, and hence concludes our argument of the value of explicit consideration of various actors in comprehensive action understanding.

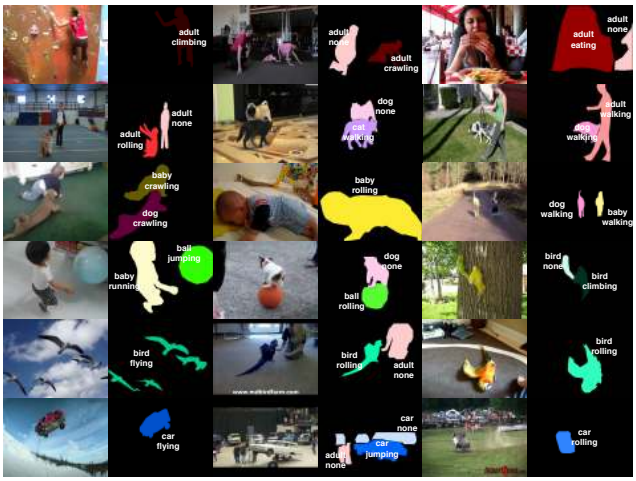

Figure 1. Montage of labeled videos in our new actor-action dataset, A2D. Examples of single actor-action instances as well as multiple actors doing different actions are present in this montage. Label colors are picked from the HSV color space, so that the same objects have the same *hue* (refer to Fig. 2 for the color-legend). Black is the background. **View zoomed and in color.**

1. Introduction

Like verbs in language, action is the heart of video understanding. As such, it has received a significant amount of attention in the last decade. Our community has moved from small datasets of a handful of actions [12, 50] to large datasets with many dozens of actions [27, 45]; from constrained domains like sporting [42, 46] to videos in-the-wild [38, 45]. Notable methods have demonstrated that low-level features [25, 33, 58, 59], mid-level atoms [67], high-level exemplars [48], structured models [42, 56], and attributes [37] can be used for action recognition. Impressive methods have even pushed toward action recognition for multiple views [40], event recognition [20], group-based activities [32], and even human-object interactions [15, 44].

However, these many works emphasize a small subset of the broader action understanding problem. First, aside from Iwashita et al. [19] who study egocentric animal activities, these existing methods all assume the agent of the action, which we call the *actor*, is a human adult. The only work we are aware of that jointly considers different types of actors and actions is Xu et al. [61], but this work uses a dataset two-orders of magnitude smaller (32 videos versus 3782 videos), groups all animals together into one class separate from humans, and is primarily a visual-psychophysical study using off-the-shelf vision methods.

Although *looking at people* is certainly a relevant application domain for computer vision, it is not the only one; consider recent advances in video-to-text [1, 14] that can be used for semantic indexing of large video databases [36], or advances in autonomous vehicles [11]. In these applications, understanding both the actor and the action are critical for success: e.g., the autonomous vehicle needs to distinguish between a child, a deer and a squirrel running into the

road so it can accurately make an avoidance plan. Applications like these, e.g., robotic autonomy [55], are abundant and growing.

Second, these works largely focus on *action recognition*, which is posed as the classification of a pre-temporally trimmed clip into one of k action classes from a closed-world. The direct utility of results based on this problem formulation is limited. The community has indeed begun to move beyond this simplified problem into action detection [56, 65], action localization [22, 39], action segmentation [23, 24], and actionness ranking [8]. But, all of these works do so strictly in the context of human actors.

In this paper, we overcome both of these narrow viewpoints and introduce a new level of generality to the action understanding problem by considering multiple different classes of actors undergoing multiple different classes of actions. To be exact, we consider seven actor classes (*adult*, *baby*, *ball*, *bird*, *car*, *cat*, and *dog*) and eight action classes (*climb*, *crawl*, *eat*, *fly*, *jump*, *roll*, *run*, and *walk*) not including the no-action class, which we also consider. We formulate a general actor-action understanding framework and implement it for three specific problems: actor-action recognition with single- and multiple-label, and actor-action semantic segmentation. These three problems cover different levels of modeling and hence allow us to analyze the new problem thoroughly. We further distinguish our work from multi-task learning [5] that focuses on getting a shared representation for training better classifiers, whereas we focus on modeling the relationship and interactions of the actor and action under a unified graphical model.

To support these new actor-action understanding problems, we have created a new dataset, which we call the Actor-Action Dataset or A2D (see Fig. 1), that is labeled at the pixel-level for actors and actions (densely in space over actors, sparsely in time). The A2D has 3782 videos with at least 99 instances per valid actor-action tuple (Sec. 3 and Fig. 2 have exact statistics). We thoroughly analyze empirical performance of both state-of-the-art and baseline methods, including naïve Bayes (independent over actor and action), a joint product-space model (each actor-action pair is considered as one class), and a bilayer graphical model inspired by [31] that connects actor nodes with action nodes.

Our experiments demonstrate that inference jointly over actors and actions outperforms inference independently over them, and hence, supports the explicit consideration of various actors in comprehensive action understanding. In other words, although a *bird* and an *adult* can both *eat*, the space-time appearance of a *bird eating* and an *adult eating* are different in significant ways. Furthermore, the various mannerisms of the way *birds eat* and *adults eat* mutually reinforces inference over the constituent parts. This result is analogous to Sadeghi and Farhadi's visual phrases work [49] in which it is demonstrated that joint detection over small groups of objects in images is more robust than separate detection over each object followed a merging process and to Gupta et al.'s [15] work on human object-interactions in which considering specific objects while modeling human actions leads to better inferences for both parts.

Our paper marks the first effort in the computer vision community to jointly consider various types of actors undergoing various actions. As such, we pose two goals: first, we seek to formulate the general actor-action understanding problem and instantiate it at various granularities, and second, we seek to assess whether or not it is beneficial to explicitly jointly consider actors and actions in this new problem-space. The paper describes the new A2D dataset (Sec. 3), the actor-action problem formulation (Sec. 4) and our experiments to answer this question (Sec. 5).

2. Related Work

A related work from the action recognition community is the recent Bojanowski et al. [2] paper, which focuses on finding different human *actors* in movies, but these are the actor-names and not different types of actors, like *dog* and *cat* as we consider in this paper. Similarly, the existing work on actions and objects, such as [15, 44], is strictly focused on interaction between human actors manipulating various objects and not different types of actors, which is our focus.

The remainder of the related work section discusses segmentation, which is a major emphasis of our broader view of the action understanding problem-space and yet was not discussed in the introduction (Sec. 1). Semantic segmentation methods can now densely label more than a dozen classes in images [13, 29, 30, 41] and videos [21, 57] undergoing rapid motion; semantic segmentation methods have even been unified with object detectors and scene classification [63], extended to 3D [18, 28, 53] and posed jointly with attributes [66], stereo [9, 31, 51] and SFM [4, 10]. Although the underlying optimization problems in these methods tend to be expensive, average-per-class accuracy scores has significantly increased, for example, from 67% in [52] to nearly 80% in [26, 29, 63] on the popular MSRC semantic segmentation benchmark. Further works have moved beyond full supervision to weakly supervised object discovery and learning [16, 54].

Other related works include unsupervised video object segmentation [34, 35, 43, 64] and joint temporal segmentation with action recognition [17]. These video object segmentation methods are class-independent and assume a single dominant object (actor) in the video; they are hence not directly comparable to our work although one can foresee a potential method using video object segmentation as a precursor to the actor-action understanding problem.

There is a clear trend moving toward video semantic segmentation and toward weak supervision. But, these exist-

	climb	crawl	eat	fly	jump	roll	run	walk	none
adult	101	105	105		174	105	175	282	761
baby	104	106				107		113	36
ball				109	105	117			87
bird	99		105	106	102	107		112	26
car				102	107	104	120		99
cat	106		110		105	103	99	113	53
dog		109	107		104	104	110	176	46

Figure 2. Statistics of label counts in the new A2D dataset. We show the number of videos in our dataset in which a given [actor, action] label occurs. Empty entries are joint-labels that are not in the dataset either because they are invalid (a *ball* cannot *eat*) or were in insufficient supply, such as for the case *dog-climb*. The background color in each cell depicts the color we use throughout the paper; we vary hue for actor and saturation for action.

ing works in semantic segmentation focus on labeling pixels/voxels as various objects or background-stuff classes. They do not consider the joint label-space of what actions these "objects" may be doing. Our work differs from them by directly considering this actor-action problem, while also building on the various advances made in these papers.

3. A2D—The Actor-Action Dataset

We have collected a new dataset consisting of 3782 videos from YouTube; these videos are hence unconstrained "in-the-wild" videos with varying characteristics. Figure 1 has single-frame examples of the videos. We select seven classes of actors performing eight different actions. Our choice of actors covers articulated ones, such as *adult*, *baby*, *bird*, *cat* and *dog*, as well as rigid ones, such as *ball* and *car*. The eight actions are *climbing*, *crawling*, *eating*, *flying*, *jumping*, *rolling*, *running*, and *walking*. A single action-class can be performed by various actors, but none of the actors can perform all eight actions. For example, we do not consider *adult-flying* or *ball-running* in the dataset. In some cases, we have pushed the semantics of the given action term to maintain a small set of actions: e.g., *car-running* means the car is moving and *ball-jumping* means the ball is bouncing. One additional action label *none* is added to account for actions other than the eight listed ones as well as actors in the background that are not performing an action. Therefore, we have in total 43 valid actor-action tuples.

To query the YouTube database, we use various text-searches generated from actor-action tuples. Resulting videos were then manually verified to contain an instance of the primary actor-action tuple, and subsequently temporally trimmed to contain that actor-action instance. The trimmed videos have an average length of 136 frames, with a minimum of 24 frames and a maximum of 332 frames. We split the dataset into 3036 training videos and 746 testing videos divided evenly over all actor-action tuples. Figure 2 shows

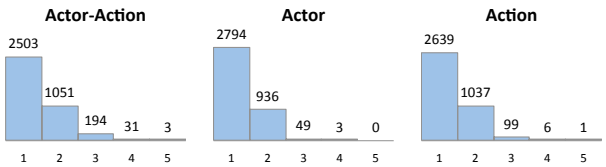

Figure 3. Histograms of counts of joint actor-actions, and individual actors and actions per video in A2D; roughly one-third of the videos have more than one actor and/or action.

the statistics for each actor-action tuple. One-third of the videos in A2D have more than one actor performing different actions, which further distinguishes our dataset from most action classification datasets. Figure 3 shows exact counts for these cases with multiple actors and actions.

To support the broader set of action understanding problems in consideration, we label three to five frames for each video in the dataset with both dense pixel-level actor and action annotations (Fig. 1 has labeling examples). The selected frames are evenly distributed over a video. We start by collecting crowd-sourced annotations from MTurk using the LabelMe toolbox [47], then we manually filter each video to ensure the labeling quality as well as the temporal coherence of labels. Video-level labels are computed directly from these pixel-level labels for the recognition tasks. To the best of our knowledge, this dataset is the first video dataset that contains both actor and action pixel-level labels.

4. Actor-Action Understanding Problems

Without loss of generality, let $\mathcal{V} = \{v_1, \ldots, v_n\}$ denote a video with n voxels in space-time lattice Λ^3 or n supervoxels in a video segmentation [7, 60, 62] represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the neighborhood structure of the graph is given by the supervoxel segmentation method; when necessary we write $\mathcal{E}(v)$ where $v \in \mathcal{V}$ to denote the subset of $\mathcal{V}$ that are neighbors with v. We use $\mathcal{X}$ to denote the set of actor labels: {*adult*, *baby*, *ball*, *bird*, *car*, *cat*, *dog*}, and we use $\mathcal{Y}$ to denote the set of action labels: {*climbing*, *crawling*, *eating*, *flying*, *jumping*, *rolling*, *running*, *walking*, *none*[1]}.

Consider a set of random variables $\mathbf{x}$ for actor and another $\mathbf{y}$ for action; the specific dimensionality of $\mathbf{x}$ and $\mathbf{y}$ will be defined later. Then, the general actor-action understanding problem is specified as a posterior maximization:

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\max_{\mathbf{x},\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\mathcal{V}) \ . \quad (1)$$

Specific instantiations of this optimization problem give rise to various actor-action understanding problems, which we specify next, and specific models for a given instantiation

[1]The *none* action means either there is no action present or the action is not one of those we have considered.

will vary the underlying relationship between $\mathbf{x}$ and $\mathbf{y}$ allowing us to deeply understand their interplay.

4.1. Single-Label Actor-Action Recognition

This is the coarsest level of granularity we consider in the paper and it instantiates the standard action recognition problem [33]. Here, $\mathbf{x}$ and $\mathbf{y}$ are simply scalars x and y, respectively, depicting the single actor and action label to be specified for a given video $\mathcal{V}$. We consider three models for this case:

Naïve Bayes: Assume independence across actions and actors, and then train a set of classifiers over actor space $\mathcal{X}$ and a separate set of classifiers over action space $\mathcal{Y}$. This is the simplest approach and is not able to enforce actor-action tuple existence: e.g., it may infer *adult-fly* for a test video.

Joint Product Space: Create a new label space $\mathcal{Z}$ that is the joint product space of actors and actions: $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Directly learn a classifier for each actor-action tuple in this joint product space. Clearly, this approach enforces actor-action tuple existence, and we expect it to be able to exploit cross-actor-action features to learn more discriminative classifiers. However, it may not be able to exploit the commonality across different actors or actions, such as the similar manner in which a *dog* and a *cat walk*.

Trilayer: The trilayer model unifies the naïve Bayes and the joint product space models. It learns classifiers over the actor space $\mathcal{X}$, the action space $\mathcal{Y}$ and the joint actor-action space $\mathcal{Z}$. During inference, it separately infers the naïve Bayes terms and the joint product space terms and then takes a linear combination of them to yield the final score. It models not only the cross-actor-action but also the common characteristics among the same actor performing different actions as well as the different actors performing the same action.

In all cases, we extract local features (see Sec. 5.1 for details) and train a set of one-vs-all classifiers, as is standard in contemporary action recognition methods, and although not strictly probabilistic, can be interpreted as such to implement Eq. 1.

4.2. Multi-Label Actor-Action Recognition

As noted in Fig. 3, about one-third of the videos in A2D have more than one actor and/or action present in a given video. In many realistic video understanding applications, we find such multiple-label cases. We address this explicitly by instantiating Eq. 1 for the multi-label case. Here, $\mathbf{x}$ and $\mathbf{y}$ are binary vectors of dimension $|\mathcal{X}|$ and $|\mathcal{Y}|$ respectively. x_i takes value 1 if the ith actor-type is present in the video and zero otherwise. We define $\mathbf{y}$ similarly. This general definition, which does not tie specific elements of $\mathbf{x}$ to those in $\mathbf{y}$, is necessary to allow us to compare independent multi-label performance over actors and actions with that of the actor-action tuples. We again consider a naïve Bayes pair of multi-label actor and action classifiers, multi-label actor-action classifiers over the joint product space, as well as the trilayer model that unifies the above classifiers.

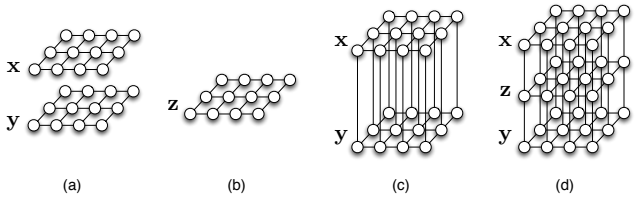

Figure 4. Visualization of different graphical models to solve Eq. 1. The figure here is for simple illustration and the actual voxel or supervoxel graph is built for a video volume.

4.3. Actor-Action Semantic Segmentation

Semantic segmentation is the most fine-grained instantiation of actor-action understanding that we consider, and it subsumes other coarser problems like detection and localization, which we do not consider in this paper for space. Here, we seek a label for actor and action per-voxel over the entire video. Define the two sets of random variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ and $\mathbf{y} = \{y_1, \ldots, y_n\}$ to have dimensionality in the number of voxels or supervoxels, and assign each $x_i \in \mathcal{X}$ and each $y_i \in \mathcal{Y}$. The objective function in Eq. 1 remains the same, but the way we define the graphical model implementing $P(\mathbf{x}, \mathbf{y}|\mathcal{V})$ leads to acutely different assumptions on the relationship between actor and action variables.

We explore this relationship in the remainder of this section. We start by again introducing a naïve Bayes-based model that treats the two classes of labels separately, and a joint product space model that considers actors and actions together in a tuple $[\mathbf{x}, \mathbf{y}]$. We then explore a bilayer model, inspired by Ladický et al. [31], that considers the inter-set relationship between actor and action variables. Finally, we introduce a new trilayer model that considers both intra- and inter-set relationships. Figure 4 illustrates these various graphical models. We then evaluate the performance of all models in terms of joint actor and action labeling in Sec. 5.

Naïve Bayes-based Model First, let us consider a naïve Bayes-based model, similar to the one used for actor-action recognition earlier:

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}|\mathcal{V}) &= P(\mathbf{x}|\mathcal{V})P(\mathbf{y}|\mathcal{V}) \\ &= \prod_{i \in \mathcal{V}} P(x_i)P(y_i) \prod_{i \in \mathcal{V}} \prod_{j \in \mathcal{E}(i)} P(x_i, x_j)P(y_i, y_j) \\ &\propto \prod_{i \in \mathcal{V}} \phi_i(x_i)\psi_i(y_i) \prod_{i \in \mathcal{V}} \prod_{j \in \mathcal{E}(i)} \phi_{ij}(x_i, x_j)\psi_{ij}(y_i, y_j) \end{aligned} \tag{2}$$

where ϕ_i and ψ_i encode the separate potential functions defined on actor and action nodes alone, respectively, and ϕ_{ij}

and ψ_{ij} are the pairwise potential functions within sets of actor nodes and sets of action nodes, respectively.

We train classifiers $\{f_c|c \in \mathcal{X}\}$ over actors and $\{g_c|c \in \mathcal{Y}\}$ on sets of actions using features described in Sec. 5.3, and ϕ_i and ψ_i are the classification scores for supervoxel i. The pairwise edge potentials have the form of a contrast-sensitive Potts model [3]:

$$\phi_{ij} = \begin{cases} 1 & \text{if } x_i = x_j \\ \exp(-\theta/(1+\chi^2_{ij})) & \text{otherwise,} \end{cases} \quad (3)$$

where χ^2_{ij} is the χ^2 distance between feature histograms of nodes i and j, θ is a parameter to be learned from the training data. ψ_{ij} is defined analogously. Actor-action semantic segmentation is obtained by solving these two *flat* CRFs independently.

Joint Product Space We consider a new set of random variables $\mathbf{z} = \{z_1, \ldots, z_n\}$ defined again on all supervoxels in a video and take labels from the actor-action product space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. This formulation jointly captures the actor-action tuples as unique entities but cannot model the common actor and action behaviors among different tuples as later models below do; we hence have a single-layer graphical model:

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}|\mathcal{V}) \doteq P(\mathbf{z}|\mathcal{V}) &= \prod_{i\in\mathcal{V}} P(\mathbf{z}_i) \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} P(z_i, z_j) \\ &\propto \prod_{i\in\mathcal{V}} \varphi_i(z_i) \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} \varphi_{ij}(z_i, z_j) \\ &= \prod_{i\in\mathcal{V}} \varphi_i([x_i, y_i]) \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} \varphi_{ij}([x_i, y_i], [x_j, y_j]) \ , \end{aligned} \quad (4)$$

where φ_i is the potential function for joint actor-action product space label, and φ_{ij} is the inter-node potential function between nodes with the tuple $[\mathbf{x}, \mathbf{y}]$. To be specific, φ_i contains the classification scores on the node i from running trained actor-action classifiers $\{h_c|c \in \mathcal{Z}\}$, and φ_{ij} has the same form as Eq. 3. Fig. 4 (b) illustrates this model as a one layer CRF defined on the actor-action product space.

Bilayer Model Given the actor nodes $\mathbf{x}$ and action nodes $\mathbf{y}$, the bilayer model connects each pair of random variables $\{(x_i, y_i)\}_{i=1}^n$ with an edge that encodes the potential function for the tuple $[x_i, y_i]$, directly capturing the *covariance* across the actor and action labels. We have

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}|\mathcal{V}) &= \prod_{i\in\mathcal{V}} P(x_i, y_i) \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} P(x_i, x_j) P(y_i, y_j) \\ &\propto \prod_{i\in\mathcal{V}} \phi_i(x_i)\psi_i(y_i)\xi_i(x_i, y_i) \cdot \\ &\quad \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} \phi_{ij}(x_i, x_j)\psi_{ij}(y_i, y_j) \ , \end{aligned} \quad (5)$$

where $\phi_\cdot$ and $\psi_\cdot$ are defined as earlier, $\xi_i(x_i, y_i)$ is a learned potential function over the product space of labels, which can be exactly the same as φ_i in Eq. 4 above or a compatibility term like the contrast sensitive Potts model, Eq. 3 above. We choose the former in this paper. Fig. 4 (c) illustrates this model. We note that additional links can be constructed by connecting corresponding edges between neighboring nodes across layers and encoding the occurrence among the bilayer edges, such as the joint object class segmentation and dense stereo reconstruction model in Ladický et al. [31]. However, their model is not directly suitable here.

Trilayer Model So far we have introduced three baseline formulations in Eq. 1 for semantic actor-action segmentation that relate the actor and action terms in different ways. The naïve Bayes model (Eq. 2) does not consider any relationship between actor $\mathbf{x}$ and action $\mathbf{y}$ variables. The joint product space model (Eq. 4) combines features across actors and actions as well as inter-node interactions in the neighborhood of an actor-action node. The bilayer model (Eq. 5) adds actor-action interactions among separate actor and action nodes, but it does not consider how these interactions vary spatiotemporally.

Therefore, we introduce a new trilayer model that explicitly models such variations (see Fig. 4d) by combining nodes $\mathbf{x}$ and $\mathbf{y}$ with the joint product space nodes $\mathbf{z}$:

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V}) &= P(\mathbf{x}|\mathcal{V})P(\mathbf{y}|\mathcal{V})P(\mathbf{z}|\mathcal{V}) \prod_{i\in\mathcal{V}} P(x_i, z_i)P(y_i, z_i) \\ &\propto \prod_{i\in\mathcal{V}} \phi_i(x_i)\psi_i(y_i)\varphi_i(z_i)\mu_i(x_i, z_i)\nu_i(y_i, z_i) \cdot \\ &\quad \prod_{i\in\mathcal{V}} \prod_{j\in\mathcal{E}(i)} \phi_{ij}(x_i, x_j)\psi_{ij}(y_i, y_j)\varphi_{ij}(z_i, z_j) \ , \end{aligned} \quad (6)$$

where we define

$$\mu_i(x_i, z_i) = \begin{cases} w(y_i'|x_i) & \text{if } x_i = x_i' \text{ for } z_i = [x_i', y_i'] \\ 0 & \text{otherwise} \end{cases}$$

$$\nu_i(y_i, z_i) = \begin{cases} w(x_i'|y_i) & \text{if } y_i = y_i' \text{ for } z_i = [x_i', y_i'] \\ 0 & \text{otherwise} \end{cases} . \quad (7)$$

Terms $w(y_i'|x_i)$ and $w(x_i'|y_i)$ are classification scores of conditional classifiers, which are explicitly trained for this trilayer model. These conditional classifiers are the main reason for the increased performance found in this method: separate classifiers for the same action conditioned on the type of actor are able to exploit the characteristics unique to that actor-action tuple. For example, when we train a conditional classifier for action *eating* given actor *adult*, we use all other actions performed by *adult* as negative training samples. Therefore our trilayer model considers all relationships in the individual actor and action spaces as well as

	Single-Label			Multiple-Label		
	Classification Accuracy			Mean Average Precision		
Model	Actor	Action	<A, A>	Actor	Action	<A, A>
Naïve Bayes	70.51	74.40	56.17	76.85	78.29	60.13
JointPS	72.25	72.65	61.66	76.81	76.75	63.87
Trilayer	**75.47**	**75.74**	**64.88**	**78.42**	**79.27**	**66.86**

Table 1. Single-label and multiple-label actor-action recognition in the three settings: independent actor and action models (naïve Bayes), joint actor-action models in a product-space and the trilayer model. The scores are not comparable along the columns (e.g., the space of independent actors and actions is significantly smaller than that or actor-action tuples); the point of comparison is along the rows where we find the joint model to outperform the independent models when considering both actors and actions. <A, A> denotes evaluating in the joint actor-action product-space.

the joint product space. In other words, the previous three baseline models are all special cases of the trilayer model. It can be shown that the solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ maximizing Eq. 6 also maximizes Eq. 1 (see Appendix).

5. Experiments

We thoroughly study each of the instantiations of the actor-action understanding problem with the overarching goal of assessing if the joint modeling of actor and action improves performance over modeling each of them independently, despite the large space. We follow the training and testing splits discussed in Sec. 3; for assigning a single-label to a video for the single-label actor-action recognition, we choose the label associated with the query for which we searched and selected that video from YouTube.

5.1. Single-Label Actor-Action Recognition

Following the typical action recognition setup, e.g., [33], we use the state-of-the-art dense trajectory features (trajectories, HoG, HoF, MBHx and MBHy) [58] and train a set of 1-versus-all SVM models (with RBF-χ^2 kernels from LIBSVM [6]) for the label sets of actors, actions and joint actor-action labels. Specifically, when training the *eating* classifier, the other seven actions are negative examples; when we train the *bird-eating* classifier, we use the 35 other actor-action labels as negative examples.

Table 1-left shows the classification accuracy of the naïve Bayes, joint product space and trilayer models, in terms of classifying actor, action and actor-action labels. To evaluate the joint actor-action (the <A, A> columns) for the naïve Bayes models, we train the actor and action classifiers independently, apply them to the test videos independently and then score them together (i.e., a video is correct if and only if actor and action are correct). We observe that the independent model for action outperforms the joint product space model for action; this can be explained by the regularity across different actors for the same action that can be exploited in the naïve Bayes model, but that results in more inter-class overlap in the joint product space. For example, a *cat-running* and a *dog-running* have similar signatures in space-time: the naïve Bayes model does not need to distinguish between these two, but the joint product space does. However, we find that when we consider both the actor and action in evaluation, it is clearly beneficial to jointly model them. This phenomenon occurs in all of our experiments. Finally, the trilayer model outperforms the other two models in terms of both individual actor or action tasks as well as the joint actor-action task. The reason is that the trilayer model incorporates both types of relationships that are separately modeled in the naïve Bayes and joint product space models.

5.2. Multiple-Label Actor-Action Recognition

For the multiple-label case, we use the same dense trajectory features as in Sec. 5.1, and we train 1-versus-all SVM models again for the label sets of actor, action and actor-action pairs, but with different training regimen to capture the multiple-label setting. For example, when training the *adult* classifier, we use all videos containing any actor *adult* as positive examples no matter the other actors that coexist in the positive videos, and we use the rest of videos as negative examples. For evaluation, we adapt the approach from HOHA2 [40]. We treat multiple-label actor-action recognition as a retrieval problem and compute mean average precision (mAP) given the classifier scores. Table 1-right shows the performance of the three methods on this task. Again, we observe that the joint product space has higher mAP than naïve Bayes for the joint actor-action evaluation. We also observe the trilayer model further improves the scores following the same trend as in the single-label case.

However, we also note that large improvement in the both individual tasks from the trilayer model. This implies that the "side" information of the actor when doing action recognition (and vice versa) provides useful information to improve the inference task, thereby answering the core question in the paper.

5.3. Actor-Action Semantic Segmentation

State-of-the-Art Pixel-Based Segmentation. We first apply the state-of-the-art robust P^N model [29] at the pixel level; we apply their supplied code off-the-shelf as a baseline. The average-per-class performance is 13.74% for the joint actor-action task, 47.2% for actor and 34.49% for action. We suspect that the modeling at pixel and superpixel level can not well capture the motion changes of actions, which explains why the actor score is high but the other scores are comparatively lower. The P^N model could be generalized to fit within our framework, which we leave for future work. We use supervoxel segmentation and extract spatiotemporal features for assessing the various models posed for actor-action semantic segmentation.

		bird							cat							dog						
Model	BK	climb	eat	fly	jump	roll	walk	none	climb	eat	jump	roll	run	walk	none	crawl	eat	jump	roll	run	walk	none
Naïve Bayes	79.5	21.0	6.2	28.7	17.3	28.3	2.8	**29.3**	28.2	24.3	1.6	38.2	43.6	1.0	**4.4**	6.1	13.2	**5.3**	21.9	**35.9**	25.8	**4.3**
JointPS	75.1	23.0	15.5	36.0	19.2	26.6	7.5	0.0	19.4	24.6	4.1	32.4	28.5	7.5	0.5	**10.9**	24.2	2.1	21.1	21.2	38.2	0.0
Conditional	79.5	23.2	8.4	40.7	**25.4**	30.5	7.5	0.0	26.0	30.5	8.0	31.7	**53.3**	**9.1**	0.0	7.4	16.2	3.1	24.6	29.3	**53.6**	0.0
Bilayer	**79.7**	24.5	13.3	40.8	13.0	35.4	7.0	0.0	32.7	**32.9**	1.1	38.0	37.0	7.5	0.1	2.5	22.8	2.4	**35.9**	27.0	29.6	0.0
Trilayer	78.5	**28.1**	**18.2**	**55.3**	20.3	**42.5**	**9.0**	0.0	**33.1**	27.2	**6.1**	**49.8**	48.5	6.6	0.0	9.9	**31.0**	2.0	27.6	23.6	39.4	0.0

		adult							baby					ball				car				
Model	climb	crawl	eat	jump	roll	run	walk	none	climb	crawl	roll	walk	none	fly	jump	roll	none	fly	jump	roll	run	none
Naïve Bayes	21.5	30.4	21.5	11.3	5.0	18.1	11.5	**25.8**	21.6	23.5	20.5	8.6	**7.4**	2.9	13.6	6.6	**8.6**	10.0	71.2	22.2	5.5	**13.7**
JointPS	23.1	59.3	44.0	17.5	17.6	34.6	28.4	21.4	18.3	24.0	28.1	17.2	0.6	0.0	6.5	4.7	2.8	13.2	74.7	43.9	30.5	8.1
Conditional	18.5	43.1	36.3	**25.4**	17.4	31.8	30.7	12.1	**26.5**	20.4	36.7	13.9	5.6	**3.7**	**16.2**	**21.4**	9.0	**27.7**	**77.6**	43.5	37.2	1.7
Bilayer	27.2	49.6	**51.6**	25.1	**28.4**	27.9	**39.2**	0.6	13.2	**25.4**	**44.0**	24.0	0.0	0.3	10.3	6.0	0.0	20.9	76.8	37.2	39.6	0.5
Trilayer	**33.1**	**59.8**	49.8	19.9	27.6	**40.2**	31.7	24.6	20.4	21.7	39.3	**25.3**	0.0	1.0	11.9	6.1	0.0	24.4	75.9	**44.3**	**48.3**	2.4

		Average Per Class Accuracy		
	Model	**Actor**	**Action**	**<A, A>**
Unary Term Only	Naïve Bayes	43.02	40.08	16.35
	JointPS	40.89	38.50	20.61
	Conditional	43.02	41.19	22.55
	Bilayer	43.02	40.08	16.35
	Trilayer	43.08	41.61	22.59
		Average Per Class Accuracy		
	Model	**Actor**	**Action**	**<A, A>**
Full Model	Naïve Bayes	44.78	42.59	19.28
	JointPS	41.96	40.09	21.73
	Conditional	44.78	41.88	24.19
	Bilayer	44.46	43.62	23.43
	Trilayer	**45.70**	**46.96**	**26.46**

Table 2. Average per-class semantic segmentation accuracy in percentage of joint actor-action labels for all models (for individual classes, left, and in summary, right). The leading scores of each label are displayed in bold font. The summary scores on the right and indicate that the trilayer model, which considers the action and actor models alone as well as the actor-action product-space, performs best.

Supervoxel Segmentation and Features. We use TSP [7] to obtain supervoxel segmentations due to its strong performance on the supervoxel benchmark [60]. In our experiments, we set $k = 400$ yielding about 400 supervoxels touching each frame. We compute histograms of textons and dense SIFT descriptors over each supervoxel volume, dilated by 10 pixels. We also compute color histograms in both RGB and HSV color spaces and dense optical flow histograms. We extract feature histograms from the entire supervoxel 3D volume, rather than a single representative superpixel [57]. Furthermore, we inject the dense trajectory features [58] to supervoxels by assigning each trajectory to the supervoxels it intersects in the video.

Frames in A2D are sparsely labeled; to obtain a supervoxel's groundtruth label, we look at all labeled frames in a video and take a majority vote over intersecting labeled pixels. We train sets of 1-versus-all SVM classifiers (linear kernels) for actor, action, and actor-action as well as conditional classifiers separately. The parameters of the graphical model are tuned by empirical search, and loopy belief propagation is used for inference. The inference output is a dense labeling of video voxels in space-time, but, as our dataset is sparsely labeled in time, we compute the average per-class segmentation accuracy only against those frames for which we have groundtruth labels. We choose average per-class accuracy over global accuracy because our goal is to compares actor and action rather than full video labeling.

Evaluation. Table 2–right shows the overall performance of the different methods. The upper part is results with only the unary terms and the lower part is the full model performance. We not only evaluate the actor-action pairs but also individual actor and action tasks. The conditional model is a variation of bilayer model with different aggregation—we infer the actor label first then the action label conditioned on the actor. Note that the bilayer model has the same unary scores as the naïve Bayes model (using actor ϕ_i and action ψ_i outputs independently) and the actor unary of the conditional model is the same as that of the naïve Bayes model (followed by the conditional classifier for action).

Over all models, the naïve Bayes model performs worst, which is expected as it does not encode any interactions between the two label sets. We observe that the conditional model has better action unary and actor-action scores, which indicates that knowing actors can help with action inference. We also observe that the bilayer model has a poor unary performance of 16.35% (actor-action) that is the same as naïve Bayes but for the full model it improves dramatically to 23.43%, which suggests that the performance boost again comes from the interaction of actor and action nodes in the full bilayer model. We also observe that the full trilayer model has not only much better performance in the joint actor-action task, but also better scores for actor and action individual tasks in the full model, as it is the only model considered that incorporates classifiers in both individual actor and action tasks and also in the joint space.

Table 2–left shows the comparison of quantitative performance for specific actors and actions. We observe that the trilayer model has leading scores for more actor-action tuples than the other models. The trilayer model has significant improvement on labels such as *bird-flying*, *adult-running* and *cat-rolling*. We note the systematic increase in performance as more complex actor-action variable interactions are included. We also note that the tuples with *none* action are sampled with greater variation than the action classes (Fig. 2), which contributes to the poor performance of *none* over all actors. Interestingly, the naïve Bayes model has relatively better performance on the *none* action classes. We suspect that the label-variation for *none* leads to high-entropy over its classifier density and hence when joint modeling, the actor inference pushes the action variable away from the *none* action class.

Fig. 5 shows example segmentations. Recall that the naïve Bayes model considers the actor and action labeling problem independent of each other. Therefore, the *baby-rolling* in the second video get assigned with actor label *dog* and action label *rolling* when there is no consideration between actor and action. The bilayer model partially recovers the *baby* label, whereas the trilayer model successfully recovers the *baby-rolling* label, due to the modeling of inter-node relationship in the joint actor-action space of the tri-

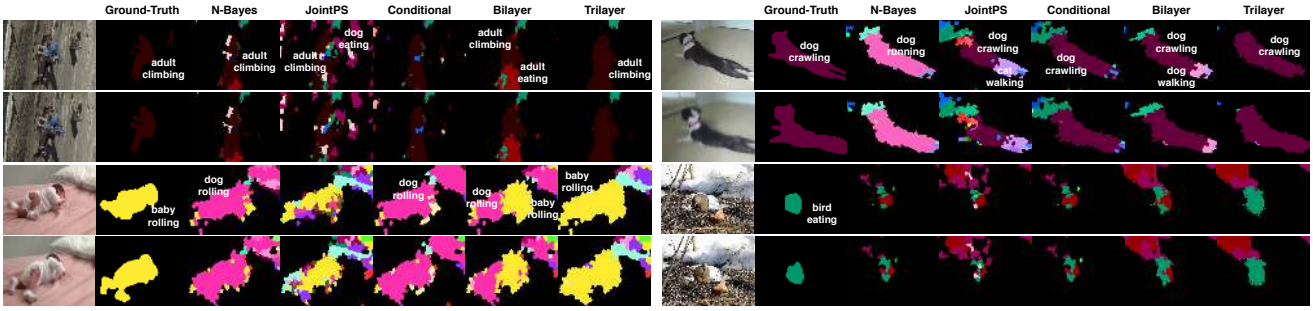

Figure 5. Comparative example of semantic segmentation results. These sample only two frames from the each dense video outputs.

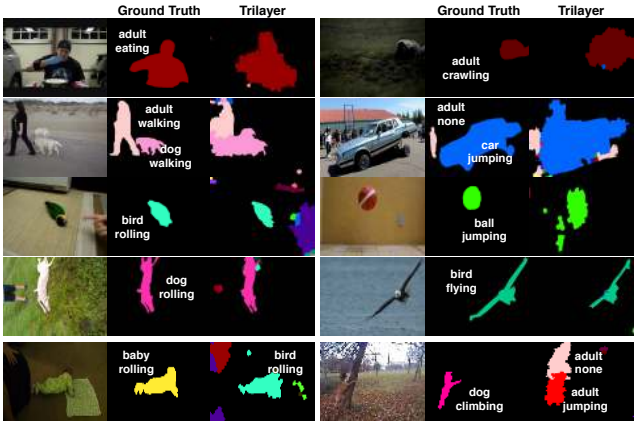

Figure 6. Example results from the trilayer model (upper are good, lower are failure cases).

layer model. We also visualize more example outputs of the trilayer model in Fig. 6. Note that the fragmented segmentation in the *ball* video is due to poor supervoxel segmentation algorithm in the pre-processing step. We also show trilayer failure cases in the bottom row of Fig. 6, which are due to weak cross-class visual evidence.

6. Discussion and Contributions

Our thorough assessment of all instantiations of the actor-action understanding problem at both coarse video-recognition level and fine semantic segmentation level provides strong evidence that the joint modeling of actor and action improves performance over modeling each of them independently. We find that for both individual actor and action understanding and joint actor-action understanding, it is beneficial to jointly consider actor and action. A proper modeling of the interactions between actor and action results in dramatic improvement over the baseline models of the naïve Bayes and joint product space models, as we observe from the bilayer and trilayer models.

Our paper set out with two goals: first, we sought to motivate and develop a new, more challenging, and more relevant actor-action understanding problem, and second, we sought to assess whether joint modeling of actors and actions improved performance for this new problem. We achieved these goals through the three contributions:

1. New actor-action understanding problem and dataset.
2. Thorough evaluation of actor-action recognition and semantic segmentation problems using state-of-the-art features and models. The experiments unilaterally demonstrate a benefit for jointly modeling actors and actions.
3. A new trilayer approach to recognition and semantic segmentation that combines both the independent actor and action variations and product-space interactions.

Our full dataset, computed features, codebase, and evaluation regimen are released[2] to support further inquiry into this new and important problem in video understanding.

Appendix

We show that a solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ maximizing Eq. 6 also maximizes Eq. 1. First, to simplify Eq. 6, we set $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$. Therefore we can obtain:

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}, [\mathbf{x}, \mathbf{y}]|\mathcal{V}) &= P'(\mathbf{x}, \mathbf{y}|\mathcal{V}) \\ &= \frac{1}{Z} \prod_{i \in \mathcal{V}} \phi_i(x_i)\psi_i(y_i)\varphi_i(z_i)w(y_i|x_i)w(x_i|y_i) \cdot \\ &\quad \prod_{i \in \mathcal{V}} \prod_{j \in \mathcal{E}(i)} \phi_{ij}(x_i, x_j)\psi_{ij}(y_i, y_j)\varphi_{ij}(z_i, z_j) \ . \end{aligned} \tag{8}$$

Theorem 1. *Let* $(\mathbf{x}^*, \mathbf{y}^*) = \arg\max_{\mathbf{x},\mathbf{y}} P'(\mathbf{x}, \mathbf{y}|\mathcal{V})$ *be the optimal solution of Eq. 8, then* $(\mathbf{x}^*, \mathbf{y}^*, [\mathbf{x}^*, \mathbf{y}^*]) = \arg\max_{\mathbf{x},\mathbf{y},\mathbf{z}} P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V})$ *are optimal results.*

Proof. First, by construction, when $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$ then $P^{'}(\mathbf{x}, \mathbf{y}|\mathcal{V}) = P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V})$. The rest of the proof follows in two parts:

- Assume that $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$ in the optimal solution of Eq. 6. Then: $\arg\max P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V}) = \arg\max_{\mathbf{z}=[\mathbf{x},\mathbf{y}]} P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V}) = \arg\max P^{'}(\mathbf{x}, \mathbf{y}|\mathcal{V})$.
- Assume that in the optimal solution $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ that $\mathbf{z} = [\mathbf{x}', \mathbf{y}'] \neq [\mathbf{x}, \mathbf{y}]$. Thus, there exists some $\mathbf{x}' \neq \mathbf{x}$ or $\mathbf{y}' \neq \mathbf{y}$. According to the definition of $\mu_i(x_i, z_i)$ and $\nu_i(y_i, z_i)$ in Eq. 7, we would obtain $\mu_i(x_i, z_i) = 0$ or $\nu_i(y_i, z_i) = 0$ which results in $P(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{V}) = 0$, which is a contradiction.

Therefore, we prove the Theorem. □

[2] http://web.eecs.umich.edu/~jjcorso/r/a2d/

Acknowledgments. This work was partially supported by the National Science Foundation CAREER grant (IIS-0845282), the Army Research Office (W911NF-11-1-0090) and the DARPA Mind's Eye program (W911NF-10-2-0062).

References

[1] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. J. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. W. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Video in sentences out. In *UAI*, 2012.

[2] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *IEEE International Conference on Computer Vision*, 2013.

[3] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in nd images. In *ICCV*, 2001.

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, 2008.

[5] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[6] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

[7] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[9] G. Floros and B. Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[10] V. Garro, A. Fusiello, and S. Savarese. Label transfer exploiting three-dimensional structure for semantic segmentation. In *MIRAGE*, Jun 2013.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[13] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *IEEE International Conference on Computer Vision*, 2009.

[14] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision*, 2013.

[15] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[16] G. Hartmann, M. Grundmann, J. Hoffman, D. Tsai, V. Kwatra, O. Madani, S. Vijayanarasimhan, I. Essa, J. Rehg, and R. Sukthankar. Weakly supervised learning of object segmentations from web-scale video. In *European Conference on Computer Vision Workshops*, 2012.

[17] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[19] Y. Iwashita, A. Takamine, R. Kurazume, and M. S. Ryoo. First-person animal activity recognition from egocentric videos. In *IEEE International Conference on Pattern Recognition*, 2014.

[20] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *European Conference on Computer Vision*, 2012.

[21] A. Jain, S. Chatterjee, and R. Vidal. Coarse-to-fine semantic video segmentation using supervoxel trees. In *IEEE International Conference on Computer Vision*, 2013.

[22] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, C. Snoek, et al. Action localization with tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[23] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *IEEE International Conference on Computer Vision*, 2013.

[24] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[25] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *European Conference on Computer Vision*, 2012.

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011.

[27] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision*, 2011.

[28] A. Kundu, Y. Li, F. Daellert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision*, 2014.

[29] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *IEEE International Conference on Computer Vision*, 2009.

[30] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *European Conference on Computer Vision*, 2010.

[31] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[32] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *Advances in Neural Information Processing Systems*, 2010.

[33] I. Laptev. On space-time interest points. *International Jour-*

nal of Computer Vision, 64(2):107–123, 2005.

[34] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *IEEE International Conference on Computer Vision*, 2011.

[35] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *IEEE International Conference on Computer Vision*, 2013.

[36] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[37] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[38] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[39] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff. Action recognition and localization by hierarchical space-time segments. In *IEEE International Conference on Computer Vision*, 2013.

[40] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[41] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[42] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*, 2010.

[43] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *IEEE International Conference on Computer Vision*, 2013.

[44] M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *IEEE International Conference on Computer Vision*, 2011.

[45] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications Journal*, 2012.

[46] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[47] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[48] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[49] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[50] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *IEEE International Conference on Pattern Recognition*, 2004.

[51] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr. Urban 3d semantic modelling using stereo vision. In *IEEE International Conference on Robotics and Automation*, 2013.

[52] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[53] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, 2012.

[54] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[55] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, 2011.

[56] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[57] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, 2012.

[58] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 2013.

[59] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of IEEE International Conference on Computer Vision*, 2013.

[60] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[61] C. Xu, R. F. Doell, S. J. Hanson, C. Hanson, and J. J. Corso. A study of actor and action semantic retention in video supervoxel segmentation. *International Journal of Semantic Computing*, 2014.

[62] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *European Conference on Computer Vision*, 2012.

[63] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[64] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[65] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *IEEE International Conference on Computer Vision*, 2013.

[66] S. Zheng, M.-M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother, and P. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[67] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *IEEE International Conference on Computer Vision*, 2013.