End-to-End Joint Semantic Segmentation of Actors and Actions in Video

Jingwei Ji[1], Shyamal Buch[1], Alvaro Soto[2], and Juan Carlos Niebles[1]

[1]Stanford Vision and Learning Lab,[2]Pontificia Universidad Catlica de Chile
{jingweij, shyamal, jniebles}@cs.stanford.edu, asoto@ing.puc.cl

Abstract. Traditional video understanding tasks include human action recognition and actor/object semantic segmentation. However, the combined task of providing semantic segmentation for different actor classes simultaneously with their action class remains a challenging but necessary task for many applications. In this work, we propose a new end-to-end architecture for tackling this task in videos. Our model effectively leverages multiple input modalities, contextual information, and multitask learning in the video to directly output semantic segmentations in a single unified framework. We train and benchmark our model on the Actor-Action Dataset (A2D) for joint actor-action semantic segmentation, and demonstrate state-of-the-art performance for both segmentation and detection. We also perform experiments verifying our approach improves performance for zero-shot recognition, indicating generalizability of our jointly learned feature space.

Keywords: semantic segmentation · actor · action · video · end-to-end · zero-shot

1 Introduction

Action understanding is one of the key tasks in the field of video analysis. Recent progress has been primarily focused on obtaining a relatively coarse understanding of human-centric actions in video [10, 12]. However, a more comprehensive understanding of actions requires to identify fine-grained details from a video sequence, such as what actors are involved in an action, how are they interacting, and where are their precise spatial locations. Such pixel-level joint understanding of actors and actions can open a series of new exciting applications, such as activity-aware robots able to accurately localize potential users, understand their needs, and interact with them to assist. Furthermore, expanding action understanding to non-human actors is essential for autonomous vehicles.

More fundamentally, delving deeper into the synergies between action recognition and object segmentation can be mutually beneficial, and improve overall video understanding. As an example, an accurate and fine grained spatial identification of the main actors involved in an action may increase the robustness of action recognition. Similarly, the correct recognition of the underlying action in a video sequence can facilitate the identification of relevant finer details, such as

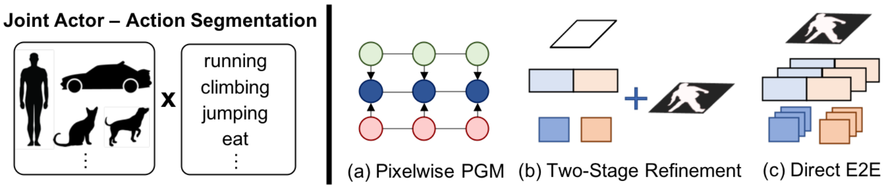

Fig. 1: We tackle the problem of joint actor-action semantic segmentation in videos, which requires simultaneous pixelwise recognition of different actor and action classes. Prior work have proposed (a) pixelwise probabilistic graphical model (PGM) approaches [28, 27] and (b) two-stage refinement approaches [11]. (c) In this work, we propose a new direct end-to-end architecture that combines video action recognition and actor segmentation in a single unified model.

precise actor locations. Building a working model that takes advantage of these insights requires careful architecture design, incorporating two design components in a synergistic manner. First, this would require integrating a finer localization of the main actors executing the action within the action recognition pipeline. Second, the model must also have a strong understanding of the activities occurring within the video, a task that is more traditionally context-dependent. With the above observation and philosophy, we tackle the problem of **joint actor-action semantic segmentation**, which asks the perception algorithm to predict actor and action class labels at the pixel level in the input video clip.

This task of actor-action semantic segmentation is inherently challenging. First, we desire actor and action knowledge to be learned jointly to benefit each other's prediction. At the same time, the learned representations should be decoupled well enough to prevent an explosion of joint classes in the actor-action cross product space. Second, although the problem can be addressed by a multi-stage refinement approach – where actor detection, semantic segmentation, and action recognition are separate – a direct end-to-end design can reduce multi-stage engineering. Third, in contrast with pixelwise segmentation on a static image, contextual information from other frames may need to be considered to predict accurate action labels.

A few prior works [28, 27, 29, 11] have examined the challenging joint task of actor-action semantic segmentation, as illustrated in Figure 1. For example, Xu *et al.* [27] proposed a graphical model that adaptively groups spatial and temporal information from supervoxels in videos. Kalogeiton *et al.* [11] proposed a joint actor-action detector on single frames and then perform segmentation. While all of these methods made important progress towards the actor-action semantic segmentation problem, they either do not decouple the actor and action label spaces, rely on two-stage refinement, or do not effectively leverage contextual information. In our work, we address all of these challenges simultaneously. Thus, our **contributions** can be summarized as follows:

- We propose a new *end-to-end* architecture for actor-action semantic segmentation in video that effectively leverages multiple input modalities, contextual information from video, and joint multitask learning.
- We observe that our approach significantly outperforms prior state-of-the-art methods on both actor-action segmentation and detection in videos.
- Finally, we demonstrate the generalization capabilities of our network for stronger zero-shot detection of actor-action pairs over previous work.

2 Related Work

In this section, we discuss related work in instance segmentation from single images, recent advances on convolutional networks for video analysis and actor-action semantic segmentation.

2.1 Instance Segmentation

The instance segmentation problem for images has been widely studied with significant recent advances [21, 16, 17, 4, 5, 8]. Recent progress in this field includes DeepMask [16] and its following works [17, 4, 5] resort to only instance segmentation without semantic labels, or predicting semantic labels as a second stage. Another approach is predicting masks and semantic labels in parallel, as in Mask R-CNN [8], which is more flexible and straightforward. Although these approaches focus on static images, they provide a gateway to perform per-frame semantic segmentation in videos.

Another line of work directly tackles the problem of video object segmentation [2, 24, 13]. These algorithms generally require access to ground truth mask annotation in the first frame of test video. In practice, such detailed annotation is not present in real-world applications at inference time. Furthermore, these approaches attempt to build object-agnostic algorithms that do not have access to the object class during training time, and are not capable of predicting object labels during test time. In this paper, we are interested in performing actor-action segmentation when no annotations are available at inference time, and in generating pixel-wise label inference of foreground actors and background pixels.

2.2 3D ConvNets for Action Recognition

A significant amount of research [10, 23, 3, 26, 18, 22, 12] has considered the problem of action classification in video clips. In that setting, the input is a short video sequence, and the goal is to provide a single action label for the full clip, typically focused on human actions. Recent work [23, 3, 26, 18] has focused on leveraging 3D convolutional networks as the core of the action recognition framework. Recently, Carreira *et al.* [3] proposed the I3D architecture, which considers a two-stream network configuration [22, 12] and performs late fusion of the outputs of individual networks trained on RGB and optical flow input, trained separately. Other recent works [26, 18] have proposed similar 3D architectures for recognition, focusing on improving performance while reducing computation cost. In this work, while we aim to tackle a more fine-grained and spatially-oriented action understanding problem, we draw inspiration from these frameworks in our model design. We elaborate on some of the key architectural advances for our stronger joint action-actor performance in Section 3.2.

2.3 Actor-Action Semantic Segmentation

The actor-action semantic segmentation problem is first raised by Xu et al. [28], where they collected the dataset A2D to study the problem and introduced a trilayer model as a first approach to solve this problem. Following [28], Xu *et al.* [27] proposed a Grouping Process Model (GPM) which adaptively groups segments during inference, and Yan *et al.* [29] proposed a weakly supervised method with only video-level tags being used in training. These methods depend on Conditional Random Fields (CRF) [21] for pixel-level segmentation, and can be classified as probabilistic graphical model (PGM) approaches. With the recent success of object detection and instance segmentation using deep neural networks, Kalogeiton *et al.* [11] proposed an actor-action detection network on single frames in video, then applied SharpMask [17] to generate actor-action semantic segmentation. This approach is one of two-stage refinement, whereby the main model provides detection boxes which are used in tandem with output segmentation masks from another method to provide refined outputs.

Our work advances the state-of-the-art in actor-action semantic segmentation. To the best of our knowledge, our method is the first end-to-end deep model for this task. In particular, we propose a unified framework to jointly consider temporal context, actor classification, action recognition, bounding box detection and pixel level segmentation.

3 Proposed Model

Our goal task is to provide semantic segmentation across the joint actor-action class space in input video data. To meet the challenges described in section 1, we hold the following model design philosophy: (1) To be able to decouple actor and action learning, actor and action classification heads should be separated and have their own set of parameters. (2) The network should be end-to-end with knowledge sharing between actor and action understanding, thus we have actor and action sharing the backbone structure for frame feature extraction. (3) The temporal context should be utilized for better action recognition, thus we leverage the short-term and contextual motion cues by 3D convolution layers and flow input.

We propose to tackle this with an end-to-end deep architecture, as illustrated in Fig. 2. Our approach takes both RGB and flow video clips as input streams, leveraging information from both appearance and motion in the video. Our network simultaneously outputs mask segmentation, and classification for actors in the branch of pixel-level actor localization, which will be elaborated in Section

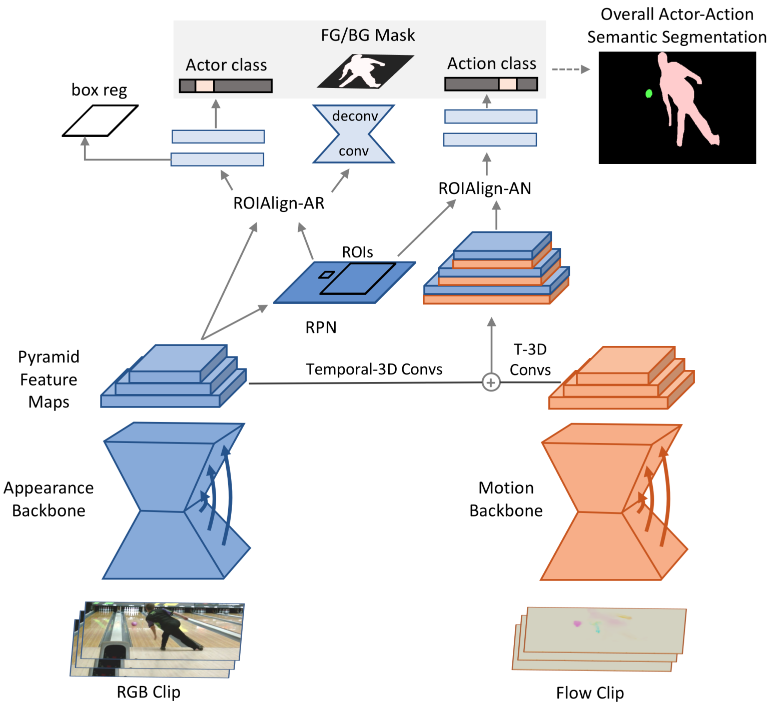

Fig. 2: Overview of our end-to-end architecture for joint actor-action segmentation. The model takes as input a context window of both RGB and optical flow frames, and outputs the semantic segmentation for all actor classes of interest jointly with their actions. Note that in the above example both the bowling ball and the adult are segmented in the same forward pass - we visualize the FG/BG mask for the adult only for clarity. See Section 3 for more model details.

3.1. With the actor localization provided, 3D feature maps from RGB and flow streams are jointly employed to perform action recognition. Details for action recognition will be found in section 3.2. We share the appearance backbone parameters and activations between actor and action branches so that they benefit from each other's knowledge, and the parameters are jointly optimized through the end-to-end joint learning of our architecture (Section 3.3).

3.1 Pixel-level Actor Localization

For the sub-task of actor localization, we build upon recent successful architectures for 2D object detection and semantic segmentation, such as Faster R-CNN [19] and Mask R-CNN [8]. In particular, we adopt a structure similar to that of Mask R-CNN to achieve pixel-level actor localization.

Appearance Backbone. Given a RGB input clip, each frame would go through the appearance backbone first to generate feature maps that will be used for next steps. On the one hand, the generated feature maps should be of high-level abstraction such that they capture the essential concepts of actor, on the other hand they also maintain the pixel-wise information to be leveraged for segmentation prediction. Therefore, we choose to use Feature Pyramid Network (FPN) [14] backbone feature extractor. Here the FPN is composed of a vanilla ResNet-101 [9] and a top-down architecture with skip connections between activations of the same resolution. The ResNet-101 is powerful at extracting high-level features, while the skip connections avoid low-level information being lost. Note that FPN is fully convolutional, which preserves the spatial correspondence between the output feature maps and input frames. Rather than outputting a single feature map for every single frame, the appearance backbone outputs a pyramid of feature maps, which is composed of feature maps from different resolution. In our network, we utilize feature maps from 4 different resolutions. Denote the height and width of input frame as H and W, the resolutions are (H, W) devided by 4, 8, 16, 32, respectively. Considering that we are working on a variety of 'actors' from *baby* and *adult* to *bird* and *ball*, the pyramid feature maps help detecting actors of different scales.

Region Proposal Network. As in Mask R-CNN, the next step of actor localization is done by Region Proposal Network (RPN) [19]. Given the pyramid feature maps, the RPN generates Region of Interests (RoIs) in the form of bounding boxes . Feature maps of different resolution will go through the same RPN to generate a bunch of RoIs, and the final RoIs are the concatenation of all of them. Note that different from [11], we only use the feature maps output by appearance model to generate RoIs, while they also use features from motion model. We found in experiment that RoIs from appearance models are of much higher quality than those from motion model, thus we stick with RoIs from appearance model only.

Multitask Heads. With the RoIs generated by RPN, and the pyramid feature maps of each video frame, an RoIAlign operation [8] is executed to crop and resize feature map according to RoI bounding boxes. Different from the RoI pooling operation in [19], RoIAlign fixes the misalignment and unnecessary quantization in spatial dimensions, and has shown better performance in [8]. An important fact is that RoIAlign finds the matching resolution of feature map from the pyramid according to the size of RoI, which enables the network to capture small actors such as *ball* and *bird*. To distinguish with the RoIAlign operation in action part, we name them RoIAlign-AR and RoIAlign-AN respectively.

The cropped and resized feature patch output by RoIAlign-AR will be fed into multiple heads to fulfill different sub-tasks. This is in line with the similar setup in Mask R-CNN. In total, there are three parallel sub-tasks in the pixel-level actor localization: (1) bounding box regression, (2) actor classification, and (3) foreground/background segmentation. The bounding box regressor and actor classifier is composed of fully connected layers operated on flattened feature patch, while the segmentation head is fully convolutional (conv and deconv layers).

3.2 Two-stream Action Recognition with Temporal Aggregation

Backbones. For action recognition, different from actor branch, two backbones are used. On the one hand, the same backbone from appearance model is shared such that appearance features of the actors also contribute in action understanding. Besides information from appearance, as shown in [22], motion patterns are also valuable in action recognition. Therefore we build a mirrored motion backbone with a separate set of parameters, which takes in flow clips, and extracts motion patterns from them. These two backbones formulate the two-stream attributes of our model. Following [22, 3, 7], the input of the flow branch is a tensor of three channels with the x and y coordinates and the magnitude of flow.

Temporal Aggregation. As we discussed in section 1, one challenge in the actor-action semantic segmentation in video is how to leverage the temporal context information for better action recognition. Here we resort to 3D CNN as the ingredient to achieve temporal aggregation. We apply separate 3D convolutional layers on the top of the pyramid feature maps output by each backbone to aggregate the temporal context. The pyramid feature maps from two backbones are then concatenated at the corresponding resolution, which will be further employed for action recognition. Specifically, $3 \times 1 \times 1$ conv layers [18] are applied to feature maps of every spatial scale, so the information of neighboring frames are aggregated into 3D pyramid feature maps. We note that we adopt an efficient "top-heavy" design [26], focusing 3D temporal convolutions on the upper portion of the network. We demonstrate in section 4.3 that such aggregation of temporal context is helpful for improved performance.

After temporal 3D conv layers, the 3D pyramid feature maps from each backbone are concatenated on the corresponding resolution level. As suggested by [11] for 2D architecture, and corroborated by our own experiments, late fusion in standard action recognition approaches [22, 3, 7] does *not* work well when considering the joint task of actor/action recognition and semantic segmentation. Therefore we choose to fuse appearance and motion in the mid-level.

Action Classification. With the RoIs provided by actor localization branch, fused 3D pyramid feature maps go through another RoIAlign-AN layer. The cropped and resized 3D feature map output by RoIAlign-AN incorporate information not only from the local actor, but also temporal context via temporal layers, and spatial context with proper receptive fields. The rich spatial and temporal contexts provide sufficient information for pixelwise action recognition over localized regions.

3.3 Joint Learning of Actors and Actions

Our end-to-end network enables joint learning for actor and action classification and segmentation. Joint learning all subtasks force the backbone features to contain necessary information for actor detection, actor classification, action recognition and actor-action segmentation. We use a multitask loss for parameter optimization:

$$L = \lambda_1 L_{RoI-cls} + \lambda_2 L_{box-reg} + \lambda_3 L_{actor-cls} + \lambda_4 L_{action-cls} + \lambda_5 L_{mask} \quad (1)$$

where $L_{RoI-cls}$ and $L_{box-reg}$ are as defined in [6], and λ's are hyperparameters. Actor and action classification losses are the negative log likelihood of the ground truth class. Denote the set of actor classes as X, the set of action classes as Y, the ground truth actor class as x, action class as y, we have:

$$L_{actor-cls} = -\log p_X(x), L_{action-cls} = -\log p_Y(y). \tag{2}$$

The mask head generates $|X|$ masks corresponding to every possible actor. Assuming the ground truth actor class is k, then L_{mask} will only be computed on the k-th mask. As in [8], L_{mask} is defined as the average binary cross-entropy loss.

Note that the losses are computed respect to frames rather than the whole video. Together with the temporal layers, our network design and learning setup are able to train even when some of the context frames have missing annotations, while still leveraging temporal context to obtain better spatial action recognition for the joint task.

4 Experiments

Dataset Details. We train and evaluate our model on the Actor-Action (A2D) dataset [28] for joint actor/action semantic segmentation To the best of our knowledge, A2D is the largest dataset that covers multiple actor and action classes and provides pixel-level semantic labels, and is the only joint action-actor segmentation benchmark for video reported in prior work [28, 27, 11]. This dataset comprises of 3782 YouTube videos, with sparse pixel-level joint semantic segmentation annotations and instance bounding boxes over 3-5 frames for each video. A2D covers 7 actor classes: *adult*, *baby*, *ball*, *bird*, *car*, *cat*, *dog*, and 9 action classes: *climb*, *crawl*, *eat*, *fly*, *jump*, *roll*, *run*, *walk*, *none* (no action). We note that some of the joint classes in the cross products are invalid, e.g. *car-eating*, and we exclude them in training and inference, as per prior work.

Implementation Details. We implement our end-to-end architecture in TensorFlow [1]. For the spatial dimensions of our 3D network, we initialize the model by leveraging pre-trained weights from Mask R-CNN [8] on MS-COCO [15]. The ResNet-101 backbone in the optical flow input branch is separately initialized with pre-trained weights on ImageNet [20], as per prior work [11, 25]. The weights for the temporal convolutions do not leverage pre-trained weights and are randomly initialized. We use SGD optimizer with learning rate of 2e-4. Additional details and code are provided in our supplementary.

4.1 Joint Actor-Action Semantic Segmentation

Table 1 shows a comparison of our joint method against prior state-of-the-art methods. We note that these prior methods leverage external techniques to generate initial semantic segmentation masks, such as GBH [27] and SharpMask (SM) [17], before refining them. However, our approach is trained end-to-end to directly output pixelwise segmentation for both actors and actions.

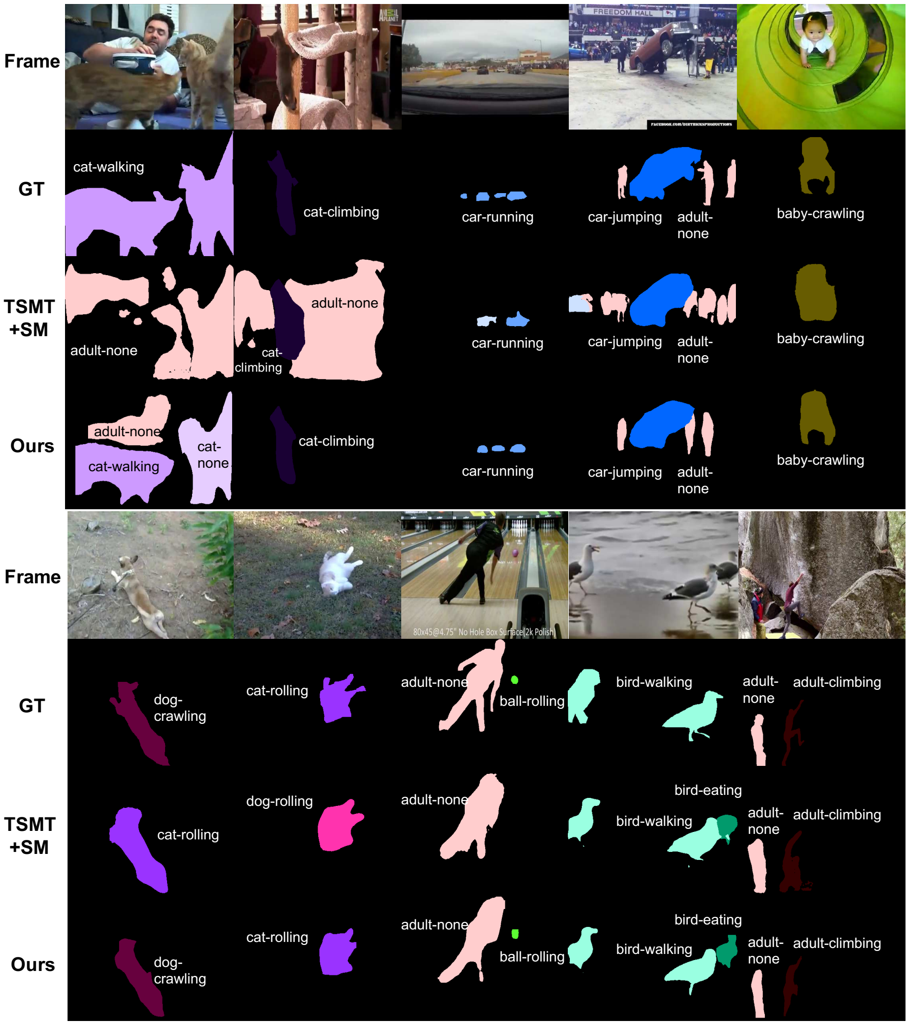

Fig. 3: Qualitative results. We visualize the input key frame and groundtruth (GT) semantic segmentation masks. The TSMT model + SharpMask (SM) outputs are provided by the authors of [11]. We qualitatively observe improved actor-action semantic segmentation performance of our end-to-end model in many cases over the prior work. Interestingly, we note that in some cases our method provides even more accurate predictions than the original groundtruth annotations, such as in the top left example with the adult and cats. See Sec. 4.1 for details and supplementary for video visualizations.

Table 1: Joint Actor-Action Semantic Segmentation Quantitative Results. We observe our end-to-end model significantly improved performance over prior state-of-the-art approaches using pixelwise PGMs or two-stage refinement architectures [28, 27, 11] for actor, action, and joint actor-action (A,A) semantic segmentation in videos. We provide additional discussion in Sec. 4.1, and detailed ablation analysis in Sec. 4.3.

Approach	**Actor**			**Action**			**Joint (A,A)**		
	ave	**glo**	**mIoU**	**ave**	**glo**	**mIoU**	**ave**	**glo**	**mIoU**
Trilayer [28]	45.7	74.6	-	47.0	74.6	-	25.4	76.2	-
GPM+TSP [27]	58.3	85.2	33.4	60.5	85.3	32.0	43.3	84.2	19.9
GPM+GBH [27]	59.4	84.8	33.3	61.2	84.9	31.9	43.9	83.8	19.9
TSMT [11] + GBH	72.9	85.8	42.7	61.4	84.6	35.5	48.0	83.9	24.9
TSMT [11] + SM	73.7	90.6	49.5	60.5	89.3	42.2	47.5	88.7	29.7
Ours	**79.1**	**94.5**	**66.4**	**62.9**	**92.6**	**46.3**	**51.4**	**92.5**	**36.9**

We report three different types of metrics following [11]: (1) *ave* - the average per-class accuracy, (2) *glo* - the global pixel accuracy, and (3) *mIoU* - mean pixel Intersection-over-Union. Pixel accuracy is the percentage of pixels whose labels are being correctly predicted. *ave* first computes pixel accuracy for each class and then average over classes, while *glo* is computed over all pixels. As noted in [11], mIoU is the most representative metric on this dataset since it's not biased towards background pixels, though we demonstrate consistent improved performance across the board. We report the comparison on evaluation with prior works in Table 1, where we measure all three metrics on actor label, action label, and actor-action pair label (Joint(A,A)) settings, as described in [11]. Since a correct labeling for actor-action pair requires both actor and action labels to be correct, the corresponding numbers in Joint(A,A) are in general lower than actor and action alone. Note that these metrics are not only indicating foreground/background segmentation quality, but also actor classification and action recognition performance at the pixel level.

Figure 3 shows qualitative results from our experiment. We observe qualitative improvement across many actor and action classes over prior work [11]. Interestingly, we observe that model predictions can pick up on new instances not labeled in the groundtruth annotations, and in some cases even provides more accurate joint class labels than the groundtruth. We note that these qualitative observations further highlight the importance of mIoU as a metric over *ave* and *glo* going forward for joint actor-action semantic segmentation. We also include qualitative results on video in our supplementary.

4.2 Joint Spatial Detection of Actors and Actions

Our joint model predicts bounding boxes as an auxiliary task for the segmentation in the same forward pass. Thus, we also verify that our joint end-to-end approach for actor-action segmentation results in improved spatial detection of actors and actions for spatial *detection* as well, to compare against prior work [11]. Once again, we evaluate our method for actor and action spatial detection

Table 2: Joint Spatial Detection of Actors and Actions (mAP). Since our end-to-end model outputs detection bounding boxes as an auxiliary task, we can also benchmark our approach against prior work [11] - we observe significant improvement in performance for actor, action, and joint actor/action (A,A) spatial detection.

Method	**Actors**	**Actions**	**Joint Actor-Action (A,A)**
TSCT [11]	67.2	60.2	49.2
TSHR [11]	67.9	59.6	49.6
TSMT [11]	68.3	60.0	48.9
Ours (Current)	**77.2**	**62.4**	**55.5**

performance separately, as well as the overall performance on the joint task. Our experimental results are summarized in Table 2. We demonstrate that our method also outperforms the state-of-the-art over all three metrics, which is natural since we aim at finer level problem and have boosted the performance on that, hence the performance of coarser level problem is also improved.

4.3 Ablation Analysis

In this section, we examine critical components in our architecture to verify that each of them play an important role in contributing to the overall performance. **Mask R-CNN baseline.** Due to the recent success of Mask R-CNN on instance-wise semantic segmentation, we first perform a baseline experiment where we evaluate the power of Mask R-CNN as a direct input to achieve actor-action semantic segmentation. Consequently, this baseline considers only a single semantic label. Similar as the setup in [28, 27], we use the cross-product of the actor and action labels in A2D as single semantic labels, e.g. *baby-crawling*. With number of actor classes $|X| = 7$, number of action classes $|Y| = 9$, there are 63 cross-product labels, out of which 43 are valid. Invalid labels include *adult-flying*, etc, and are not considered during training and testing. The baseline experiment is performed only on single frame and single stream, with no temporal information or flow input. The parameters are initialized with pretrained weights on MS-COCO [15].

Comparing the first two rows in Table 3, we can observe the significant performance gap between our full model and Mask R-CNN baseline. Tackling actor-action segmentation problem directly using Mask R-CNN has the following weaknesses: (1) The number of cross-product classes is $O(|X||Y|)$, which makes it hard to scale up with more classes of actor and action to be considered in future works. (2) Mask R-CNN aims at segmentation on single RGB frame, while in video, especially when action recognition is involved, temporal context and motion patterns should be leveraged in the model. (3) Actor and action classification are treated symmetrically, which does not reflect the intuition that actor is more defined spatially while action is also relying on the motion and temporal cues. Considering all these weaknesses, we design our network to decouple the actor and action classification heads, include temporal architecture, take in flow information, and use actor spatial localization to guide action recognition on feature maps aggregating temporal context.

Table 3: Ablation Analysis: Actor-Action Segmentation Results. We verify the contribution of each component of our network over baselines such as Mask R-CNN [8] and a Two-Stage Refinement variant of our method (Ours w/o Mask + SM), based on [11] and also leveraging SharpMask (SM) [17]. Please see Sec. 4.3 for more details.

Approach	**Actor**			**Action**			**Joint (A,A)**		
	ave	**glo**	**mIoU**	**ave**	**glo**	**mIoU**	**ave**	**glo**	**mIoU**
Ours (Full)	*79.1*	**94.5**	*66.4*	**62.9**	**92.6**	**46.3**	**51.4**	**92.5**	**36.9**
Mask R-CNN [8] Baseline	62.8	84.2	33.7	59.6	84.0	30.3	41.7	82.5	19.1
Ours w/o Mask + SM [17]	76.6	92.2	60.3	60.7	90.3	42.9	49.0	89.8	32.4
Ours w/o Temp Context	79.0	94.1	66.1	61.8	92.0	45.5	50.3	90.2	35.3
Ours w/o Flow Stream	**79.5**	93.7	**66.5**	60.4	86.3	36.8	46.2	87.8	29.4
Ours w/o Joint Training	77.7	93.2	63.2	62.1	91.3	45.2	50.9	90.1	33.6

Ours w/o Mask (Two-Stage Refinement Baseline). As in [11], one approach to achieve actor-action segmentation is based on a two-stage scheme: first perform actor-action detection at bounding box level, then as a post-processing step, perform foreground/background segmentation within the bounding box limits using a standard segmentation method. This approach is not an end-to-end solution for actor-action segmentation, and the knowledge learned in segmentation part can not benefit the action recognition during training. To show the effect of having mask head in the end-to-end network, we perform an ablation experiment with mask head chopped off, only predicting actor-action bounding boxes, and using SharpMask [17] as the segmentation method following [11]. With better actor-action spatial detection as shown in 4.2, this two-stage method outperforms the similar experiment setup in [11]. Still, the two-stage method is not as good as the full model on any metrics listed in Table 3, which shows the necessity to include the mask head in the end-to-end architecture.

Ours w/o Temporal Context. To demonstrate the impact of temporal layers, we perform an experiment where we remove these layers. The removal of temporal layers turns the network into a single frame model, which does not take the neighboring frames into account when predicting action label for each RoI. As shown in the fourth row in 3, although the actor segmentation is not much affected, the action segmentation performs worse without temporal layers aggregating information from its temporal context for each frame. Our exploration in leveraging temporal context on this task is partly limited by the fact that labels in A2D are temporally sparse, which may also be a limiting factor with regards to its relative impact on the performance improvement of our overall approach. We expect more future works on this task with focus on temporal context.

Ours w/o Flow Stream. As shown in related work [22, 3], information about motion patterns contained in optical flow is crucial for many action tasks. When the flow modality is absent, the actor performance is almost untouched, since the actor localization is not using optical flow directly in our model. Comparing between 'Ours (Full)' and 'Ours - Flow' in Table 3, motion cues significantly contributes to action recognition and thus actor-action segmentation.

Ours w/o Joint Learning. In our end-to-end network, we can choose from two training procedures: jointly learn all sub-tasks at the same time, or separately learn them one-by-one. To compare with our main joint learning approach, we perform a separate learning experiment whereby the actor branch is first trained until convergence, and then we train the layers related to action recognition. Note that actor and action branches share the same backbone feature extractor, so when separately trained, the final network can be biased to the actor localization subtask or action recognition subtask. By comparing the first and last row in Table 3, we can conclude that joint learning is helpful to avoid such subtask biases and achieves best benchmark performance.

4.4 Zero-shot Learning of Actions

A successful actor-action semantic segmentation model should not only infer the actor-action cross-product labels seen during training, but also be capable of generalizing to unseen actor-action pairs. This requires our model to maintain ability to *decouple* actor and action understanding while jointly learning them.

To verify the decoupling ability of our network, we follow the zero-shot learning experiment setup from [11] on A2D. We train the network 7 times, where each time one actor class x' is excluded for training its action labels. Note that we still train the actor classification for x', so during inference, the network can still localize and segment this actor. Formally, $L_{actor-cls}$ maintains the same, while action classification loss becomes $L_{action-cls} = -\mathbb{1}\{x' \neq x\} \log p_Y(y)$.

In order to maintain consistency on evaluation with respect to [11], we report the metric of AP as shown in Fig. 4. Each actor's AP is averaged over all valid actions of that actor. It shows that our method outperforms [11] on all actors on the AP metric. We can observe that actors like *ball* and *car* has less commonalities on actions with other features. A problem of AP is that, this metric can be interpreted as benefited from the overall learning capability of the network, or just the effect of decoupling ability.

We also analyze the zero-shot learning performance using the following metric for zero-shot learning of actions:

$$r_{zs} = \frac{AP_{zero-shot}}{AP_{full}}. \tag{3}$$

The performance ratio (r_{zs}) reflects the relative performance of zero-shot learning compared to normal learning setup where all actors are seen in the training. This metric removes the impact of the overall performance of the full model, and only examines the network's ability to decouple actor and action understanding. We compare r_{zs} on [11] and ours method in Fig. 4, where we can observe that our method matches with or outperforms [11] on all actors, demonstrating superior capability to capture commonalities of an action performed by various actors.

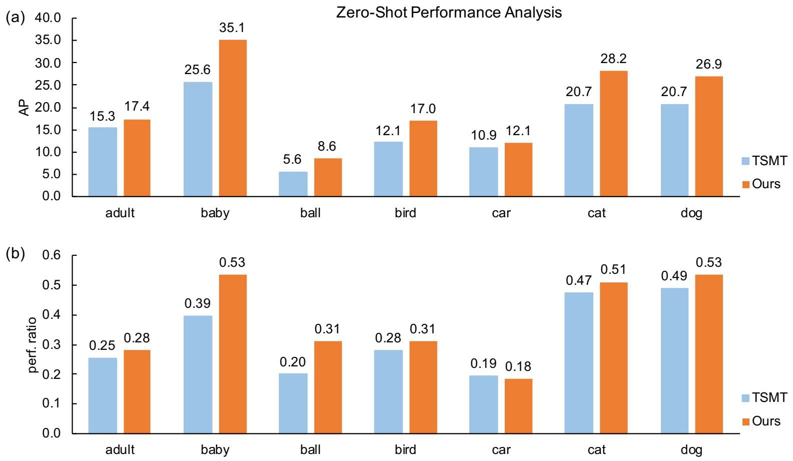

Fig. 4: Analysis of the generalizability and decoupling of the jointly learned embedding space for zero-shot detection of actor-action pairs. (a) We report *average precision* (AP) for each of the seven models, and observe stronger zero-shot performance compared with TSMT [11]. (b) We also observe a smaller *performance ratio* (r_{zs}) between full supervision and zero-shot inference. See Section 4.4 for more details.

5 Conclusion

We present a new end-to-end model able to jointly perform pixel-level actor-action segmentation and recognition. Our overall results and ablation study provide empirical support for the link between detailed spatial semantic segmentation in the joint pixelwise actor and action recognition pipeline. In particular, we demonstrate that the resulting model outperforms by a significant margin a model scheme based on a coarser bounding box actor-action localization, as well as other prior state-of-the-art work. We also show that it outperforms a model scheme based on a joint actor-action classification method that does not decouple actor and action classes at all. Consequently, our improved performance for the full joint task indicates that our overall end-to-end approach supports similar further directions for multitask research in video action understanding. Similarly, the stronger improvement in zero shot generalizability in terms of both raw performance and overall performance ratio indicates this approach shows strong promise for video representation learning directions as well.

Acknowledgments. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. This work is also partially funded by the Millennium Institute for Foundational Research on Data. We also thank NVIDIA for their DGX-1 donation.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: Computer Vision and Pattern Recognition (CVPR) (2017)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4733. IEEE (2017)
4. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: European Conference on Computer Vision. pp. 534–549. Springer (2016)
5. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3150–3158 (2016)
6. Girshick, R.: Fast r-cnn. In: International Conference on Computer Vision (ICCV) (2015)
7. Gkioxari, G., Malik, J.: Finding action tubes. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 759–768 (2015)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Herath, S., Harandi, M., Porikli, F.: Going deeper into action recognition: A survey. arXiv preprint arXiv:1605.04988 (2016)
11. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Joint learning of object and action detectors. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
13. Li, X., Qi, Y., Wang, Z., Chen, K., Liu, Z., Shi, J., Luo, P., Tang, X., Loy, C.C.: Video object segmentation with re-identification. arXiv preprint arXiv:1708.00197 (2017)
14. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. vol. 1, p. 4 (2017)
15. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
16. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. pp. 1990–1998 (2015)
17. Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: European Conference on Computer Vision. pp. 75–91. Springer (2016)
18. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5534–5542. IEEE (2017)

19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
20. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
21. Russell, C., Kohli, P., Torr, P.H., et al.: Associative hierarchical crfs for object class image segmentation. In: Computer Vision, 2009 IEEE 12th International Conference on. pp. 739–746. IEEE (2009)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. NIPS (2014)
23. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497. IEEE (2015)
24. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017)
25. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision. pp. 20–36. Springer (2016)
26. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. arXiv preprint arXiv:1712.04851 (2017)
27. Xu, C., Corso, J.J.: Actor-action semantic segmentation with grouping process models. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3083–3092 (June 2016). https://doi.org/10.1109/CVPR.2016.336
28. Xu, C., Hsieh, S.H., Xiong, C., Corso, J.J.: Can humans fly? Action understanding with multiple classes of actors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2015), `http://web.eecs.umich.edu/~jjcorso/pubs/xu_corso_CVPR2015_A2D.pdf`
29. Yan, Y., Xu, C., Cai, D., Corso, J.J.: Weakly supervised actor-action segmentation via robust multi-task ranking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)