

# Actor-Action Semantic Segmentation with Region Masks

Kang Dang<sup>1</sup>  
dang0025@e.ntu.edu.sg

Chunlun Zhou<sup>2</sup>  
czhou002@e.ntu.edu.sg

Zhigang Tu<sup>3</sup>  
tuzhigang@whu.edu.cn

Michael Hoy<sup>1</sup>  
mch.hoy@gmail.com

Justin Dauwels<sup>1</sup>  
JDAUWELS@ntu.edu.sg

Junsong Yuan<sup>4</sup>  
jsyuan@buffalo.edu

<sup>1</sup> ST Engineering - NTU Corporate Lab,  
School of EEE, Nanyang Technological  
University, Singapore

<sup>2</sup> Rapid-Rich Object Search (ROSE) Lab,  
School of EEE, Nanyang Technological  
University, Singapore

<sup>3</sup> State Key Lab of Information Engineer-  
ing in Surveying, Mapping and Remote  
Sensing, Wuhan University, China

<sup>4</sup> Department of Computer Science and  
Engineering, University at Buffalo, USA

---

## Abstract

In this paper, we study the actor-action semantic segmentation problem, which requires joint labeling of both actor and action categories in video frames. One major challenge for this task is that when an actor performs an action, different body parts of the actor provide different types of cues for the action category and may receive inconsistent action labeling when they are labeled independently. To address this issue, we propose an end-to-end region-based actor-action segmentation approach which relies on region masks from an instance segmentation algorithm. Our main novelty is to avoid labeling pixels in a region mask independently - instead we assign a single action label to these pixels to achieve consistent action labeling. When a pixel belongs to multiple region masks, max pooling is applied to resolve labeling conflicts. Our approach uses a two-stream network as the front-end (which learns features capturing both appearance and motion information), and uses two region-based segmentation networks as the back-end (which takes the fused features from the two-stream network as the input and predicts actor-action labeling). Experiments on the A2D dataset demonstrate that both the region-based segmentation strategy and the fused features from the two-stream network contribute to the performance improvements. The proposed approach outperforms the state-of-the-art results by more than 8% in mean class accuracy, and more than 5% in mean class IOU, which validates its effectiveness.

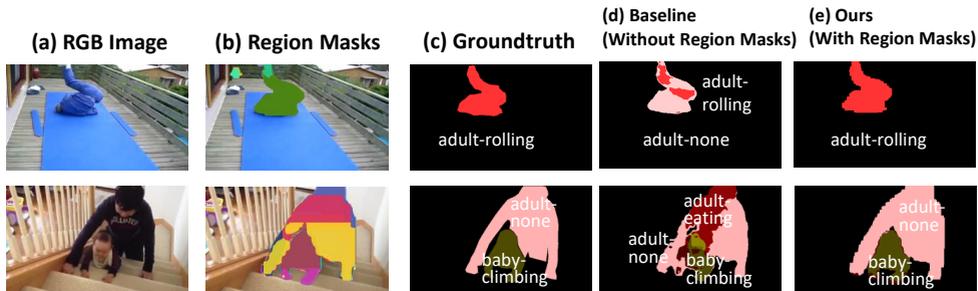


Figure 1: Examples that demonstrate region masks can be utilized to improve actor-action segmentation results. Baseline refers to the DeepLab-CNN [4] based method which does not utilize region masks.

## 1 Introduction

In this paper, we study the actor-action semantic segmentation problem. This task was recently proposed by Xu *et al.* [29], and targets joint labeling of both action categories (*e.g.*, running and jumping), and actor categories (*e.g.*, adult and cat) for every location in a scene. It has a variety of applications, ranging from robotics and autonomous driving to video captioning and video editing.

Despite much work in related research areas, such as image semantic segmentation [4, 4, 14, 19, 34, 35], video semantic segmentation [0, 16, 25], video object segmentation [12, 15, 32] and action detection [13, 21, 31], relatively few works [13, 28, 29, 30] have specifically targeted at the actor-action segmentation problem. These works study the actor-action segmentation problem though different lens; examples include a focus on the interaction between actor and action labels [13, 29], mechanisms to model long-range interaction of different actors [28], and training procedures which only require weakly-supervised actor-action labeling [30]. While the results are encouraging, we show that the performance can be amplified by harnessing region masks from instance segmentation algorithms.

Instance segmentation algorithms [00, 33] have made significant advancements recently, and is able to provide high-quality region masks. As demonstrated in previous region-based segmentation approaches [0, 3, 00], incorporating region masks into semantic segmentation networks can provide an adaptive receptive field and enforce consistent predictions over regions - such a property is also beneficial for the actor-action segmentation task. When an actor performs an action, different parts of the actor provide the different types of cues for the action [18, 26]. The movements of some parts better reflect the action than those of the others. When the pixels inside the actor regions are labeled independently, different parts may receive different action labels (see Fig. 1(d)). However, by enforcing pixels inside the actor regions to take the same action label, we can achieve consistent labeling for better actor-action segmentation performance (see Fig. 1(e)).

For the above reasons, we propose an end-to-end region-based actor-action segmentation approach as shown in Fig. 2. For each frame, we first obtain high-quality region masks using an object instance segmentation network [33] trained explicitly for a specific target dataset. Then, a two-stream network is used to learn two types of features which capture appearance and motion information respectively. Finally, the fused features from the two streams together with the region masks are passed to two region-based segmentation net-

works for predicting actor-action labels. We conduct a comprehensive set of experiments on the A2D dataset [29], and demonstrate that both region-based segmentation strategy and robust fused features from the two-stream network contribute to performance improvements. The proposed approach outperforms the state-of-the-art results by more than 8% in mean class accuracy, and more than 5% in mean class IOU, which confirms the effectiveness of the proposed method.

In summary, the contributions of our paper are as follows:

- Because many actions are reflected more prominently by movements of certain body parts instead of the whole body, different body parts of the actors may have inconsistent action labelling if pixels are labelled independently. Hence, we propose to utilize region masks to address the inconsistency issue of actor-action semantic segmentation problem. Through comprehensive experiments, we demonstrate the importance of exploiting region masks for consistent actor-action labeling.
- We design an end-to-end network specifically for actor-action semantic segmentation. In particular, 1) for each object we obtain multiple region masks and fuse their predicted categories via a region-to-pixel layer for final prediction, and 2) we use a two-stream network architecture taking an RGB image and an optical flow image as the input to further improve the semantic segmentation performance.

## 2 Related work

**Actor-action semantic segmentation.** Xu *et al.* [29] introduce the actor-action semantic segmentation application with a large-scale benchmark dataset. They further develop this application by addressing the inconsistent labeling issue with a graphical model [28], and later by proposing a weakly supervised method [30] to reduce labeling requirements. Kalogeton *et al.* [13] first perform object detection and then refine the bounding-box detections to generate segmentation results. Their main contribution is to study the joint modeling of actor and action categories with different loss functions. Qiu *et al.* [23] propose to use 3D convolution network and convolutional LSTM for modeling the spatio-temporal dependency. Different from these previous works, we directly incorporate high-quality region masks into a deep network for consistent actor-action labeling.

**Region-based semantic segmentation.** Several examples of methods which incorporate region based aggregation mechanisms into semantic segmentation frameworks are available. For example, Caesar *et al.* [9] propose a differentiable region-to-pixel layer and free-form region-of-interest pooling layer to improve the object boundary accuracies for image semantic segmentation. Arnab *et al.* [0] exploit object detection box constraints and superpixel region constraints with an end-to-end trainable higher-order CRF for image semantic segmentation. He *et al.* [10] incorporate superpixel region constraints into a convolutional network to address the multi-view semantic segmentation application. We extend this direction of study to another application, where we exploit region mask constraints to improve the consistency of actor-action labeling.

**Instance segmentation.** Many instance segmentation algorithms [5, 6, 10, 33] can be adopted to generate a set of candidate region masks as the input for our algorithm. In our paper we use the fully convolutional instance segmentation algorithm (FCIS) [33], which is an end-to-end framework for the task of instance-aware semantic segmentation with high accuracy and efficiency. Another closely related paper is from Dai *et al.* [6], which exploits

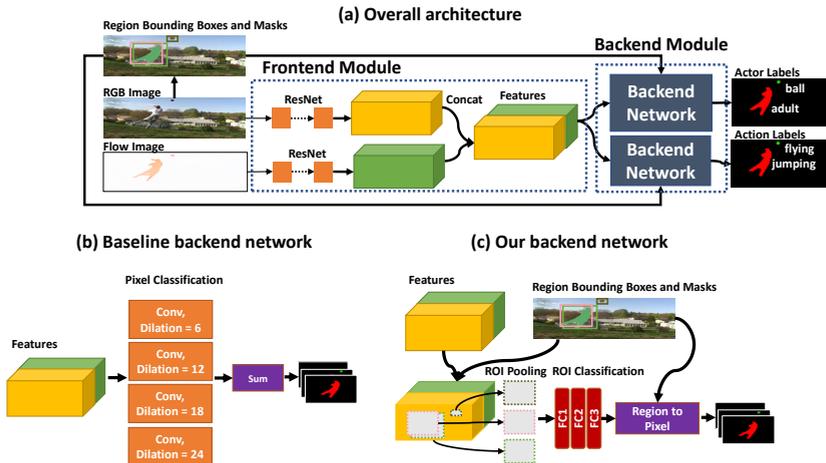


Figure 2: Overview of our end-to-end region-based actor-action segmentation approach. Our network consists of a front-end module and a back-end module containing two networks of identical structure. The front-end module is a two-stream network that learns both appearance and motion features. The concatenated features from both streams are passed to the back-end module which predict actor and action labels, respectively. Our main contribution is the adaption of region-based semantic segmentation networks as the back-end module.

multi-task network cascades for the task of instance-aware semantic segmentation. Our work shares a similar pipeline of [6] but with two major technical differences: (1) To improve robustness, we produce the labeling for the object by a region-to-pixel layer over multiple candidate masks, in contrast, object labeling is obtained by the weighted average in [6]; (2) We exploit a two-stream architecture to utilize both RGB and optical flow information to further improve the performance.

### 3 The proposed approach

For a given video, we assign an actor-action label (e.g., adult-running) to each pixel. The overview of the proposed approach is illustrated in Fig. 2. First, for each frame, a set of candidate region masks are generated using an instance segmentation algorithm [63] (Sec. 3.1). These region masks along with RGB and the corresponding optical flow image are taken as inputs. Then, a two-stream network (front-end module) is used to obtain both appearance and motion features (Sec. 3.2). Finally, the concatenation of both types of features is passed to two region-based semantic segmentation networks (back-end module) which predict actor and action labels, respectively (Sec. 3.3).

#### 3.1 Region Mask Generation

Principally, any mask proposal methods [22, 27] or instance segmentation algorithms [11, 53] can be adopted to generate a set of candidate region masks for each frame. Let  $\{\mathbf{b}_i, M_i\}_{i=1}^N$  be the set of generated candidate region masks where  $N$  is the number of region masks,  $\mathbf{b}_i \in \mathcal{R}^4$  is a bounding box and  $M_i \in \mathcal{R}^{W_b \times H_b}$  is a probability mask indicating the likelihood of each bounding box pixel belonging to the mask.

To achieve good performance, it is important to use a set of high-quality region masks. In

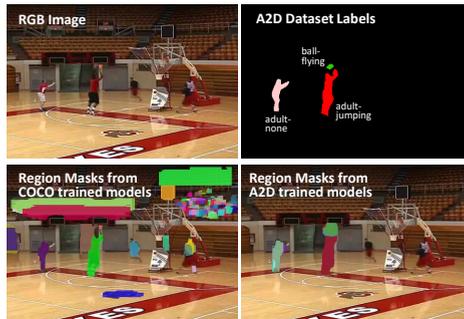


Figure 3: Examples of region masks generated from fully convolution instance segmentation (FCIS) algorithm [63]. To perform semantic segmentation on a particular dataset, *e.g.*, A2D dataset, FCIS should be fine-tuned on the same dataset, *e.g.*, A2D dataset, as region masks generated by FCIS trained on a more generic dataset, *e.g.*, COCO [47] may contain irrelevant background objects.

our experiments, we adopt a fully convolution instance segmentation (FCIS) [63] algorithm for region mask generation. For each frame, the predicted instance segmentation masks by the network are taken as candidate region masks. For actor-action segmentation on a specific dataset, we train the FCIS network on the same dataset instead of on a general large-scale dataset like COCO [47]. The FCIS network trained on the COCO dataset [47] is more generic but would produce more irrelevant background regions which could harm the actor-action segmentation performance (see Fig. 3 for illustration).

## 3.2 Front-end Module

We adopt a two-stream architecture similar to the one used by [47, 63] to generate robust feature representations to be used by the back-end semantic segmentation networks. The first stream models appearance information by taking as input an RGB image, and the second stream models motion information by taking as input the optical flows. We use FlowNet [8] to calculate optical flows which are converted into an RGB image with the method used in [47].

Each stream is adapted from a ResNet-101 classification network [9]. To preserve spatial details, the last two layers of the ResNet-101 are replaced by dilated convolutions, which increases the resolution of the output map from  $1/32$  to  $1/8$  of the original image size. After the feature maps from both streams are obtained, we concatenate them together as an input for the back-end module. Different from [47] which uses a late fusion for the sake of reducing model complexity, we apply this early fusion strategy to combine the two streams with better flexibilities.

## 3.3 Back-end Module

Given the set of candidate region masks  $\{\mathbf{b}_i, M_i\}_{i=1}^N$  and feature representation from the front-end module  $\mathbf{X} \in \mathcal{R}^{W \times H \times 2C}$  (where  $W$  and  $H$  are the width and height of the output feature map of one stream and  $C$  is the feature dimensionality), we want to predict both actor and action categories of each pixel. In other words, we predict two label maps  $\mathbf{Y}_{actor} \in \mathcal{R}^{W \times H \times K_{actor}}$  and  $\mathbf{Y}_{action} \in \mathcal{R}^{W \times H \times K_{action}}$ , where  $K_{actor}$  and  $K_{action}$  are the number of actors and actions to be predicted. We employ a multi-task formulation with two networks to achieve this goal. The first network is to predict the actor label and the second network is to

predict the action label. Similar to previous works [9, 12, 19], the cross-entropy loss is used for each branch. Specifically, the training loss for each pixel  $j$  is defined by

$$L_{j,loss} = -\log(p_{j,actor}(o_{j,actor})) - \log(p_{j,action}(o_{j,action})), \quad (1)$$

where  $o_{j,actor}$  is the ground-truth actor label and  $o_{j,action}$  is the ground-truth action label. At test time, the joint actor-action probability is the element-wise product of the two probability vectors.

In DeepLab CNN [9], multi-scale dilated convolution layers are applied to the front-end feature map to compute class probabilities directly (see Fig. 2(b)). However, class probabilities of each pixel are independently computed, and the scale set of receptive fields are fixed a priori. This may lead to undesirable results where the body of a single actor is segmented into multiple parts of different action labels (see examples in Fig. 4(a)), and small-sized objects may be missed (see examples in Fig. 4(b)). To address this problem, in our back-end network (see Fig. 2(c)), we utilize region masks to adaptively select receptive fields and enforce the pixels in each region mask to have a consistent labeling.

First, we obtain the overall region mask scores  $s_{i,actor}$  and  $s_{i,action}$  for each bounding box  $\mathbf{b}_i$ . To do this, we classify each bounding box via an ROI pooling layer and several fully connected (FC) layers, similar to the Faster R-CNN framework [24]. Secondly, after obtaining region classification scores, we use a region-to-pixel layer to transform the region scores to pixel scores following [9]:

$$\mathbf{y}_{actor}^j = \max_{i=1,\dots,N, j \in \mathbf{b}_i} (m_{i,j} \times \mathbf{s}_{i,actor}), \quad (2)$$

$$\mathbf{y}_{action}^j = \max_{i=1,\dots,N, j \in \mathbf{b}_i} (m_{i,j} \times \mathbf{s}_{i,action}), \quad (3)$$

where  $j$  refers to an image pixel,  $m_{i,j}$  refers to the probabilities that the pixel  $j$  belonging to the mask  $i$ ,  $s_{i,actor}$  and  $s_{i,action}$  refer to the classification scores of mask  $i$  for actor and action categories, respectively. This region-to-pixel layer serves to resolve labeling conflicts when a pixel belongs to multiple region masks, i.e., we take the max among them to calculate the fused score. The resulting pixel scores  $\mathbf{y}_{actor}$  and  $\mathbf{y}_{action}$  are then passed to a softmax layer to output class probabilities.

### 3.4 Network Training

As the FC layers in our network contain quite a large number of parameters, instead of training the whole network end-to-end, we adopt a two-stage training procedure to accelerate the training. We implement our approach using PyTorch [20].

We initialize the front-end network with the pretrained model from [12]. At the first stage, we train the front-end with a light-weight back-end network with fewer parameters. We train our front-end network with the baseline back-end shown in Fig. 2(b) using stochastic gradient descent (SGD). The learning rates for the front-end and back-end networks are set to  $2.5 \times 10^{-4}$  and  $5 \times 10^{-3}$ . We use a mini-batch size of 10 and train the network for 20000 iterations.

At the second stage, we fix the parameters of the front-end network and train our back-end. Note that only the fully connected layers are learn-able. Region pooling and region-to-pixel layers do not contribute to additional parameters. We use a learning rate of  $2.5 \times 10^{-4}$  and a smaller mini-batch size of 1 to train the network for 80000 iterations.

	global pixel accuracy			mean class accuracy			mean class IoU		
	actor	action	actor-action	actor	action	actor-action	actor	action	actor-action
Baseline (RGB only)	<b>95.7</b>	<b>93.5</b>	<b>93.0</b>	77.0	58.5	42.7	<b>67.1</b>	46.0	32.1
Ours (RGB only)	95.0	92.9	92.5	<b>85.5</b>	<b>68.8</b>	<b>51.5</b>	67.0	<b>48.1</b>	<b>34.5</b>
Baseline (RGB + flow)	<b>95.8</b>	<b>93.9</b>	<b>93.5</b>	78.0	61.7	46.1	<b>68.1</b>	49.1	34.9
Ours (RGB + flow)	95.3	93.4	93.0	<b>86.0</b>	<b>70.7</b>	<b>56.4</b>	<b>68.1</b>	<b>51.1</b>	<b>38.6</b>

Table 1: Comparison with a DeepLab-CNN [4] type baseline, which does not use region masks.

	global pixel accuracy			mean class accuracy			mean class IoU		
	actor	action	actor-action	actor	action	actor-action	actor	action	actor-action
Baseline (Non-Boundary)	<b>97.7</b>	<b>96.3</b>	<b>95.8</b>	85.1	67.8	51.6	<b>75.6</b>	53.9	39.0
Ours (Non-Boundary)	97.3	96.0	95.7	<b>88.9</b>	<b>73.6</b>	<b>57.8</b>	73.3	<b>56.1</b>	<b>41.8</b>

Table 2: . Comparison with a DeepLab-CNN [4] type baseline for the image regions excluding object boundaries.

## 4 Experiments

In our experiments, we have several objectives. Firstly, we demonstrate the effectiveness of our approach by comparing it with the DeepLab type baseline, as illustrated in Fig. 2(b), which does not utilize region masks. Secondly, we analyze the various factors that affect the performance, such as the inclusion of optical flow and using different mask proposals. Lastly, we demonstrate our method’s performance in comparison with the current state of the arts.

**Dataset:** We evaluate our method on the A2D dataset [29], which is the only dataset providing pixel-level dense annotation for actor and action pairs. The dataset consists of 3782 videos which are labeled with both pixel-level actors (*e.g.*, adult, boy, car, and etc.) and actions (*e.g.*, climbing, eating, walking and etc.) for sparsely sampled frames. We use identical partition scheme as previous works [13, 28, 29, 30], *i.e.*, 9561 frames for training and 2365 for testing.

**Evaluation protocol:** We measure the performance using three metrics: (1) global accuracy measures the overall pixel classification accuracy; (2) mean per-class accuracy is obtained by first calculating classification accuracy for each class separately followed by averaging over all the classes; (3) mean per-class IOU is similar to mean per-class accuracy but evaluates the performance at pixel level using intersection over union (IOU). The global accuracy is dominated by frequently occurring objects, while the other two metrics bias towards rarely occurring classes. We calculate these metrics in three settings: actor, action and joint actor-action. Unless otherwise specified, we report the performance using both RGB and optical-flow images, and using region masks from FCIS model trained on the A2D dataset.

**Usefulness of actor region masks:** Table 1 shows the results of our method compared to a DeepLab type baseline which does not use region masks. For the global pixel accuracy metric, our approach performs slightly worse than the baseline. This may be caused by the resolution of region masks generated by our FCIS model being only the half of the baseline output, which leads to boundaries inaccuracies. Our method performs around 10% better than the baseline in mean class accuracy for all the three settings, and around 3% better in mean class IOU for the action and actor-action settings. For the latter two metrics, the performance boost mainly comes from two situations. Firstly, for some actions like adult-jumping and bird-rolling which are prominently reflected by certain body parts, an actor region is enforced to have consistent labeling with region masks, while the baseline tends to

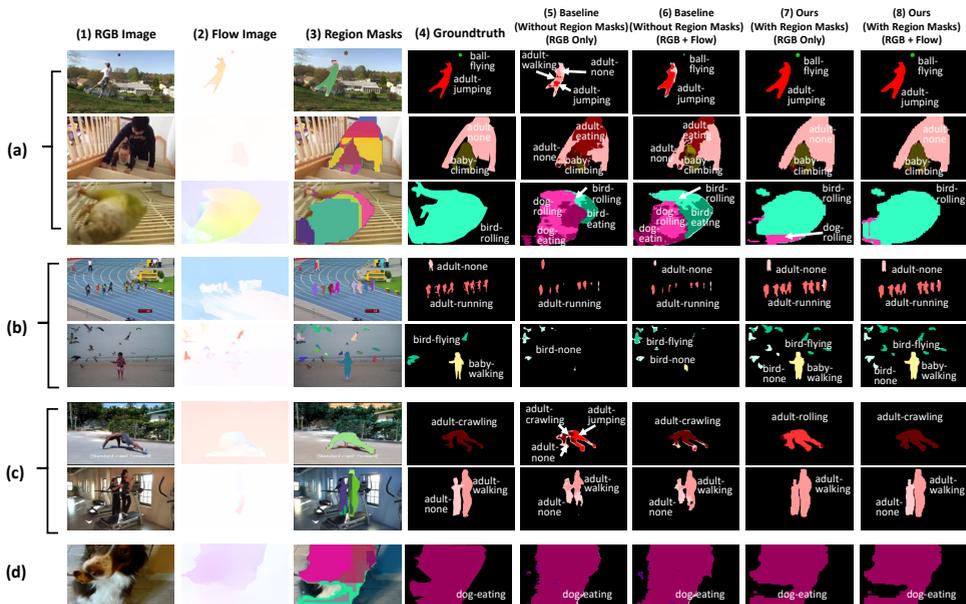


Figure 4: Examples of the actor-action semantic segmentation for our method contrasted against a DeepLab-CNN [9] type baseline which does not use region masks. (a) Examples that demonstrate our method yields more coherent segmentations for challenging poses or actions. (b) Examples that illustrate our method better captures small objects. (c) Examples that both the region masks and optical flow can improve the results. (d) A failure example due to inaccurate region masks.

segment the actor region into several parts of different action labels (see Fig. 4(a)). Secondly, small objects may be missed by the baseline because of its inflexible selection of receptive fields (see Fig. 4(b)). In contrast, the region masks used by our method can provide adaptive receptive fields according to object sizes. In Table 5, the performance of adult-jumping and bird-rolling and the performance of small objects such as bird and ball are significantly improved by our approach, which justifies our analysis. Fig. 4(c) shows examples that both the region masks and optical flow can improve the results. Fig. 4(d) shows a failure case of our method due to inaccurate region masks.

It is well known that incorporating region masks are particularly beneficial for segmentation of object boundaries [9, 10]. To find out how much of the performance gain can be attributed to resolving inconsistent labellings rather than better handling of boundaries, we perform an additional evaluation for the non-boundary regions only. We define the boundary region as a narrow band around the ground truth label changes, and the width of this band is set to 15 pixels. Non-boundary region refers to rest of the image. Table 2 shows that compared with the baseline, our method performs around 6% and 3% better in mean class accuracy and mean class IOU for the actor-action settings respectively. In addition, the improvements are more notable for action categories compared with actor categories, which is expected as our method is designed for resolving labeling inconsistency issue that is more server in action categories rather than actor categories.

**Influence of the quality of region masks:** We train two FCIS models on A2D and COCO datasets respectively for actor region mask generation. The generated masks are fed

region masks (training)	region masks (testing)	global pixel accuracy			mean class accuracy			mean class IoU		
		actor	action	actor-action	actor	action	actor-action	actor	action	actor-action
COCO + GT	COCO	94.7	93.0	92.6	82.4	69.4	54.1	65.1	50.9	37.9
A2D + GT	A2D	<b>95.3</b>	<b>93.4</b>	<b>93.0</b>	<b>86.0</b>	<b>70.7</b>	<b>56.4</b>	<b>68.1</b>	<b>51.1</b>	<b>38.6</b>
A2D + GT	GT	98.1	96.3	95.9	87.7	72.5	57.7	80.9	59.1	45.6

Table 3: Effects of actor region masks quality. Region masks generated from FCIS model trained on COCO and A2D datasets are denoted as "COCO" and "A2D" respectively. Ground truth region masks are denoted as "GT". The top two rows show the results using FCIS generated masks, while the last row shows the results when ground truth region masks are used during the testing.

	global pixel accuracy			mean class accuracy			mean class IoU		
	actor	action	actor-action	actor	action	actor-action	actor	action	actor-action
	-			47.4	49.4	30.5	-		
	85.2	85.3	84.2	61.2	60.5	43.9	-		
	83.8	83.1	81.7	-	-	41.7	-		
	90.6	89.3	88.7	73.7	61.4	48.0	49.5	42.2	29.7
	-	-	<b>93.0</b>	60.0	59.9	45.0	-	-	33.4
Ours	<b>95.3</b>	<b>93.4</b>	<b>93.0</b>	<b>86.0</b>	<b>70.7</b>	<b>56.4</b>	<b>68.1</b>	<b>51.1</b>	<b>38.6</b>

Table 4: Comparison with the previous works. When multiple results are given, the best one for each metric is displayed.

as inputs to our segmentation network at the training or testing time. As suggested by , additionally we use ground truth actor region masks during the training, as they may be important for small objects that are not tightly covered by FCIS generated region masks.

The top two rows of Table 3 show the results of our method using FCIS generated actor region masks at the testing time. We find that our approach using region masks from the FCIS model trained on A2D consistently performs better than COCO in all the three metrics, especially in the actor setting with performance improvement of around 4% in mean class accuracy and 3% in mean class IOU. These results demonstrate that high-quality, dataset-specific candidate region masks contribute to the performance improvements.

The last row shows the results when ground truth actor region masks are used at the testing time. It is an upper bound performance by assuming perfect instance segmentation results, and hence the errors can only be attributed to incorrect actor-action semantic segmentation. While the improvements are very significant compared with FCIS generated masks (*e.g.*, more than 7% in mean class IOU for all the three settings), the performance may still be unsatisfactory (*e.g.*, less than 50% in the mean class IOU for the actor-action setting). Therefore, it is useful to research on both actor-action semantic segmentation and instance segmentation methods to further improve the results.

**Influence of incorporating optical flow:** From Table 5, we see optical flow can significantly improve the performance of some fast-moving actor-action categories. For example, the per-category accuracies of adult-jumping, car-running, ball-flying, cat-running, and dog-jumping are increased by more than 10%. On average, our method with optical flow can improve the performance in both mean class accuracy and mean class IOU by more than 4% in the actor-action setting (see Table 1). These results suggest optical flow can effectively help improve our performance.

**Comparison with previous works:** Table 4 shows the results of our method and earlier works. Our approach outperforms the state-of-the-art techniques by more than 8% in mean class accuracy and more than 5% in mean class IOU. It should be noted that the network architecture and pretrained model used in our method is different from the ones used by

	adult										baby					
method	BG	climbing	crawling	eating	jumping	rolling	running	walking	none	climbing	crawling	rolling	walking	none		
Baseline (RGB only)	98.1	49.0	71.9	91.7	22.9	34.1	31.0	61.7	51.1	68.8	59.3	74.1	55.6	16.4		
Baseline (RGB + flow)	<b>98.2</b>	53.4	77.7	89.9	31.5	36.5	32.2	<b>64.6</b>	51.5	69.3	69.2	73.5	58.7	20.3		
Ours (RGB only)	96.2	<b>76.7</b>	76.3	<b>95.5</b>	27.6	<b>59.1</b>	49.5	56.4	62.2	83.0	75.3	<b>88.6</b>	57.5	8.9		
Ours (RGB + flow)	96.4	73.2	<b>79.7</b>	93.1	<b>43.0</b>	48.1	<b>53.0</b>	61.1	<b>66.1</b>	<b>83.3</b>	<b>77.3</b>	86.3	<b>61.0</b>	<b>37.0</b>		
	ball				bird						car					
method	flying	jumping	rolling	none	climbing	eating	flying	jumping	rolling	walking	none	flying	jumping	rolling	running	none
Baseline (RGB only)	0.8	2.6	49.5	6.5	57.9	45.9	60.7	5.6	43.5	46.0	25.4	28.6	80.2	42.9	44.8	50.6
Baseline (RGB + flow)	2.0	7.6	53.9	12.8	62.4	45.5	61.1	9.9	46.6	55.0	<b>28.5</b>	29.2	84.6	45.5	56.6	46.3
Ours (RGB only)	5.4	28.6	65.8	12.6	<b>68.3</b>	48.6	<b>81.6</b>	<b>29.7</b>	57.9	54.5	3.0	<b>53.0</b>	94.3	58.9	56.2	37.5
Ours (RGB + flow)	<b>26.7</b>	<b>30.8</b>	<b>81.1</b>	<b>18.5</b>	63.2	<b>50.5</b>	80.2	22.2	<b>65.0</b>	<b>61.9</b>	3.0	34.9	<b>97.9</b>	<b>62.3</b>	<b>75.7</b>	<b>62.9</b>
	cat								dog						Avg	
method	climbing	eating	jumping	rolling	running	walking	none	crawling	eating	jumping	rolling	running	walking	-		
Baseline (RGB only)	48.8	65.5	19.2	51.1	14.5	41.5	4.4	42.2	80.2	5.5	38.9	6.8	71.8	42.7		
Baseline (RGB + flow)	53.1	64.7	15.8	49.2	20.3	52.3	<b>8.1</b>	<b>51.9</b>	<b>82.9</b>	13.2	47.8	8.4	<b>74.8</b>	46.1		
Ours (RGB only)	58.2	74.6	<b>32.8</b>	<b>82.1</b>	34.6	37.0	4.0	46.1	77.9	8.8	48.4	19.5	70.3	51.5		
Ours (RGB + flow)	<b>62.8</b>	<b>75.7</b>	30.9	75.7	<b>46.1</b>	<b>49.4</b>	6.7	49.2	81.3	<b>35.7</b>	<b>61.9</b>	<b>27.0</b>	71.0	<b>56.4</b>		

Table 5: Per-category accuracy result.

[13, 23, 28, 29, 30]. Hence, to validate the effectiveness of our method, we compare it with DeepLab baseline and conduct an ablation study (see Tables 1, 2, 3 and 5). These are carried out in controlled settings to ensure a fair comparison. The results demonstrate that specific designs (exploiting region masks, optical flow, and quality of region masks) in our method contribute to the performance gains.

**Computational complexity:** Our testing platform is a server with Intel Xeon 2.4GHZ CPU and Nvidia K80 GPU. We limit the hardware usage to single CPU and GPU. The optical flow computation using FlowNet [8], region mask calculation using FCIS [63] and our region-based segmentation network inference take around 350ms, 300ms and 450ms for each frame, respectively. Hence the overall computational time of the proposed approach is around 1100ms per frame. In contrast, the overall testing time of the baseline DeepLab-type segmentation approach is around 395ms per frame inclusive of optical flow computation. We note that certain components of our network, *e.g.*, region-to-pixel layer, could be further optimized with a native CUDA implementation. As a future direction, we are also interested in integrating actor-action semantic segmentation with optical flow estimation and region mask generation into one unified deep network for better efficiency and accuracy.

## 5 Conclusion

In this paper, we propose an end-to-end region-based actor-action segmentation approach. We design a deep network consisting of a two-stream front-end and a region-based actor-action segmentation back-end. The two-stream front-end learns two types of features: one for appearance and the other for motion. The fused features from the two-stream front-end together with region masks are passed to the back-end for actor-action labeling. We validate the effectiveness of our approach on the A2D dataset. Experiment results show that both the region-based segmentation strategy and fused features from the two-stream front-end contribute to performance improvements.

## References

- [1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.
- [2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 2011.
- [3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Region-based semantic segmentation with end-to-end training. In *ECCV*, 2016.
- [4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.
- [5] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. *arXiv:1712.04837*, 2017.
- [6] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.
- [7] Kang Dang, Jiong Yang, and Junsong Yuan. Adaptive exponential smoothing for online filtering of pixel prediction maps. In *ICCV*, 2015.
- [8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv:1703.06870*, 2017.
- [11] Yang He, Wei-Chen Chiu, Margret Keuper, and Mario Fritz. Std2p: Rgb-d semantic segmentation using spatio-temporal data-driven pooling. In *CVPR*, 2017.
- [12] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017.
- [13] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Joint learning of object and action detectors. In *ICCV*, 2017.
- [14] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv:1511.02680*, 2015.
- [15] Anna Khoreva, Federico Perazzi, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. *arXiv:1612.02646*, 2016.
- [16] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016.

- [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [18] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 7, page 43, 2017.
- [19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [20] Adam Paszke, Soumith Chintala, Ronan Collobert, Koray Kavukcuoglu, Clement Farabet, Samy Bengio, Iain Melvin, Jason Weston, and Johnny Mariethoz. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, may 2017.
- [21] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016.
- [22] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NIPS*, 2015.
- [23] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Transactions on Multimedia*, 2018.
- [24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [25] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV Workshops*, 2016.
- [26] Z. Tu, W. Xie, J. Dauwels, B. Li, and J. Yuan. Semantic cues enhanced multi-modality multi-stream cnn for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.
- [27] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.
- [28] Chenliang Xu and Jason J Corso. Actor-action semantic segmentation with grouping process models. In *CVPR*, 2016.
- [29] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015.
- [30] Yan Yan, Chenliang Xu, Dawen Cai, and Jason Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. In *CVPR*, 2017.
- [31] Jiong Yang and Junsong Yuan. Common action discovery and localization in unconstrained videos. In *ICCV*, 2017.
- [32] Jiong Yang, Brian Price, Xiaohui Shen, Zhe Lin, and Junsong Yuan. Fast appearance modeling for automatic primary video object segmentation. *IEEE Transactions on Image Processing*, 2016.
- [33] Jifeng Dai Xiangyang Ji Yi Li, Haozhi Qi and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.
- [34] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv:1511.07122*, 2015.
- [35] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *CVPR*, 2015.