We don't Need Thousand Proposals: Single Shot Actor-Action Detection in Videos (Supplementary Material)

Aayush J Rana
aayushjr@knights.ucf.edu

Yogesh S Rawat
yogesh@crcv.ucf.edu

Center for Research in Computer Vision
University of Central Florida
Orlando, FL, 32816

1. Network details

The proposed SSA2D network architecture takes in a video clip and predicts pixel-wise actor and action classes using an encoder-decoder based model. Since the network has multiple blocks, we show each of the major block here with their technical details.

1.1. Encoder block

The encoder block is the first part of the architecture. This block is responsible for feature extraction from a given clip and is shared by all future decoder blocks. As shown in Figure 1, the input clip is passed through multiple convolutional layers. In the paper we use I3D as our backbone network for encoding clip features. Hence, we follow the same layer order. From Figure 1, *Conv1* and *Conv2* are convolutional layers with same kernel size and strides as the I3D network. *Conv3*, *Conv4* and *Conv5* use the inception configuration with 2, 5 and 2 inception blocks respectively. We change the pooling strides and kernel sizes accordingly to get final output of $\frac{T}{4} \times \frac{H}{16} \times \frac{W}{16}$ from an input clip of $T \times H \times W$. We take skip connection after *Conv2* layer and *Conv3* layer, which is passed to each of the decoder block accordingly.

1.2. Decoder block

The decoder block is used all three branches (*STU-Mask* detection, actor detection, action detection) of the proposed network. The purpose of decoder block is to take encoded features and produce detection masks accordingly. All three branches use identical decoder block. Only the final output layer's channel is adjusted according to the desired branch output. As A2D only provides annotation for single frame, loss for action detection is only computed for single frame. As such, decoder block in actor detection is configured to predict only single frame output. However, the object features going into the OFI block does not get affected by this configuration. Since ViDOR dataset has per frame annotation, the decoder block outputs all frames predictions. Input to the block is encoded features from the encoder block. This is passed through deconvolution layers which will perform convolution as well as upsampling of the layers to increase the size. The operation is costlier than only convolution, so we only implement two deconvolution layers. To help retain features, we add skip connection from the encoder block. *Skip connection 2* is concatenated with *Deconv1* features and *skip connection 1* is concatenated with *Deconv2* features. The output feature size of *Deconv2* layer is adjusted to be $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4}$, which is temporally half and spatially one-fourth of the input resolution. This was done to keep the network smaller and improve efficiency. On a larger memory GPU this can be further increased to original input resolution, which results to finer segmentation output. We apply dilated (atrous) convolution to capture features at multiple receptive fields (rate=3,6,9,12). Following feature pyramid network, we also take features after *Deconv1* and *Deconv2* layer and upsample it *4*$\times$ and *2*$\times$ respectively, which is then concatenated to features from dilated convolution.

1.3. Network size

We also evaluate our network's size in terms of training parameters. Our RGB only model has $\sim$35M parameters considering we have 3 decoder branches, while our RGB+OF model has $\sim$55M parameters. Considering ResNet-101 alone has $\sim$44M parameters [4] and since prior works [1, 2] use ResNet-101 as encoder backbone for each of their RGB and OF stream, they will have $\sim$88M training parameters even without additional parts of their network (decoder, RPN, segmentation). This further demonstrates why our network can be trained in a single 12 GB GPU and

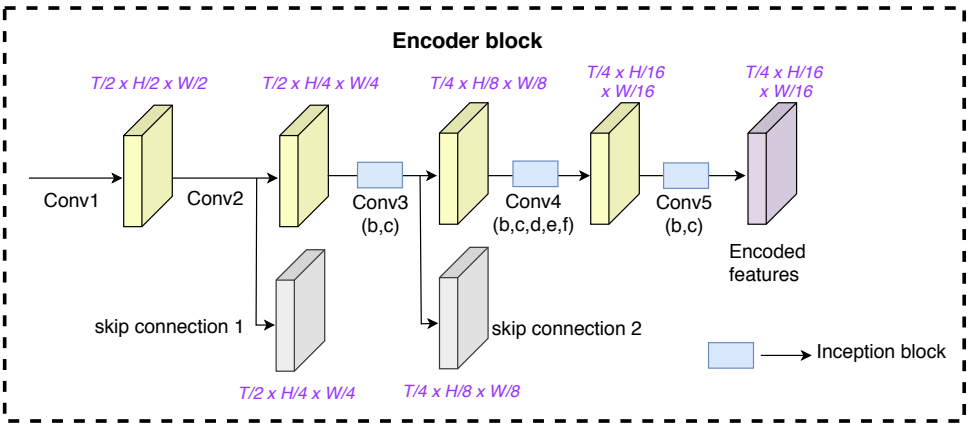

Figure 1. Encoder block details. The encoder contains multiple inception blocks to extract relevant features for decoder network. To retain fine grained features from initial layers, two skip connections are passed to the decoder network, which helps in fine grained pixel-level detection.

has faster inference time.

2. Results

Here we provide some more results on A2D dataset and VidOR dataset for the experiments reported in main manuscript.

2.1. Actor-Action per class accuracy

For a better understanding of class-wise detection, we look at the actor-action per class accuracy score. From the A2D dataset experiment reported on main manuscript, we provide a detailed per class average accuracy score using our RGB+OF model in Figure 2 and 3. We report the result on all 7 actor classes and 9 action classes along with background class for no actor/action region.

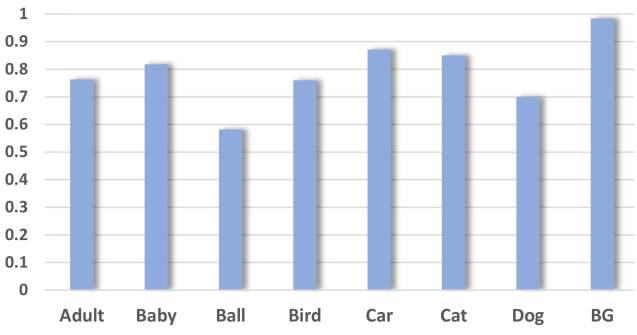

Figure 2. Per actor class average accuracy score for A2D. The network is gets high accuracy for most classes except ball. This is due to the low number of training samples for ball and small blurry region for a moving ball. The background class is for all pixels where no actor is present.

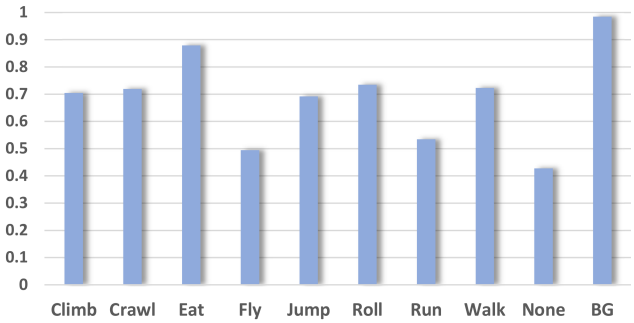

Figure 3. Per action class average accuracy score for A2D. Here, *None* class means an actor is present but it is doing no action. The background class is for all pixels where no actor/action is present.

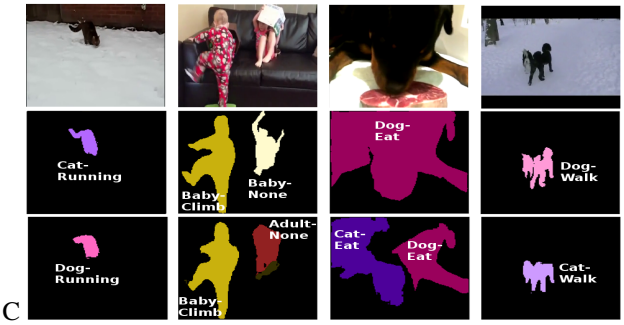

Figure 4. Qualitative analysis of some failure cases. The top, middle and bottom row represents input key frame, ground truth semantic segmentation mask and our joint actor-action detection predictions with label respectively.

2.2. Per class IoU

We also evaluate the IoU scores per class and observe the classes for which our method has issues in.

		adult								*baby*				
Modality	BG	climbing	crawling	eating	jumping	rolling	running	walking	none	climbing	crawling	rolling	walking	none
RGB	98.5	58.6	61.6	91.3	37.4	41.4	46.1	66.4	49.5	56.8	69.4	68.1	47.8	36.6
RGB+OF	98.5	68.2	65.7	92.8	54.9	37.7	49.6	64.1	48.8	70.3	78.1	77.6	65.9	23.1

	ball				*bird*							*car*				
	flying	jumping	rolling	none	climbing	eating	flying	jumping	rolling	walking	none	flying	jumping	rolling	running	none
RGB	14.9	21.2	70.9	15.3	56.2	49.9	64.1	26.5	46.2	35.1	19.7	38.1	88.6	60.9	79.3	30.9
RGB+OF	17.2	33.6	77.3	19.1	63.9	51.6	66.8	48.6	52.8	55.9	29.1	36.7	90.5	61.1	74.1	30.4

	cat							*dog*							*Avg*
	climbing	eating	jumping	rolling	running	walking	none	crawling	eating	jumping	rolling	running	walking	none	
RGB	54.1	78.9	22.6	73.5	40.6	55.9	13.9	44.5	66.4	21.2	45	29	63.9	7.6	49.3
RGB+OF	63.9	90.1	27.3	81.5	44.6	67.7	14.1	53.7	68.1	28.9	45.5	28.1	79.1	11.3	54.7

Table 1. Per class accuracy scores on A2D dataset using RGB only and RGB+OF variants. We observe that there are more classes where having explicit motion information improves the performance.

bite	caress	carry	chase	clean	close	cut	drive	feed	get off	get on	grab	hit
24.04	8.97	9.12	9.19	4.54	1.01	0.93	9.02	1.68	0.65	0.49	1.47	1.25

hold	hold hand of	hug	kick	kiss	knock	lean on	lick	lift	open	pat	play (instrument)	point to
19.08	8.68	24.88	1.38	5.13	0.52	24.05	1.29	9.48	13.40	0.64	17.86	1.57

press	pull	push	release	ride	shake hand with	shout at	smell	speak to	squeeze	throw	touch	use
2.76	8.54	7.68	2.71	19.10	9.12	0.49	4.65	15.34	0.46	0.72	1.66	4.14

watch	wave	wave hand to	Average									
42.02	3.56	7.04	7.9									

Table 2. Per class action IoU score on VidOR dataset.

A2D: From the A2D dataset experiment reported on main manuscript, we provide a detailed per class actor-action IoU score using our RGB only and RGB+OF model in Table 1. We report the result on all valid 43 actor-action pairs in this table.

VidOR: We also evaluate the per class IoU score for VidOR dataset. We report the 42 action classes IoU score. The ground truth annotations are bounding boxes which causes extra noise to be added for fine pixel-level detection task. In contrast, A2D provides pixel-wise semantic segmentation which is more fine-grained than bounding box and has fewer noise in the labels. We use the bounding box annotations provided as ground truth and let the network learn the segmentation. We observe that the network starts to reduce noise and learn a more accurate detection instead of the bounding box. This causes the IoU score to be always lower as it never matches the bounding box ground truth. We convert each detection instance into the minimum enclosed bounding box for a reasonable comparison with the ground truth bounding box. Table 2 shows the per class action IoU score we observed for VidOR dataset. Besides the immense data imbalance in actor and action classes, the activities in VidOR are active (*watch, push, open, get on, close*) and passive (*watch, speak to, lean on, shout at*) which makes detecting action based on motion challenging task.

2.3. Self-attention

Learning the local and global spatio-temporal relation between features in the entire video is challenging with small kernel size, therefore we utilize self-attention [3] to learn short and long range dependencies between the features, providing a better contextual understanding. The self-attention module SA takes the spatio-temporal features from encoder block f_{enc1} as input and learns $f_{enc} \in \mathbb{R}^{\frac{T}{4} \times \frac{H}{16} \times \frac{W}{16}}$ which has a better understanding of spatio-temporal relations. We notice that the self-attention module helps in overall action detection task.

2.4. Atrous(dilated) convolution

Since atrous convolution helps encapsulate features through multiple receptive fields, we expect it to help in improving detection of objects which are too large or too small. We remove atrous convolutions from all decoder blocks in the model and evaluate the model to see its importance. We observe network performance drops without atrous convolution as this contextual information is missing.

2.5. Multi-scale

Another component in our model is the multi-scale feature concatenation from different upsampling layers, similar to FPN. Features of a region from different granularity

Figure 5. Qualitative results for A2D using our network showing the keyframe from input video clip, ground truth with actor-action annotations and SSA2D predicted output with actor-action labels respectively.

boosts network performance as key features are highlighted. We observe that this component has less impact on the network compared to atrous convolution for multi-scale feature encoding.

2.6. RGB vs RGB+Optical Flow Analysis

We observe how our model performs per class in Table 1, where the per class accuracy score is shown for both of our RGB model and RGB+OF model. It is seen that the proposed method can detect most classes accurately in RGB model and the scores are further increased with flow model. The explicit motion information helps improve the score overall in most of the classes.

2.7. Failure cases

In Figure 4 we have shown some of the failure cases of our method. We observe that the network is able to detect correct foreground region in most cases, however, it gets confused on similar actors such as dog-cat or adult-baby. The approach suffers from data imbalance so classes with lower samples will perform lower which is observed in the prior works as well. This is one of the limitation of our approach which can be improved in future works.

2.8. Qualitative results

We provide additional qualitative results for A2D dataset in Figure 5 predicted using our network. A2D has fine pixel-level class labels in ground truth, so the network pre-

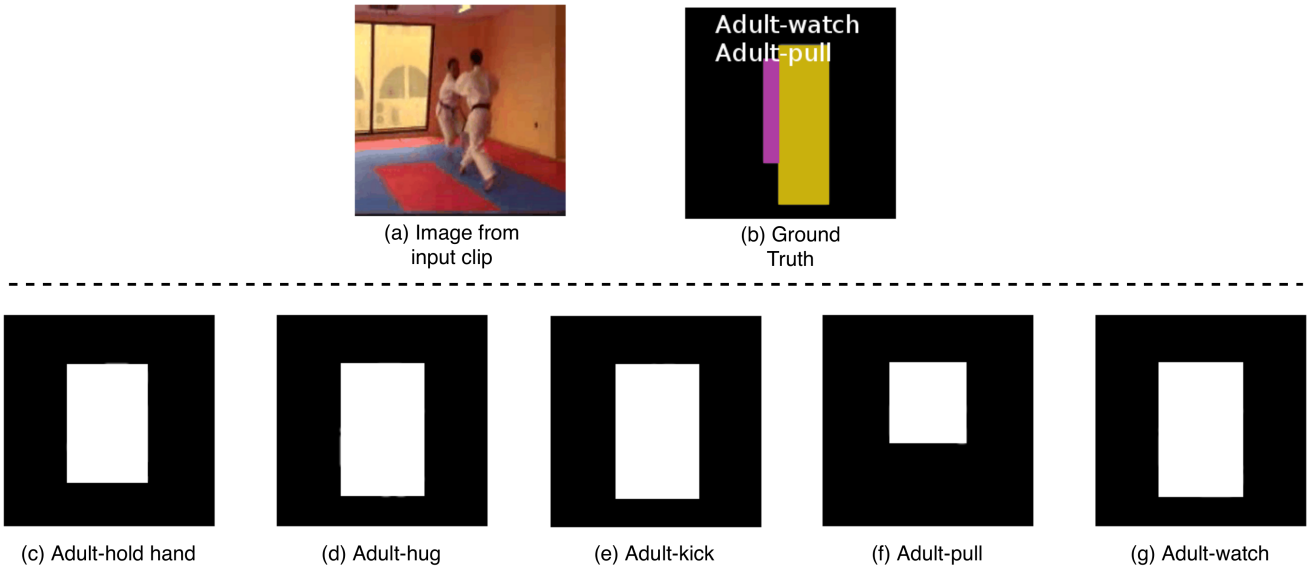

Figure 6. Qualitative results on VidOR dataset using our network. (a) Sample image from an input clip from the VidOR dataset. (b) Ground truth labels with bounding box annotation for actor-action pairs. (c-g) Predicted actor-action pairs from our network. We can see that the network is able to detect more actions related to the pair than present in ground truth.

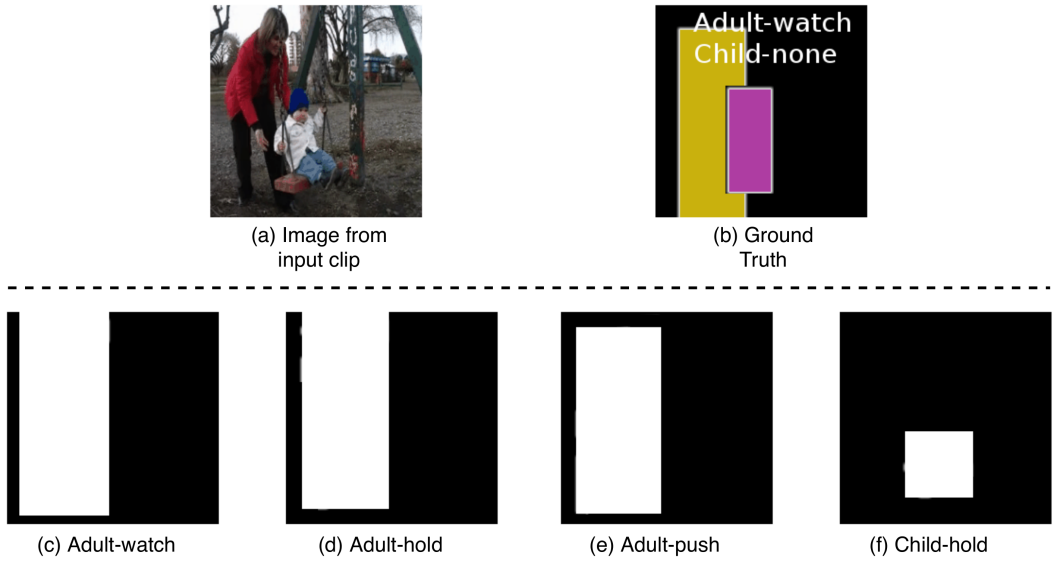

Figure 7. Qualitative results on VidOR dataset using our network. (a) Sample image from an input clip from the VidOR dataset. (b) Ground truth labels with bounding box annotation for actor-action pairs. (c-f) Predicted actor-action pairs from our network. We observe that the network predicts relations correctly even if it is missing from ground truth.

dicts precise edges for actor-action pairs.

We provide VidOR dataset results in Figure 6 and 7. VidOR has bounding box ground truth labels, so the network cannot learn precise boundary as loss calculation is done on entire bounding box. Figure 6 shows a sample clip where two adults are performing a martial arts move. The ground truth labels for this clip are *Adult-watch* and *Adult-pull*. The network predicts *Adult-watch* correctly as seen in Figure 6(g) and predicts partial *Adult-pull* in Figure 6(f) as only that region is related to pull action. The network also predicts *Adult-hold hand, Adult-hug* and *Adult-kick* in Figure 6 (c, d, e) respectively for that region as those actions also seem correct for the given sequence.

Similarly Figure 7 shows a sample clip from VidOR where an adult is pushing a child on a swing. The ground truth labels in Figure 7(b) shows labels for *adult-watch*. The

objects in the clip are *adult* and *child*, where the *child* object has no action associated with it in the ground truth. The network correctly detects *adult-watch* in Figure 7(c). Furthermore, the network also predicts *adult-hold* and *adult-push* actions in Figure 7(d),(e); both of which seem correct for the given clip sequence. It also predicts *child-hold* in Figure 7(f) which is a valid label from the dataset but missing from the ground truth annotation.

References

[1] Kang Dang, Chunluan Zhou, Zhigang Tu, Michael Hoy, Justin Dauwels, and Junsong Yuan. Actor-action semantic segmentation with region masks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[2] Jingwei Ji, Shyamal Buch, Alvaro Soto, and Juan Carlos Niebles. End-to-end joint semantic segmentation of actors and actions in video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 702–717, 2018.

[3] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[4] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.