Supplementary Material: Asynchronous Interaction Aggregation for Action Detection

Jiajun Tang*, Jin Xia*, Xinzhi Mu, Bo Pang, and Cewu Lu†

Shanghai Jiao Tong University, China
`yelantingfeng@sjtu.edu.cn, ga.xiajin@gmail.com,`
`{draconids, pangbo, lucewu}@sjtu.edu.cn`

1 Implementation Details on UCF101-24

Backbone. In UCF101-24 experiments, we utilize two different video backbone models, C2D and I3D, to further validate the effectiveness of our method. The specific designs of both backbones are basically the same as those in [7] except that all temporal max pooling operations are removed from C2D. Both backbones are with ResNet50 and we show the detailed specification of the modified C2D backbone in Table 1. For C2D backbone, we take one single RGB frame as both the input and the target frame. For I3D backbone, the input is 16 consecutive frames and the training targets (bounding boxes with action labels) are only from the center (9-th) frame.

Instance Detector. We basically follow the AVA setting to prepare our instance detectors for UCF101-24. The person detector is first pre-trained on MSCOCO [5] dataset and then fine-tuned on the UCF101-24 dataset for higher recall. The object detector is the same as that used in AVA experiments, i.e., adopted from maskrcnn-benchmark Model Zoo [6].

Training and Inference. During training, we use ground truth boxes as positive samples and those detected boxes overlapping with ground truth boxes by IOU less than 0.3 are considered as negative ones. We use detected person boxes with confidence score larger than 0.8 in training. During inference, all detected person boxes are used, the final confidence score of each person target is given by multiplying corresponding human detection score with action score.

---

*Both authors contributed equally to this work.

†Cewu Lu is the corresponding author, he is also a member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China.

Table 1. The modified ResNet50 C2D backbone. We show the specification used in Kinetics pre-training. For Kinetics, the input video size is set as $32 \times 224 \times 224$. All convolution layers are shown as the kernel size $H \times W$ followed by the channel number. In $\text{res}_3, \text{res}_4, \text{res}_5$, the first convoluation layer has a stride 2 to downsample the feature map

stage	specification	output size
conv_1	$7 \times 7, 64$, stride 2, 2	$32 \times 112 \times 112$
pool_1	3×3 max, stride 2, 2	$32 \times 56 \times 56$
res_2	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$32 \times 56 \times 56$
res_3	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$32 \times 28 \times 28$
res_4	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$32 \times 14 \times 14$
res_5	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$32 \times 7 \times 7$
	global average pooling, fc	$1 \times 1 \times 1$

2 Implementation Details on EPIC-Kitchens

Backbone. For EPIC-Kitchens dataset, we use ResNet-50 SlowFast Network [2] as our video backbone. This backbone is the same as what we use in AVA experiments. It is pre-trained on Kinetics-700 dataset and then fine-tuned as the baseline in EPIC-Kitchens experiments.

Instance Detector. Since the videos are egocentric and there is no presence of complete human but hands, we train a hand detector to provide person boxes. The detector model is the same as what we use in AVA experiments, but is trained with different dataset. The detector is first pre-trained on Visual Genome [3] dataset. For hand detection, we fine-tune the detector with EGTEA [4] dataset. For object detection, we fine-tune the detector using EPIC-Kitchens object annotations.

Training and Inference. The task of EPIC-Kitchens is to recognize noun, verb, action categories given the segment. Following [1], we split the original training set into a new training set and a new validation set. Verb models and noun models are trained separately. Actions are obtained by combining their predictions. During training, we randomly select a video segment of one second in annotated segment as video input. During inference, the center one second of given segment is passed for the prediction. Hand features and object features are cropped and fed into IA to model person-person and person-object interactions.

The temporal memory is different for verb model and noun model. For the verb model, the memory consists of query features extracted by the video backbone. Some features could be missing since not all segments are chosen for training. The missed features are re-estimated at the beginning of each epoch. For the noun model, we follow [8] to construct an *object-centric* feature bank, so the memory are frozen object RoI features extracted from the object detector's feature maps.

References

1. Baradel, F., Neverova, N., Wolf, C., Mille, J., Mori, G.: Object level visual reasoning in videos. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 105–121 (2018)
2. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6202–6211 (2019)
3. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 32–73 (2017)
4. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 619–635 (2018)
5. 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)
6. Massa, F., Girshick, R.: maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark` (2018), accessed: 2020-2-29
7. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
8. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-term feature banks for detailed video understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)