Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization

1st place solution for AVA-Kinetics Crossover in AcitivityNet Challenge 2020

Siyu Chen[2*] Junting Pan[1*] Guanglu Song[2] Manyuan Zhang[2]
Hao Shao[2] Ziyi Lin[1] Jing Shao[2] Hongsheng Li[1] Yu Liu[2]
[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]SenseTime Research

Abstract

This technical report introduces our winning solution to the spatio-temporal action localization track, AVA-Kinetics Crossover, in ActivityNet Challenge 2020. Our entry is mainly based on Actor-Context-Actor Relation Network [14]. We describe technical details for the new AVA-Kinetics dataset, together with some experimental results. Without any bells and whistles, we achieved ***39.62 mAP*** *on the test set of AVA-Kinetics, which outperforms other entries by a large margin. Code will be available at: https://github.com/Siyu-C/ACAR-Net.*

1. Method

Our approach features Actor-Context-Actor Relation Network (ACAR-Net), details of which can be found in [14]. Our proposed ACAR-Net gives an efficient yet effective algorithm to explicitly model and utilize higher-order relations built upon the basic first-order actor-context relations for assisting action localization.

1.1. Overall Framework

We first introduce our overall framework for action localization, where our proposed ACAR-Net is its key module for high-order relation modeling. The framework is designed to detect all persons in an input video clip and predict their action labels.

We combine an *off-the-shelf person detector* (*e.g.* Faster R-CNN [15]) with a *video backbone network* (*e.g.* I3D [3]). In details, the detector operates on the center frame (*i.e.* key frame) of the clip and obtains N detected actors. Such detected boxes are duplicated to other frames of the clip. In the mean time, the backbone network extracts a spatio-temporal feature volume from the input video clip. We perform average pooling along the temporal dimension considering computational efficiency, which results in a feature map $V \in \mathbb{R}^{C \times H \times W}$, where C, H, W correspond to channel, height and width respectively. We apply RoIAlign [8] (7×7 spatial output) followed by spatial max pooling to the feature map V and the N actor boxes, producing a series of N actor features, $A_1, A_2, \cdots, A_N \in \mathbb{R}^C$. Each actor feature describes the spatio-temporal appearance and motion of one Region of Interest (RoI).

The final classification head takes the aforementioned video feature map V and RoI features $\{A_i\}_{i=1}^N$ as inputs, and outputs the final action predictions possibly after relation reasoning. The simplest baseline is a ”Linear” head, which directly applies a linear classifier to the RoI features.

1.2. Actor-Context-Actor Relation Network

We first encode the first-order actor-context relations between each actor and each spatial location of the spatio-temporal context. More specifically, it concatenates each actor feature $A_i \in \mathbb{R}^C$ to all $H \times W$ spatial locations of the context feature $V \in \mathbb{R}^{C \times H \times W}$ to form a concatenated feature map $F'_i \in \mathbb{R}^{2C \times H \times W}$. The actor-context relation feature F^i for actor i can then be computed by applying convolutions to this concatenated feature map.

Based on the actor-context relations, we further add a **High-order Relation Reasoning Operator** (HR^{2}O) for modeling the connections established on first-order relations, which are indirect relations mostly ignored by previous methods. Let $F^i_{x,y}$ record the first-order feature between the actor A_i and the scene context V at the spatial location (x, y). We introduce *High-order Relation Reasoning*, in order to explicitly model the relations between first-order actor-context relations, which encode more informative scene semantics. However, since there are a large number of actor-context relation features, $F^i_{x,y}, i \in \{1, \cdots, N\}, x \in [1, H], y \in [1, W]$, the number of their possible pairwise combinations are generally overwhelming. We therefore propose to focus on learning the high-order relations between different actor-context relations at the same spatial location (x, y), i.e. $F^i_{x,y}$ and $F^j_{x,y}$. In this way, the proposed relational reasoning operator limits the relation learning to second-order actor-context-

[*] Equal contribution

actor relations, *i.e.* two actors i and j can be associated via the same spatial context as $i \leftrightarrow (x, y) \leftrightarrow j$ to help the prediction of their action labels.

In general, our high-order relation reasoning block ACAR-Net is weakly-supervised, which only requires action labels as supervision.

Instantiation. We implement the High-order Relation Reasoning Operator HR^{2}O as a location-wise attention operator, which is natural for modeling the connections between multiple first-order relations at the same spatial location. The operator HR^{2}O$_{\text{NL}}$ consists of one or two modified non-local blocks [17]. Since we are operating on a spatial grid of features, we replace the fully-connected layers in the non-local block with convolutional layers, and the attention vector is computed separately at every spatial location. Following [18], we also add layer normalization and dropout to our modified non-local block for improving regularization.

$$
\begin{aligned}
Q^i &= \text{Conv2D}_{\text{q}}(F^i) \\
K^j &= \text{Conv2D}_{\text{k}}(F^j) \\
V^j &= \text{Conv2D}_{\text{v}}(F^j) \\
S^{i,j}_{x,y} &= \left\langle Q^i_{x,y}, K^j_{x,y} \right\rangle / \sqrt{d} \\
A^i_{x,y} &= \text{Softmax}_j(S^{i,j}_{x,y}) \\
W^i_{x,y} &= \text{ReLU}\left(\text{norm}\left(\sum_j A^{i,j}_{x,y} V^j_{x,y}\right)\right) \\
H^i &= \text{dropout}(\text{Conv2D}_{\text{f}}(W^i))
\end{aligned}
$$

For saving memory, spatial 2×2 max pooling is applied by default to the first-order relation maps before feeding them into our operator. The high-order relation map H^i will be spatially average-pooled, and then channel-wise concatenated to the basic actor RoI feature vector A_i for final classification. All relation vectors are of dimension $d = 512$ in our implementation.

1.3. Actor-Context Feature Bank

Inspired by the Long-term Feature Bank (LFB) [18], which creates a feature bank over a large time span to facilitate first-order actor-actor relation reasoning, we consider creating an **Actor-Context Feature Bank** F_{bank} which is built upon the first-relation features computed in our ACAR-Net. Formally, $F_{\text{bank}} = [F_0, F_1, \cdots, F_{T-1}]$, where F_t is the first-order actor-context relation map extracted from a short video clip around time t. This bank of features can be obtained by running a trained ACAR-Net over the entire video at evenly spaced intervals (by default 1 second) and saving the intermediate first-order relation maps. Different from the original LFB, our relational feature preserves spatial context information. Equipped with such a relational feature bank, our ACAR-Net can leverage its High-order Relation Reasoning Operator for reasoning actor-context-actor relations over a much longer time span, and thus better capture what is happening in the entire video for achieving more accurate action localization at the current time stamp.

Instantiation. We experiment on ACFB with the aforementioned HR^{2}O$_{\text{NL}}$ implementation of high-order relation reasoning in ACAR-Net. We stack two modified non-local blocks, and replace the self-attention mechanism with an attention between current and long-term actor-context relations. For AVA videos, we set the long-term time span to ∼20 seconds, and for a Kinetics video, the bank simply spans across the entire video whose length is at most 10 seconds. For faster convergence, we do not apply spatial max pooling before HR^{2}O$_{\text{NL}}$.

2. Experiments

2.1. Implementation Details

Dataset. For this year's challenge, Kinetics-700 [2] videos with AVA [7] style annotations are introduced. The new AVA-Kinetics dataset [11] of spatio-temporally localized atomic visual actions contains over 238k unique videos and more than 624k annotated frames. For AVA, box annotations and their corresponding action labels are provided on key frames of 430 15-minute movie clips with a temporal stride of 1 second, while for Kinetics only a single frame is annotated for each video. Following the guidelines of the challenge, we evaluate on 60 action classes, and the performance metric is mean Average Precision (mAP) using a frame-level IoU threshold of 0.5.

Person Detector. As for person detector on key frames, we adopted the detection model from [13], which is a Faster R-CNN [15] with an SENet-154-FPN-TSD [9, 12, 16] backbone. The model is pre-trained on OpenImage [10], and then fine-tuned on AVA and Kinetics respectively. The final models obtain 95.8 AP@50 on the AVA validation set, and 84.4 AP@50 on the Kinetics validation set.

Backbone Network. We use SlowFast networks [4] as the backbone in our localization framework, and we also increase the spatial resolution of res$_5$ by 2×. We use SlowFast R101 and R152 instantiations with input sampling $T \times \tau = 8 \times 8$ and 16×8 (without non-local) pre-trained on the Kinetics-700 dataset [2].

Training. We use per-class binary cross entropy loss as the training loss function. Since one person should only have one pose label, following [19], we apply a softmax function instead of sigmoid to the logits corresponding to pose classes.

We train all models in an end-to-end fashion (except the feature bank part) using synchronous SGD with a minibatch

model	head	3S+F	val mAP	test mAP
SlowFast, R101, 8 × 8	Linear	×	32.98	-
SlowFast, R101, 8 × 8	ACAR		34.58	-
SlowFast, R152, 8 × 8	ACAR		35.12	34.99
SlowFast, R101, 8 × 8	ACFB		**35.84**	-
SlowFast, R101, 8 × 8	Linear	✓	33.96	-
SlowFast, R101, 8 × 8	ACAR		35.44	-
SlowFast, R152, 8 × 8	ACAR		35.96	-
SlowFast, R101, 8 × 8	ACFB		$\underline{36.36}$	-
SlowFast, ensemble	mixed		**40.49**	**39.62**

Table 1: **AVA-Kinetics v1.0 results**. ”3S+F” in the third column refers to inference with 3 scales and horizontal flips. Models submitted to the test server are trained on both training and validation data.

model	head	dataset	AVA val mAP	AVA test mAP
SlowFast, R101, 8 × 8	Linear	AVA	30.30	-
SlowFast, R101, 8 × 8	Linear	AVA-Kinetics	**32.25**	-
SlowFast, R101, 8 × 8	ACAR	AVA	32.29	-
SlowFast, R101, 8 × 8	ACAR	AVA-Kinetics	**34.15**	-
SlowFast, ensemble [5]	Linear	AVA	-	34.25
SlowFast, ensemble	mixed	AVA-Kinetics	-	**38.30**

Table 2: **Effect of adding Kinetics data on AVA v2.2**. The four single models are tested with single scale (256).

size of 32 clips. We freeze batch normalization layers in the backbone network. We train most models for 55k steps (6 epochs on the training set) with a base learning rate of 0.064, which is decreased by a factor of 10 at iterations 51k and 53k. A few models are trained with an extended 8-epoch schedule for final ensemble. We perform linear warm-up [6] during the first 9k iterations. For models submitted to the test server, we train on both training and validation data for the same number of epochs. We use weight decay of 10^{-7} and Nesterov momentum of 0.9. For a model with $T \times 8$ input sampling, we use $4T$ frames centered at the key frame as input, sampled with a temporal stride of 2. Note that in some Kinetics data, the annotated timestamps are too close to the end of the videos, and in these cases we simply sample the last $4T$ frames. In order to better preserve spatial structure, we do not use spatial random cropping augmentation. Instead, we only scale the shorter side of the input frames to 256 pixels, and zero pad the longer side to the same size in order to simplify mini-batch training. For AVA, we use both ground-truth boxes and predicted human boxes from [18] for training, and only ground-truth boxes for generating feature banks. For Kinetics, we only use ground-truth boxes for training, and our detection boxes for generating feature banks. The bank of features are extracted from short clips sampled with a temporal stride of 1 second from both AVA and Kinetics videos. We use bounding box jittering augmentation, which randomly perturbs box coordinates by a scale at most 7.5% relative to the original size of the bounding box during training.

Inference. At test time, we use AVA detections with confidence $\geq$ 0.7 and Kinetics detections with confidence $\geq$ 0.65. We scale the shorter side of input frames to 256 pixels, and apply the backbone feature extractor fully-convolutionally. We also report results tested with three spatial scales $\{256, 288, 320\}$ and horizontal flips.

2.2. Main Results

We present our results on AVA-Kinetics v1.0 in Table 1. The default backbone instantiation is SlowFast R101 8×8. The simplest baseline, linear classifier head, already has nearly 33mAP. Switching to our ACAR-Net still brings about a significant 1.6 increase in mAP. This highlights the importance of modeling high-order relations. Further adding long-term support (ACFB head) gives a total boost of 2.86mAP. We also experiment on a more advanced backbone (SlowFast R152 8×8) which brings some extra improvement in performance (+0.54mAP). We re-trained this SlowFast R152 model on training and validation data, and submitted it to the test server. Its performance almost did not drop (-0.13mAP).

For final ensemble, we combined predictions of 20 models with 4 different backbones, several different heads (Linear, LFB [18], ACAR, ACFB), and different schedules (6 *vs.* 8 epochs). Similar to [1], for each action class, we set weights in the ensemble according to APs of the models on this class.

2.3. Ablation Experiments

Effect of Adding Kinetics Data. For two SlowFast R101 8×8 models, we train the same model with two different datasets, AVA-Kinetics and AVA only, and evaluate on AVA v2.2 validation set. As shown in Table 2, adding Kinetics data brings consistent mAP increases (roughly +2mAP) to these two models. Moreover, with the help of both high-order relation reasoning and Kinetics data, our ensemble achieves a significant enhancement of **+4.05mAP** on AVA v2.2 test set compared to last year's winner [5].

Different Detectors. We investigate the effect of person detection AP@50 on action detection mAP. We perform the comparison on SlowFast R101 8×8 with ACAR head. As presented in Table 3 and 4, person detection AP@50 on Kinetics is much lower than that on AVA. In addition, even though our detector have reached 95.8 AP@50 on AVA person detection, there is still a large gap (8.1mAP) in final mAP between our detection and ground-truth (GT). These results suggest that how to improve person detection for action localization still remains to be explored.

detector	AP@50	mAP
LFB [18]	93.9	33.73
Ours	**95.8**	34.15
GT	-	**42.25**

Table 3: **Different detectors on AVA v2.2**. Final action detection results are evaluated on SlowFast R101 8×8 with ACAR head.

detector	AP@50	mAP
Ours (AVA)	77.2	28.41
Ours	**84.4**	30.88
GT	-	**43.60**

Table 4: **Different detectors on Kinetics v1.0**. We test our AVA detector on Kinetics data with confidence threshold set to 0.9. Final action detection results are evaluated on SlowFast R101 8×8 with ACAR head.

2.4. Pre-training on Kinetics

We used 4 SlowFast models pre-trained from scratch on Kinetics-700 classification task. We show their single center crop (224×224) accuracy on Kinetics-700 validation set in Table 5. Note that our models might have not reached full convergence due to time limitation.

model	val acc.
SlowFast, R101, 8×8	60.2
SlowFast, R101, 16×8	62.6
SlowFast, R152, 8×8	63.4
SlowFast, R152, 16×8	66.1

Table 5: **Pre-training on Kinetics-700**. All results are obtained with spatial and temporal center crop (single crop).

References

[1] Takuya Akiba, Tommi Kerola, Yusuke Niitani, Toru Ogawa, Shotaro Sano, and Shuji Suzuki. Pfdet: 2nd place solution to open images challenge 2018 object detection track. *arXiv preprint arXiv:1809.00778*, 2018. 3

[2] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 2

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition in activitynet challenge 2019. http://static.googleusercontent.com/media/research.google.com/en/ava/2019/fair_slowfast.pdf, 2019. 3, 4

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 3

[7] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018. 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[10] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 2

[11] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020. 2

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[13] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place

solutions for openimage2019–object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2020. 2

[14] Junting Pan, Siyu Chen, Zheng Shou, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. `https://junting.github.io/preprint/acar.pdf`. 1

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[16] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2020. 2

[17] 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. 2

[18] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 2, 3, 4

[19] Jin Xia, Jiajun Tang, and Cewu Lu. Three branches: Detecting actions with richer features. *arXiv preprint arXiv:1908.04519*, 2019. 2