Symbiotic Attention: UTS-Baidu Submission to the EPIC-Kitchens 2020 Action Recognition Challenge

Xiaohan Wang[1,2*], Yu Wu[1,2*], Linchao Zhu[1], Yi Yang[1], Yueting Zhuang[3]
{xiaohan.wang-3,yu.wu-3}@student.uts.edu.au;
{linchao.zhu,yi.yang}@uts.edu.au; yzhuang@zju.edu.cn
[1]The ReLER lab, University of Technology Sydney, [2]Baidu Research, [3]Zhejiang University

Abstract

In this report, we describe the technical details of our solution to the EPIC-Kitchens Action Recognition Challenge 2020. The EPIC-Kitchens dataset contains various small objects, intense motion blur, and occlusions. We tackle the egocentric action recognition task by suppressing background distractors and enhancing action-relevant interaction. First, we take candidate objects information to enable concentration on the occurring interactions. Second, we leverage a symbiotic attention mechanism with object-centric alignment to encourage the mutual interaction between the two branches and select the most action-relevant candidates for classification. Third, we incorporate multiple modality inputs, i.e.*, RGB frames and optical flows, to further improve the performance by a multi-modal fusion. Our model ranked the first on both the seen and unseen test set on EPIC-Kitchens Action Recognition Challenge 2020. The code for our model will be available at* `https://github.com/wxh1996/SAP-EPIC`*.*

1. Introduction

Egocentric action recognition provides a uniquely naturalistic insight into how a person or an agent interacts with the world, which requires distinguishing the object that human is interacting with from various small distracting objects. In EPIC-Kitchens [4], due to the large action vocabulary, researchers [4, 14, 5] usually decouple actions into verbs and nouns, and then further train separate CNN models for the verb classification and noun classification, respectively. The verb branch focuses on classifying actions (verbs) that the actor is performing, e.g., put and open, while the noun branch is to identify the object which the actor is interacting with. The predictions from the two branches are usually directly merged without further interactions for action classification in previous works [4, 14, 5]. However, these works ignore the mutual relation between the standalone branches.

Recently, Wang *et al.* [13] introduced a novel Symbiotic Attention with Object-centric Alignment (SAOA) framework for egocentric video recognition. SAOA better exploits the benefits of the interactions among different sources, enabling mutual communication between the verb and noun branches via object detection features. Our solution to EPIC-Kitchens Action Recognition Challenge 2020 is based on the SAOA framework [13]. Our ensemble SAOA model was ranked fist on both the seen and the unseen test set.

The SAOA framework [13] introduces an object-centric feature alignment method to dynamically integrate location-aware information to the verb and the noun branches. SAOA extends symbiotic attention with privileged information (SAP) [12] by introducing the local-alignment method for the verb branch and evaluating more backbones and input modalities. The object-centric feature alignment encourages the meticulous reasoning between the actor and the environment. The object features and locations are extracted by an object detection model, providing finer local information that is beneficial to the attendance of an on-going action. The noun branch and the verb branch integrate location-aware information by two different approaches, *i.e.*, the global alignment and the local alignment. With the object-centric alignment, we obtain a set of candidate verb features and noun features. The symbiotic attention mechanism [13] is then introduced to enable mutual interactions between the two branches and select the most action-relevant features. It consists of two modules, *i.e.*, cross-stream gating mechanism and action-attended relation module. The SAOA method dynamically integrates three sources of information towards better action recognition.

Our final submission was obtained by an ensemble of the SAOA model [13], the SAP model [12] trained on both RGB and flow modalities. Our results demonstrate that the

[*] Equal contribution. This work was done when Xiaohan and Yu were interned at Baidu Research.

SAOA model [13] achieves the state-of-the-art performance in action recognition on the EPIC-Kitchens dataset.

2. Our Approach

As illustrated in Fig. 1, the SAOA model [13] includes three stages. First, the location-aware information is integrated into the feature from one branch by the object-centric alignment method. Second, the fused object-centric features are recalibrated by the other branch utilizing a cross-stream gating mechanism. After that, the normalized feature matrix is attended by the other branch to aggregate the most action-relevant information within an action-attended relation module. More details can be found in our paper [13].

2.1. Object-centric Feature Alignment

We decouple the action labels into verbs and nouns, and train two individual 3D CNNs as the backbones in our framework, with one for the verb feature extraction and the other for the noun feature extraction. The object-centric features are extracted by an object detection model, providing finer local information that is beneficial to the attendance of an on-going action. Specifically, we use a pre-trained detection model to provide detailed information of objects in the video. For each video, we use M sampled frames for detection inference. We keep top-K object features and corresponding proposals according to their confidence scores for each sampled frame. The output of the $RoIAlign$ layer of the detection model is regarded as the feature and location for each detected object. The noun branch and the verb branch integrate location-aware information by two different approaches.

Global alignment for noun classification. The noun features and the detection features are complementary to each other, and proper integration of these two features produces more accurate identification of the interacted object. In the global alignment, we concatenate each detection feature with the global noun feature followed by a nonlinear activation. The generated feature matrix incorporates both local relevant features and global contextual features, which restrain the features of irrelevant objects.

Local alignment for verb classification. The verb feature contains motion information, which is quite different from the appearance information in noun feature and object features. Thus, we integrate spatially-aligned verb features with object features. In this way, the most relevant verb features will be generated for better alignment with local object features. It eases the difficulties of the integration between verb features and local object features. For each object detection feature, we have a corresponding spatial detection location. We extract regional verb features from the verb branch by pooling from the spatial feature map with the given candidate spatial location. The regional motion feature is then combined with the corresponding detection feature. The final motion-object paired feature incorporates local detection features and location-aware motion features.

The fused object-centric feature matrix contains useful local details. However, due to the existence of inaccurate detection regions, there are a few disturbing background noises in the features. To address this problem, [13] utilized a cross-stream gating mechanism to enhance the interaction between the verb stream and the noun stream. Furthermore, [13] proposed an action-attended relation module to underline the action relevant information.

2.2. Cross-Stream Gating

Taking the noun classification as an example, for an input noun feature matrix, we use the global verb feature to generate gating weights for it. The output features are produced by re-scaling the noun feature matrix with the gating weights. After re-calibrating the object-centric noun feature by the verb feature, the action-unrelated noise can be suppressed. Moreover, the cross-stream gating mechanism enables mutual communication between the two branches, which adaptively exploits the correlations of verbs and nouns. The detailed formulation of cross-stream gating can be found in [13].

2.3. Action-attended Relation Module

The calibrated object-centric feature matrix contains the action-relevant information and implicit guidance about the spatio-temporal position of an on-going action. To make full use of the information, we consider uncovering the relationships among the features [13]. First, we assess the relevance between the global feature and location-aware object-centric features. Second, we sum the object-centric features weighted by the relevance coefficients. Specifically, we perform attention mechanism on the normalized object-centric noun features and the global verb feature. Through the interaction of global feature and object-centric features, our model selects the most action-relevant feature for classification.

2.4. Action Re-weighting

The actions are determined by the pairs of verb and noun. The primary method of obtaining the action score is to calculate the multiplication of verb and noun probability. However, there are thousands of combinations and most verb-noun pairs that do not exist in reality, *e.g.*, “open knife”. In fact, there are only 149 action classes with more than 50 samples in the EPIC-Kitchens dataset [4]. Following the approach in [14], we re-weight the final action probability by the occurrence frequency of action in training set.

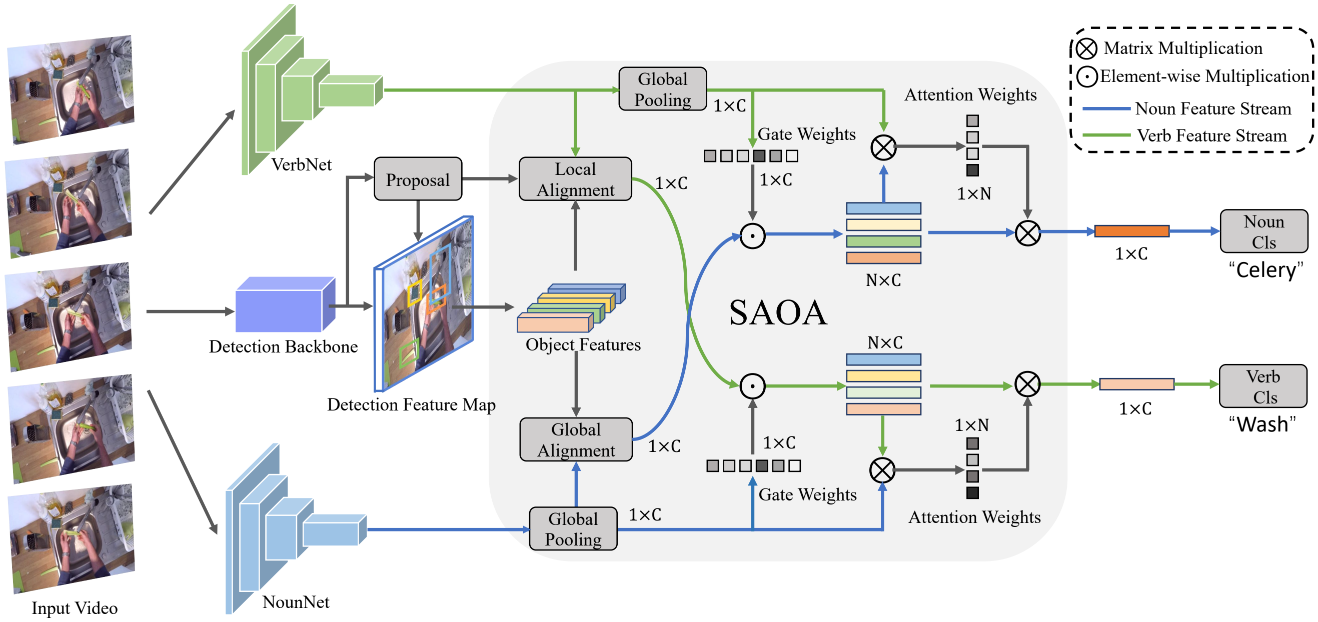

Figure 1. The SAOA framework. The framework consists of three feature extractors and one interaction module. The detection model generates a set of local object features and location proposals. This location-aware information is injected to the two branches by an object-centric alignment method. More details can be found in [13].

Table 1. The results on the EPIC-Kitchens validation set. "Obj" indicates the method leverages the information from the object detection model.

Method	Backbone	Input Type	Pre-training	Actions		Verbs		Nouns	
				top-1	top-5	top-1	top-5	top-1	top-5
SAP [12]	R-50	RGB+Obj	Kinetics	25.0	44.7	55.9	81.9	35.0	60.4
SAP [12]	R-50	Flow+Obj	Kinetics	24.5	43.1	56.1	81.4	33.6	58.7
SAP [12]	R-50	2-Stream	Kinetics	26.6	47.6	59.5	83.1	36.5	62.2
SAP [12]	R(2+1)D-34	RGB+Obj	IG-Kinetics	27.9	47.9	59.1	82.8	38.9	62.2
SAOA [13]	R-50	RGB+Obj	Kinetics	25.7	45.9	57.7	82.3	34.8	59.7
SAOA [13]	I3D	RGB+Obj	Kinetics+ImageNet	24.3	44.3	55.1	80.1	34.7	61.4
SAOA [13]	I3D	Flow+Obj	Kinetics+ImageNet	25.2	43.1	56.9	79.7	35.0	59.7
SAOA [13]	I3D	2-Stream	Kinetics+ImageNet	28.8	48.4	60.4	82.8	37.4	63.8
Ensemble	-	-	-	30.3	50.6	63.4	84.7	40.3	65.6

3. Experiments

3.1. Implementation Details

We followed [12, 13] to train our model. We train the framework in a two-stage optimization scheme. Specifically, we firstly pre-train the base models (VerbNet, NounNet, and the detector) individually. After that, we optimize the subsequent SAOA using extracted features from the base models. Damen *et al.* [3] train the recognition models on EPIC-Kitchens with a dropout layer. This strategy is not used in our models.

Backbone details. We adopt three typical 3D CNNs as our backbones, *i.e.*, ResNet50-3D [6], I3D [2], and R(2+1)D-34 [10].

For ResNet50-3D and I3D, we take the Kinetics [2] pre-trained weights to initialize the backbone. We then train the backbone models (VerbNet and NounNet) individually on the target dataset using 64-frame input clips. The targets for the VerbNet and NounNet are the verb label and noun label, respectively. The videos are decoded at 60 FPS. We adopt the stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0001 to optimize the parameters for 35 epochs. The overall learning rate is initialized to 0.003, and then it is changed to 0.0003 in the last 5 epochs. The batch size is 32. During the first training stage, the input frame size is 224 × 224, and the input frame is randomly cropped from a random scaled video whose side is randomly sampled in [224, 288]. We sample 64 successive frames with stride=2 from each segment to constitute the input clip. The center index of the input clip is randomly chosen in the seg-

Table 2. Results on the leaderboard of EPIC-Kitchens Action Recognition Challenge.

	Method	Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Seen	Baidu-UTS 2019 [11]	69.80	52.26	41.37	90.95	76.71	63.59	63.55	46.86	25.13	46.94	49.17	26.39
	TBN Single Model [7]	64.75	46.03	34.80	90.70	71.34	56.65	55.67	43.65	22.07	45.55	42.30	21.31
	TBN Ensemble [7]	66.10	47.89	36.66	91.28	72.80	58.62	60.74	44.90	24.02	46.82	43.89	22.92
	SAP R-50(RGB)	63.22	48.34	34.76	86.10	71.48	55.91	36.98	41.94	14.60	31.56	45.24	15.94
	SAOA I3D(2-Stream)	67.58	47.79	37.68	89.21	71.83	59.25	57.79	42.13	19.62	42.65	44.75	20.72
	Ensemble w/o IG	70.13	52.49	41.78	90.97	76.71	63.92	60.20	47.38	25.00	45.40	49.57	25.84
	Ensemble	70.41	52.85	42.57	90.78	76.62	63.55	60.44	47.11	24.94	45.82	50.02	26.93
Unseen	Baidu-UTS 2019 [11]	59.68	34.14	25.06	82.69	62.38	45.95	37.20	29.14	15.44	29.81	30.48	18.67
	TBN Single Model [7]	52.69	27.86	19.06	79.93	53.78	36.54	31.44	21.48	12.00	28.21	23.53	12.69
	TBN Ensemble [7]	54.46	30.39	20.97	81.23	55.69	39.40	32.57	21.68	10.96	27.60	25.58	13.31
	SAP R-50(RGB)	53.23	33.01	23.86	78.15	58.01	40.53	24.29	28.22	11.02	22.76	28.11	13.72
	SAOA I3D(2-Stream)	58.14	34.38	25.81	82.59	60.40	45.13	38.86	28.69	14.83	28.70	30.06	17.52
	Ensemble w/o IG	60.60	36.09	26.60	83.07	62.89	47.39	40.06	32.09	16.49	29.80	31.80	18.92
	Ensemble	60.43	37.28	27.96	83.06	63.67	46.81	35.23	32.60	17.35	28.97	32.78	19.82

ment during training. For the testing, we sample a center clip per segment. We resize the clip to the size of 256×256 and use a single center crop of 224×224.

For the R(2+1)D-34 backbone training, we use the weights pre-trained on IG-Kinetics-65M [5] as the initialization. The input frames are kept as 32 with stride=4 due to the large GPU memory cost of R(2+1)D-34. We train the model for 20 epochs. The learning rate is initialized to 0.0002 and then decayed by a factor of 0.1 after 9 and 18 epochs. The input size is 112×112 pixels randomly cropped from frames whose side is randomly sampled in [112, 144]. During the second-stage training and the final testing, the input size is 128×128 without cropping.

Detector details. Following [14, 13], we use the same Faster R-CNN to detect objects and extract object features. The detector is first pre-trained on Visual Genome [8] and then fine-tuned on the training split of the EPIC-Kitchens dataset. We use SGD optimizer to train the model with momentum 0.9 and weight decay 0.0001. We use a batch size of 12 and train the model on EPIC-Kitchens for 180k iterations for the trainval/test split. We use an initial learning rate of 0.005, which is decreased by a factor of 10 at iteration 140k and 160k. For the train/val split, we train the model for 150k iterations, and the learning rate decays at iteration 116k and 133k. Finally, our object features are extracted using *RoIAlign* from the detector's feature maps. For each video clip, we perform object detection on a set of frames that are sampled around the clip center within a fixed time duration. The time duration is set to 6 seconds for global alignment and 4 seconds for local alignment. The sample rate is at two frames per second. For each frame, we keep the top five features and proposals according to the confidence scores. Therefore, given a video clip, we obtain 60 detection features during global alignment. In local alignment, we obtain 40 detection features and corresponding locations.

SAOA details. We leverage the pre-trained backbone models and the detection models as the feature extractors. During the second-stage training, only the weights of SAOA are updated. We use SGD with momentum 0.9 and weight decay 0.0001 to optimize the parameters with batch-size of 32. For the model equipped with the I3D backbone, we train the model for 15 epochs. The learning rate is initialized to 0.001 and then reduced to 0.0001 in the last 5 epochs. For the models based on R-50, we train the model for 15 epochs, and the learning rate is set to a constant value 0.0001. Notably, since the detection features have different scales from the I3D features, the features from the I3D backbone need to be normalized before concatenation with detection features in the alignment modules. However, the feature from the R-50 backbone can be directly fed to the SAOA module without normalization. The main reason is the different network types between the detection backbones (based on residual block) and the I3D model (based on Inception block). Specifically, the features produced by the I3D backbone and detection model are l_2-normalized before concatenation. The combined feature is then multiplied by the l_2-norm of the I3D feature to scale the amplitude. A similar normalization strategy is introduced in [9]. During the training and testing of SAOA, we utilize the same temporal sampling strategy during the training and testing of the backbone. For each input video clip, we resize it to the size of 256. Then we feed the 64-frame clip to the network without spatial cropping. More training details of SAOA can be found in [13].

3.2. Results

We train three backbones and two models (SAP and SAOA) on the EPIC-Kitchens dataset. Following [1], we split the original training set of EPIC-Kitchens into the new train and validation set. The results of different models on the validation set are shown in Table 1. "2-Stream" indicates the results obtained by fusing the predictions of the "RGB+Obj" model and the "Flow+Obj" model. Our 2-Stream SAOA based on the I3D backbone achieves the highest performance compared to other models without ensemble. This shows that our 2-Stream SAOA framework is capable of integrating benefits from both RGB and Flow input. Our SAOA R-50 (RGB) achieves higher verb top-1 accuracy than SAP R-50 (RGB) by 1.8%. This demonstrates the effectiveness of the local alignment method for the verb branch. "Ensemble" indicates the result obtained by fusing the predictions of the above model. The ensemble improves the top-1 action accuracy by 0.9% over the SAOA I3D (2-stream) model.

The results on the test seen set and unseen set are shown in Table 2. Our single model SAOA I3D (2-stream) outperforms the ensemble of TBN [7]. The best result is achieved by "Ensemble" that fuses the predictions of all models (trained on the entire training set) in Table 1. we also show the result of the ensemble ("Ensemble w/o IG"), which fuses the predictions of all other models except the SAP R(2+1)D-34 model. The SAP R(2+1)D-34 model is first pre-trained on the IG-Kinetics dataset and then fine-tuned on EPIC-Kitchens. We observe that pre-training on such a large-scale video dataset (in the format of verb-noun labeling) clearly boosts the noun classification on the unseen set. The ensemble ("Ensemble") is ranked first on both seen and unseen set in the EPIC-Kitchens Action Recognition Challenge 2020.

4. Conclusion

In this report, we described the model details of SAP and SAOA. We introduced the object features and locations to enable concentration on the occurring actions. Moreover, we utilize the symbiotic attention mechanism to discriminate interactions in the egocentric videos. We reported the results of the two models with different input modalities and backbones. Our method achieved state-of-the-art on the EPIC-Kitchens dataset.

References

[1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018.

[2] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[3] Dima Damen, Hazel Doughty, Giovanni Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE T-PAMI*, 2020.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[5] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[6] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[7] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.

[8] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2016.

[9] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018.

[10] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[11] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Baidu-uts submission to the epic-kitchens action recognition challenge 2019. In *CVPR-W*, 2019.

[12] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *AAAI*, 2020.

[13] Xiaohan Wang, Linchao Zhu, Yu Wu, and Yi Yang. Symbiotic attention for egocentric action recognition with object-centric alignment. *In Submission, available at* `https://drive.google.com/file/d/1v5Oq_QO5q0zlXdTXpsHY9t1hp7jYiXLC`.

[14] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019.