Detecting Egocentric Actions with ActionFormer

Chenlin Zhang[1,2] Lin Sui[2] Abrar Majeedi[3] Viswanatha Reddy Gajjala[3] Yin Li[3,4]
[1]4Paradigm Inc., Beijing, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
[3]Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison
[4]Department of Computer Sciences, University of Wisconsin-Madison

Abstract

This report describes our submission to EPIC Kitchens 100 action detection challenge 2022. Our submission builds on ActionFormer – our previous work on temporal action localization [15], and integrates latest video features from SlowFast [7] and ViViT [1]. Our solution achieves 21.36 mAP on the validation set and 20.95 mAP on the test set, outperforms previous best results from the 2021 challenge by 4.84 absolute percentage points in average mAP, and is ranked 2nd on the public leaderboard of the 2022 challenge. Our code is available at https://github.com/happyharrycn/actionformer_release.

1. Introduction

Temporal action detection seeks to simultaneously localize action instances in time and recognize their categories. Many prior works have studied action detection in third person videos [2,4,9,10,12,14,16], yet few has focused on egocentric videos. Key challenges arise for egocentric action detection, as manifested in the EPIC-Kitchens dataset [6]. For example, most previous works have considered using action proposals [9] or anchor windows [10] to represent actions in time. An egocentric video, however, often contains hundreds of action instances from many categories spanning from a few seconds to a few minutes, making it difficult to design proposals or anchors.

Our solution instead considers an anchor-free model from our previous work [15]. Our work of ActionFormer presents one of the first Transformer based single-stage anchor-free model, capable of localizing moments of actions in a single shot without using action proposals or predefined anchor windows [15]. ActionFormer adapts local self-attention to model temporal context in untrimmed videos, classifies every moment in an input video, and regresses their corresponding action boundaries.

We explore the integration of different video features in ActionFormer, including SlowFast [7] and ViViT [1] (used by the winning team in the 2021 challenge [11]). We train two separate models for detecting the motion in the action (defined by verbs) and the active objects (defined by nouns), and further combine their outputs for action detection. Our submission achieves 21.36 mAP on the validation set and 20.95 mAP on the test set, outperforms previously best results from 2021 challenge by 4.84 absolute percentage points in average mAP. Our results are ranked 2nd on the public leaderboard of 2022 challenge, with a gap of 0.32 average mAP to the top ranked solution.

2. Our Approach

Our solution firsts extract clip-level video features using pre-trained video backbones. Each clip is thus represented as a feature vector, and each video a sequence of feature vectors. This sequence is further used by ActionFormer for action detection. ActionFormer considers every moment within the sequence as an action candidate, classifies their action category, and regress their action boundaries. We train two separate models to detect motion (verbs) and active objects (nouns), and combine their outputs. In what follows we describe the details of our approach.

2.1. Encoding Video Features

To extract video features, we consider two different video backbones, including (a) a variant (SlowFast R101-NL using 3D ResNet 101 with non-local blocks) of the SlowFast network [7] widely used for video understanding; and (b) a more recent video Transformer model (ViViT [1]) that has proven to be effective on EPIC-Kitchens dataset [8]. Both backbones are pre-trained on third person videos using Kinetics-600 [5]. Following [8], we further fine-tune the backbones on EPIC-Kitchens Action Recognition task, allowing the models to better adapt to egocentric videos. The fine-tuned backbones are then used to extract clip-level video features for action detection.

Fine-tuning on EPIC-Kitchens Action Recognition. Our first step is to fine-tune SlowFast R101-NL and ViViT for

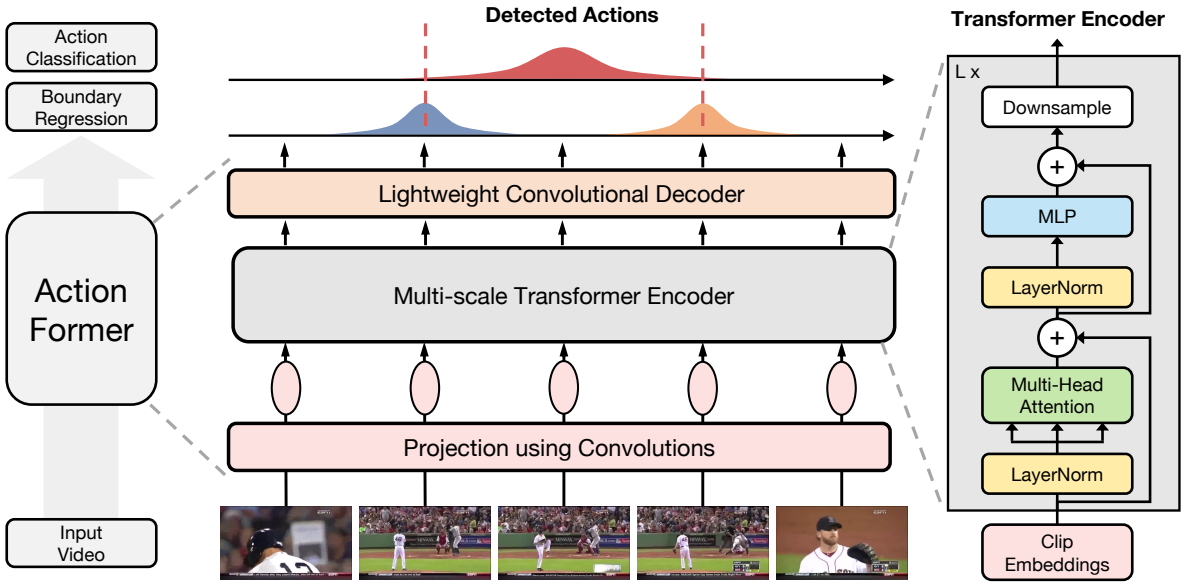

Figure 1. Overview of ActionFormer (taken from our paper [15]). Our method builds a Transformer based model to detect action instances in time by classifying every moment and estimating action boundaries, thereby providing a single-stage anchor-free model for temporal action localization.

action recognition on the training set of EPIC-Kitchens 100.

- **SlowFast R101-NL**: We attach a verb and a noun head to the pre-trained model, and fine-tune all weights on EPIC-Kitchens. Specifically, we randomly sample 32 frames with a temporal stride of 1 from downsampled videos (512×288 at 30 FPS). The model is fine-tuned by 30 epochs with batch size 64, weight decay 0.0001, and initial learning rate 0.01. The learning rates decays by 0.1 at 20th and 25th epoch. The fine-tuned model has 51.6% top-1 noun accuracy and 65.3% top-1 verb accuracy on the validation set with single-crop test.
- **ViViT**: We take the released model from [8], which are already fine-tuned on EPIC-Kitchens. Similar to SlowFast R101-NL, this version of ViViT include separate verb and noun heads for classification. The model reaches 58.9% top-1 noun accuracy and 67.4% top-1 verb accuracy on the validation set with multi-crop test. We refer to [8] for the training details.

Video Feature Extraction. Given the fine-tuned backbones, our next step is to extract clip-level video features for action detection. For both SlowFast and ViViT, we extract a feature vector for every clip of 32 RGB frames with a temporal stride of 8. Optical flow is not used for computing video features.

- **SlowFast R101-NL**: SlowFast network is fully convolutional. Thus, we input video frames with a higher resolution of 512×288, and perform an average pooling before the classification heads to extract a feature vector for each clip. The feature vector is of dimension 2304.
- **ViViT**: ViViT from [8] is trained on a resolution of 320×320 with 60 FPS, yet takes every other frames in the video (temporal stride 2). Altering the input resolution will require interpolating the learned position embeddings. Thus, we downsample the videos to 320×569 at 30 FPS, and feed 32 consecutive frames along with 3 horizontal crops each of size 320×320. The model processes these 3 crops independently, and feature vectors from the CLS token are further averaged to produce a 768-D clip-level feature.

We experimented with using individual features for action detection, yet found that a simple concatenation of the features yields the best performance.

2.2. Temporal Action Detection with ActionFormer

The extracted video features are further used by our ActionFormer for temporal action detection. ActionFormer first embeds each of the clip-level features. The embedded features are further encoded into a feature pyramid using a multi-scale transformer. The resulting feature pyramid is then examined by shared classification and regression heads, predicting action candidates at every time step. Our method is illustrated in Figure 1. We refer the readers to our paper for more technical details [15].

A Two Stream Model. While it is possible to attach separate verb and noun heads in a single ActionFormer model, we found it helpful to train individual models to detect motion (verbs) and active objects (nouns) and then combine their outputs, resembling the key idea of a two stream net-

Split	Method	Feature	Task	mAP@tIoU					
				0.1	0.2	0.3	0.4	0.5	mean
Val	BMN [6, 9]	SlowFast [7]	Verb	10.83	9.84	8.43	7.11	5.58	8.36
			Noun	10.31	8.33	6.17	4.47	3.35	6.53
			Action	6.95	6.10	5.22	4.36	3.43	5.21
	Huang [11]	ViViT [1]	Verb	22.92	21.86	20.89	18.33	15.66	19.93
			Noun	30.09	27.59	25.81	22.80	19.26	25.11
			Action	21.14	20.10	19.02	17.32	15.11	18.53
	Ours (ActionFormer [15])	ViViT [1]	Verb	23.23	22.35	21.28	19.69	16.50	20.61
			Noun	28.85	27.33	25.52	23.01	18.92	24.73
			Action	22.48	21.39	20.24	18.57	16.20	19.78
	Ours (ActionFormer [15])	SlowFast [7]+ViViT [1]	Verb	25.98	24.80	23.26	21.22	18.08	22.67
			Noun	30.49	29.14	26.88	24.77	20.70	26.40
			Action	23.87	22.91	21.70	20.28	18.04	21.36
Test	BMN [6, 9]	SlowFast [7]	Verb	11.10	9.40	7.44	5.69	4.09	7.54
			Noun	11.99	8.49	6.04	4.10	2.80	6.68
			Action	6.40	5.37	4.41	3.36	2.47	4.40
	Huang [11]	ViViT [1]	Verb	22.77	22.01	19.63	17.81	14.65	19.37
			Noun	26.44	24.55	22.30	19.82	16.25	21.87
			Action	18.76	17.73	16.26	14.91	12.87	16.11
	Ours (ActionFormer [15])	SlowFast [7]+ViViT [1]	Verb	26.97	25.90	24.21	21.77	18.47	23.46
			Noun	28.61	27.14	24.92	22.13	18.69	24.30
			Action	23.90	22.98	21.37	19.57	16.94	20.95

Table 1. Results of action detection on EPIC Kitchens 100. All results on the test set are evaluated on the test server. Our method achieves an average mAP of 20.95 for the 2022 challenge, surpassing previous best results from [11].

work [13]. A possible explanation is that doing so facilities implicit model ensemble. Specifically, each stream of ActionFormer predicts the classifications scores ($\boldsymbol{p}(verb)$ or $\boldsymbol{p}(noun)$) and regresses the temporal boundaries ($\boldsymbol{d}(verb)$ or $\boldsymbol{d}(noun)$) at each time step on the feature pyramid. We combine the outputs by using

$$\begin{aligned} \boldsymbol{p}(action) &= \boldsymbol{p}(verb)^{\alpha}\, \boldsymbol{p}(noun)^{(1-\alpha)}, \\ \boldsymbol{d}(action) &= \omega \boldsymbol{d}(verb) + (1-\omega)\boldsymbol{d}(noun), \end{aligned} \tag{1}$$

where $\alpha = 0.45$ (selected based on validation results) is used to "calibrate" the classification scores, and $\omega = \boldsymbol{p}(verb)/(\boldsymbol{p}(verb) + \boldsymbol{p}(noun))$ is used to re-weighted the regression outputs.

Implementation Details. Our model takes the concatenated features (3072-D for each clip with a temporal stride of 8) as the input, uses 6 levels of feature pyramid, and samples a sequence with maximum length of 4608 steps (approximately 20 minutes) for each video during training. The training epochs is 12 and 16 for verb and noun, respectively, as we observed overfitting issues with pro-longed training schedule. The results are further processed using multiclass SoftNMS [3]. We set the maximum predictions of each video to 15,000. Our code will be released in our public repository available at https://github.com/happyharrycn/actionformer_release.

3. Action Detection Results

We now present our results on EPIC Kitchens dataset.

Dataset. Our results are reported on EPIC Kitchens 100 action detection dataset [6]. EPIC Kitchens 100 is the largest egocentric action dataset with more than 100 hours of videos from 700 sessions capturing cooking activities across several kitchen environments. The dataset has an average 128 actions from a large array of categories per session. Each action is defined as a combination of a verb (action) and a noun (active object).

Evaluation Protocol and Metrics. We follow the official splits of train, validation and test set. When reporting results on validation set, we train our model on the training set. For the results on test set, we combine both training and validation sets for training and evaluate the results using the official server. Our results are reported for noun, verb and action, respectively. The metrics include the mean average precision (mAP) at different tIoU thresholds [0.1:0.1:0.5], as well as the average mAP , following [6].

Results. Table 1 summarizes our results on on the validation and test set. When using the same ViViT backbone and evaluated on the validation set, our method reaches an average mAP of 19.73% for action detection in comparison to the previous best result of 18.53% from Huang et al. [11] (also last year's winning solution). Adding SlowFast fea-

tures further improves the average mAP to 22.67%, 26.40%, and 21.36% for verb, noun, and action, respectively, largely outperforming the previous best [11] by 2.74%, 1.29%, and 2.83%. On the test set, our final model achieves 23.46%, 24.30%, and 20.95% mAP on verb, noun, and action, which is 4.09%, 2.43% and 4.84% higher than the previous best results [11]. Our average mAP for actions is slightly lower than the best ranked solution in the 2022 challenge, with a small gap of 0.32%.

4. Conclusion

In this report, we presented our solution using ActionFormer and latest video backbones for temporal action detection in egocentric videos. Notwithstanding its simplicity, our approach has demonstrated strong performance on the EPIC Kitchens dataset, ranked 2nd on the public leaderboard of 2022 challenge, surpassing previous best results and with a gap of 0.32 average mAP to the top ranked solution. We hope that our model can shed light on temporal action localization and egocentric vision, and the more broader problem of video understanding.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Int. Conf. Comput. Vis.*, 2021. 1, 3

[2] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Eur. Conf. Comput. Vis.*, volume 12373 of *LNCS*, pages 121–137, 2020. 1

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *Int. Conf. Comput. Vis.*, pages 5561–5569, 2017. 3

[4] Shyamal Buch, Victor Escorcia, Bernard Ghanem, and Juan Niebles Carlos. End-to-end, single-stream temporal action detection in untrimmed videos. In *Brit. Mach. Vis. Conf.*, pages 93.1–93.12, 2017. 1

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4724–4733, 2017. 1

[6] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1, 3

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Int. Conf. Comput. Vis.*, pages 6202–6211, 2019. 1, 3

[8] Ziyuan Huang, Zhiwu Qing, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Zhurong Xia, Mingqian Tang, Nong Sang, and Marcelo H Ang Jr. Towards training stronger video vision Transformers for EPIC-Kitchens-100 action recognition. *arXiv preprint arXiv:2106.05058*, 2021. 1, 2

[9] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *Int. Conf. Comput. Vis.*, pages 3889–3898, 2019. 1, 3

[10] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 344–353, 2019. 1

[11] Zhiwu Qing, Ziyuan Huang, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Changxin Gao, Marcelo H Ang Jr, and Nong Sang. A stronger baseline for ego-centric action detection. *arXiv preprint arXiv:2106.06942*, 2021. 1, 3, 4

[12] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5734–5743, 2017. 1

[13] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Adv. Neural Inform. Process. Syst.*, pages 568–576, 2014. 3

[14] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-TAD: Sub-graph localization for temporal action detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10156–10165, 2020. 1

[15] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 1, 2, 3

[16] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Int. Conf. Comput. Vis.*, pages 2914–2923, 2017. 1