Seperate Features Matter for Detecting Egocentric Actions

Chen-Lin Zhang[1] Lin Sui[2] Fangzhou Mu[4] 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 2023. Our submission builds on ActionFormer – our previous work on temporal action localization [22]. Our key finding is that using specialized video features for the noun and verb sub-tasks yield better results. Our submission (Team mzs) achieves a record 22.52 average mAP on the test set, outperforming the previous best results from the 2022 challenge by 1.24 absolute percentage points, and is the 1st place solution on the public leaderboard for the 2023 challenge. Our code is available at https://github.com/happyharrycn/actionformer_release.

1. Introduction

Temporal action detection aims to localize action instances and recognize their categories in untrimmed videos. Many prior works have studied action detection in third-person videos [2, 4, 13, 14, 17, 21, 23], yet few have focused on egocentric videos. Key challenges arise for egocentric action detection, as manifested in the EPIC-Kitchens dataset [6]. For example, egocentric actions are often defined by the combination of a verb (action) plus a noun (active object). However, most third-person actions only contain a single verb (action). Moreover, an egocentric video, however, often contains various action instances from many categories while some third-person datasets only contain a few action instances of the same category.

Our solution is built on an anchor-free model from our previous work [22]. 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 pre-defined anchor windows [22]. ActionFormer based methods have been popular among recent temporal action localization communities and competitions.

In EPIC-Kitchens 100 2022 action detection challenge, we explore the integration of different video features in ActionFormer, including SlowFast [8] and ViViT [1]. Also we train two separate ActionFormer models with the same features 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.

However, we found that shared features for verb and noun action detection are sub-optimal to egocentric action detection. Shared features will hurt the performance of both verb and noun detection, especially for verb detection. Thus, we use separate features for verb and noun detection. Then we perform post-processing with verb and noun sub-models. With a single video backbone VideoMAE [19] and InternVideo pretraining [20], our submission achieves 23.20 mAP on the validation set and 22.52 mAP on the test set, outperforming previously best results from 2022 challenge by 1.28 absolute percentage points in average mAP. Though we only use one video backbone, we outperform the 1st ranked solution in EPIC-Kitchens 100 2022 Action Detection challenge which uses a combination of multiple video features. Our results are ranked 1st on the public leaderboard of the EPIC-Kitchens 100 2023 challenge, with a gap of 4.35 average mAP to the 2nd ranked solution.

2. Our Approach

Our solution firsts extract clip-level video features using pre-trained video backbones. We do not use shared features for verb and noun detection. In contrast, we train two individual verb and noun classification models. The verb classification model only classifies the motion in the action while the noun classification model only recognizes the object. Then, we extract verb and noun features with the verb/noun classification models. Each clip is thus represented as a verb feature vector plus a noun feature vector, and each video is represented by two sequences of feature vectors. These sequences are further used by ActionFormer for verb and noun detection. We train an ActionFormer model for verb detection using verb features and we train a noun detection ActionFormer model with noun features. We combine the

output of individual models to form the final egocentric action predictions. In what follows we describe the details of our approach.

2.1. Encoding Video Features

To extract video features, we use a recent Transformer based model: VideoMAE [19]. VideoMAE extends the popular masked image modeling [11] framework into video action recognition. VideoMAE also expands the commonly used ViT [7] with joint space-time attention and achieves state-of-the-art performances over major action recognition datasets. Besides VideoMAE, a recent work InternVideo [20] utilizes multiple video datasets along with extra CLIP [16] models for unsupervised video pre-training. Thus, we choose VideoMAE-L models with InternVideo pre-training. VideoMAE-L is based on the ViT-L image recognition model with space-time joint attention. VideoMAE-L model is pre-trained on hybrid datasets: Kinetics [5], Something-Something V2 [9], AVA [10] and WebVid2M. We further fine-tune the backbones on EPIC-Kitchens 100 Action Recognition task, allowing the models to better adapt to egocentric videos. Please note that previous works often perform joint training with both verb and noun labels (two heads with a shared backbone). We perform separate training for verb and noun tasks, i.e., we train an individual VideoMAE-L model for verb prediction and we train an individual VideoMAE-L model for noun prediction. 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 VideoMAE-L models for verb/noun recognition on the training plus validation set of EPIC-Kitchens 100.

We first take the released VideoMAE-L model from [20]. Then we attach a single verb/noun classification head to the pre-trained VideoMAE-L model. Thus we have two individual models for verb and noun predictions. We randomly sample 32 frames with a temporal stride of 1 from downsampled videos (512×288 at 30 FPS). Following hyperparameters on Something-Something V2, The model is fine-tuned by 50 epochs with batch size 16. We use AdamW [15] with 0.05 weight decay, an initial learning rate 0.0003, and we use the cosine learning rate decay strategy. When only using the training subset, The fine-tuned model has 53.6% top-1 noun accuracy and 67.2% top-1 verb accuracy on the validation set with the single-crop test. We use training + validation subset to train our final feature extractor. We also experimented with shared features for both noun and verb detection (fine-tuning the backbone with two heads), yet it will result in a large action recognition and action detection performance drop.

Video Feature Extraction. Given the fine-tuned backbones, our next step is to extract clip-level video features for action detection. 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.

2.2. Egocentric 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 [22].

A Two Stream Model. Following suggestions in last year's challenge, we train individual models to detect motion (verbs) and active objects (nouns) and then combine their outputs, resembling the key idea of a two-stream network [18]. However, in contrast to last year's solutions, we use features from the verb backbone to detect motions and we use features from the noun backbone to detect active objects. We have a major performance gain over ActionFormer with shared features. The possible reason could be as follows: It has been proven that there exists a dilemma between multiple objects like classification and regression in object detection. There may also have dilemmas in egocentric action detection. Following previous works, each stream of ActionFormer predicts the classification 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 input features (1024-D for each clip with a temporal stride of 8) as the input, uses 8 levels of the feature pyramid, and samples a sequence with a maximum length of 4608 steps (approximately 20 minutes) for each video during training. The training epochs is 16 for both verb and noun, as we observed overfitting issues with pro-longed training schedule. The results are further processed using multi-class 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.

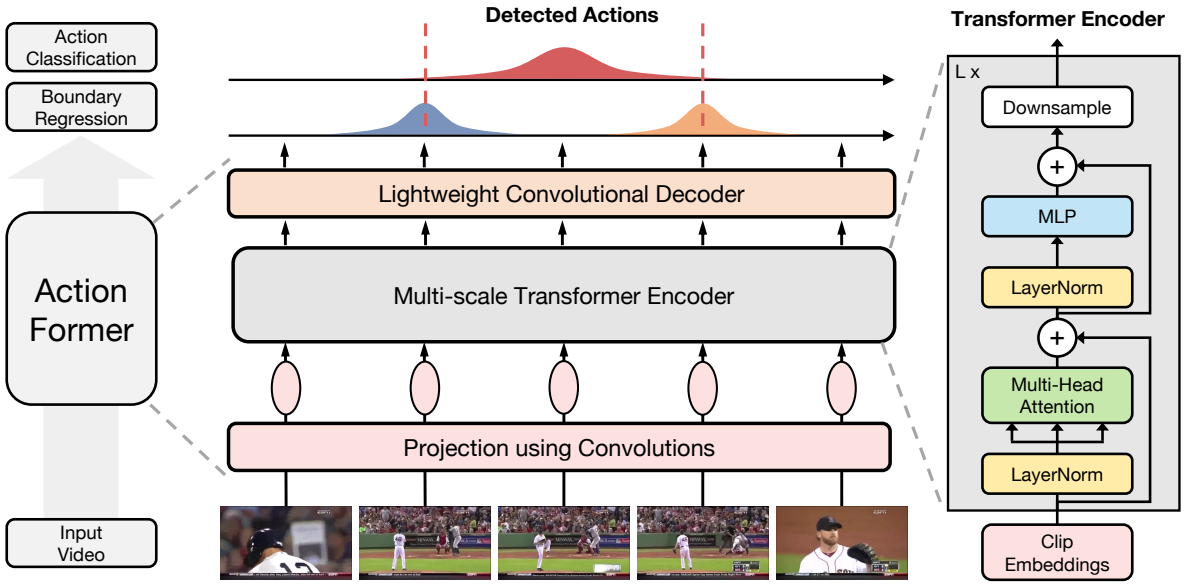

Figure 1. Overview of ActionFormer (taken from our paper [22]). Our method builds a Transformer based model to detect action instances in time by classifying every moment and estimating action boundaries, thereby providing a single-stage anchor-free model for temporal action localization.

Split	Method	Feature	Task	mAP@tIoU					
				0.1	0.2	0.3	0.4	0.5	mean
Val	Li [12]	TimeSFormer+SlowFast+MViT +Omnivore+MotionFormer	Verb	-	-	-	-	-	-
			Noun	-	-	-	-	-	-
			Action	27.19	26.23	24.38	22.47	19.82	24.02
	Ours 2022 Submission [22]	SlowFast [8]+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
	Ours 2023 Submission [22]	VideoMAE [19, 20]	Verb	**32.73**	**31.60**	**29.13**	**26.74**	**23.67**	**28.77**
			Noun	**31.32**	**29.70**	**27.25**	**25.32**	**21.33**	**26.98**
			Action	**25.73**	**24.98**	**23.72**	**22.46**	**19.11**	**23.20**
Test	Li [12]	TimeSFormer+SlowFast+MViT +Omnivore+MotionFormer	Verb	30.67	29.40	26.81	24.34	20.51	26.35
			Noun	**30.96**	**29.36**	26.78	23.27	18.80	25.83
			Action	24.57	23.50	21.94	19.65	16.74	21.28
	Ours 2022 Submission [22]	SlowFast [8]+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
	Ours 2023 Submission [22]	VideoMAE [19, 20]	Verb	**31.01**	**30.04**	**28.01**	**25.44**	**22.32**	**27.36**
			Noun	30.32	28.76	**27.20**	**24.28**	**20.74**	**26.26**
			Action	**25.54**	**24.54**	**23.16**	**21.04**	**18.35**	**22.52**

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 22.52 for the 2023 challenge, surpassing previous best results from [12].

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 the validation and test set. With a single VideoMAE [19] backbone and evaluated on the validation set, our method reaches an average mAP of 23.20% for action detection in comparison to the previous best result of 24.02% from Li et al. [12] (also last year's winning solution). We have a minor performance gap due to single features. However, on the test set, our final model achieves 27.36%, 26.26%, and 22.52% mAP on verb, noun, and action, which is 1.01%, 0.43% and 1.24% higher than the previous best results [12]. This phenomenon shows the robustness of our proposed training strategy.

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 1st on the public leaderboard of 2023 challenge, surpassing previous best results and with a gap of 4.35 average mAP to the 2nd 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. 2

[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. 2

[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, 4

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 2

[8] 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

[9] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Int. Conf. Comput. Vis.*, pages 5842–5850, 2017. 2

[10] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6047–6056, 2018. 2

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16000–16009, 2022. 2

[12] Lijun Li, Li'an Zhuo, and Bang Zhang. One-stage action detection transformer. *arXiv preprint arXiv:2206.10080*, 2022. 3, 4

[13] 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

[14] 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

[15] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in Adam. *arXiv preprint arXiv:1711.05101*, 2017. 2

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pages 8748–8763, 2021. 2

[17] 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

[18] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Adv. Neural Inform. Process. Syst.*, pages 568–576, 2014. 2

[19] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 1, 2, 3, 4

[20] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 1, 2, 3

[21] 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

[22] Chen-Lin Zhang, Jianxin Wu, and Yin Li. ActionFormer: Localizing moments of actions with transformers. In *Eur. Conf. Comput. Vis.*, pages 492–510, 2022. 1, 2, 3

[23] 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