DeeperAction Workshop at ECCV 2022: 3rd place solution for Kinetics-TPS Challenge on Part-level Action Parsing Challenge Track

Technical Report: End-to-End Part-level Action Parsing with Cascaded Transformer

Lianli Gao Ji Zhang Beitao Chen Pengpeng Zeng
Center for Future Media, University of Electronic Science and Technology of China, Chengdu, China
lianli.gao@uestc.edu.cn, jizhang.jim@gmail.com, chenbeitao@gmail.com
is.pengpengzeng@gmail.com

Abstract

Unlike previous works that follow a multi-stage process manner for part-level action parsing, we present a single-stage pipeline that unifies person detection, part detection and action state parsing into one model. Our motivation is that parsing human actions relies on the characteristics of persons, thus eliminating computational burdens caused by the multi-stage paradigm. Specifically, we borrow the idea from DETR [1] and regard part-level action parsing as a set prediction problem. In this way, we re-design the transformer decoder in DETR by adopting a cascaded structure, which is significantly beneficial to part-level action parsing. Our experiments show that simple variant of DETR achieves very competitive results, where it outperforms all previous entries [3, 2] in the last year, including two-stage or three-stage methods. Furthermore, the proposed method records a global mean of 66% *score in 2022 Kinetics-TPS Challenge.*

1. Introduction

The Part-level Action Parsing aims to recognize a human action by compositional learning of body part state in videos. In this work, we present a simple yet effective parsing pipeline, called action parsing with cascaded transformer (APCT). This method forms the basis for the submission to the Kinetics-TPS Challenge from **CFM-CMG** team.

The motivation behind the proposed method comes from these observations: First, the part-level action parsing requires to solve many sub-tasks, including person detection, instance-specific part detection, part-level action recognition and video-level action recognition. Secondly, the strategy of divide-and-conquer has been demonstrated to be effective to this task, since all previous entries to the KineticsTPS-Track follow this strategy and achieve promising results. Third, part-level identification (i.e., part detection and part state recognition) is the bottleneck for this task, as demonstrated in [3].

Table 1. Bottleneck analysis for part-level action parsing, which is summarized in [3]. ‘‘✓” means the predictions are replaced by corresponding ground truth. The results indicate that the part-level identification is the bottleneck for this task.

Frame-level			Video-level	Metric
Actor Detection	Part parsing		Action parsing	Acc^p
	part_det	state_parsing		
				33.32%
✓				35.33%
	✓			36.22%
	✓	✓		72.46%
			✓	42.20%
		✓		45.60%
✓	✓			38.10%
✓	✓	✓		77.30%
✓			✓	44.76%
	✓	✓	✓	93.41%
✓	✓	✓	✓	99.90%

Above observations motivate us to study two problems: 1) How to design a simple yet effective parsing pipeline for eliminating the computational burdens caused by multi-stage paradigm; and 2) how to equip this pipeline with the ability to improve the performance of part-level identification. In the next, we present all technical details.

2. Part-level Action Parsing

The overall pipeline of the proposed APCT is shown in Fig. 1. We follow the design rule of DETR [1] and build our model for supporting part-level action parsing. In particular, the proposed model aims to simultaneously perform person detection, part detection and part state identification. Based on this, the whole pipeline consists of two major modules: 1) cascaded transformer based architecture that models re-

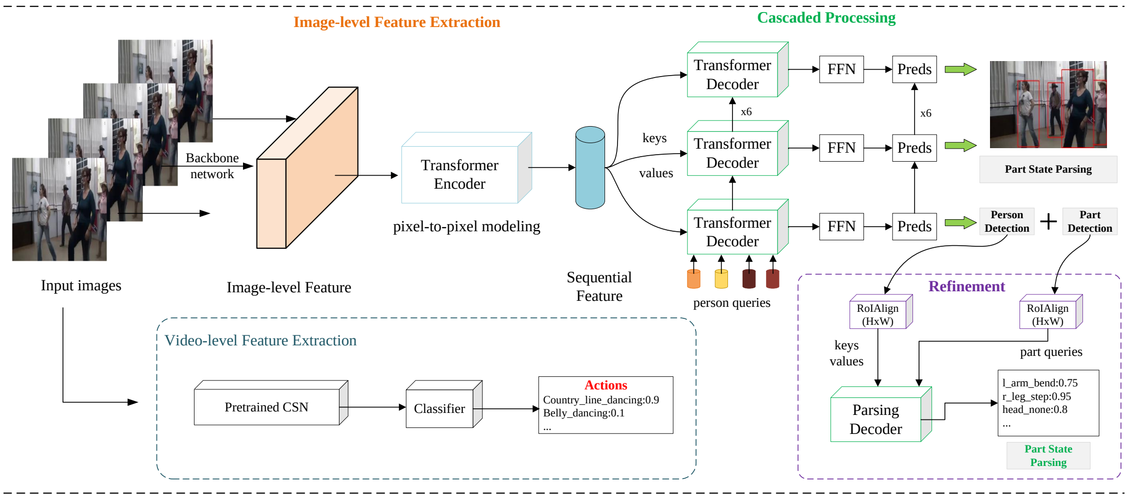

Figure 1. The overview of the proposed pipeline for part-level action parsing.

Table 2. Ablation study. The investigation of part parsing variants.

baseline (DETR)	cascaded processing	refinement	big backbone	2x learning schedule	Acc^p
✓					53.2%
✓	✓				59.5%
✓	✓	✓			60.5%
✓	✓	✓	✓		64.5%
✓	✓	✓	✓	✓	66.4%

lationships between persons and predicts a set of parsing results; 2) a multi-task loss that forces the model to accurately parse persons.

2.1. APCT Architecture

Following DETR, the overall APCT architecture contains four components: a backbone network for image-level feature extraction, a transformer based encoder for modeling relations between pixels, a cascaded transformer based decoder that performs multiple instance-level predictions and a refinement transformer that further refines predicted action states.

To implement above three components, we adopt a CNN or ViT based backbone network for image-level feature extraction, such as ResNet-50 or Swin Transformer. Formally, the extracted image feature is denoted as $f \in \mathbb{R}^{C\times H\times W}$, where $C = 2048$ is the number of feature channels and $\{H, W\}$ is the spatial size of the feature. Next, we reduce the channel dimension to a smaller one (i.e., 256) by using 1×1 convolution, resulting in a new image feature $f_z \in \mathbb{R}^{c\times H\times W}, c = 256$. Next, to model the pixel-to-pixel relations, we adopt a normal transformer encoder with 6 layers, each of which consists of one multi-head self-attention module and a feed forward network. Taking as input f_z, the transformer encoder models pixel-pixel relations and outputs a new feature $\hat{f}_z \in \mathbb{R}^{HW\times c}$.

The parsing decoder follows standard pipeline of the DETR decoder, which transforms N instance query to a set of prediction using multi-head attention mechanism. The difference with original DETR transformer decoder is that each transformer decoder layer in our model parses each instance conditioning on prior prediction from previous decoder layer. In this way, the parsing pipeline is performed in a cascaded manner. Specifically, our transformer decoder consists of 6 decoder layers, each of which consists of one multi-head cross-attention module and a feed forward network. Furthermore, pre-defined N instance embeddings are regarded as query, while the transformed image feature $\hat{f}_z$ is used as key and value. In this way, each decoder layer outputs N updated instance embeddings and N predictions. Note that the predictions involve many outputs, including instance classification score $b_c \in \mathbb{R}^{N\times 1}$, relative offsets $b_l \in \mathbb{R}^{N\times 4}$ to reference instance location, relative offsets $p_l \in \mathbb{R}^{N\times K\times 4}$ to reference part location and part-specific states $p_s \in \mathbb{R}^{N\times K\times S}$, where K is the number of part classes and S is the number of state classes. Both reference instance location and reference part location in the i-th (i=1,...5) decoder layer are decided by previous outputs from the (i-1)-th decoder layer, while they are initialized by zero values at first decoder layer.

Inspired by the work [4], we further design a refinement module, where it refines predicted action states by modeling part-state relations. Specifically, we separately extract region features from predicted part bbox and person bbox via RoiAlgin operation. Then, regarding the part features as the *query* and the person features as the *key* or *value*, we adopt a normal transformer to model part-person relations and predict new action states $\hat{p_s} \in \mathbb{R}^{N \times K \times S}$ for all parts.

2.2. Multi-task loss:

To enable the model to perform part-level action parsing, we design our learning objectives as follows:

$$\begin{aligned} \ell_{det} &= Cross\_Entropy(b_c, b_c^*) + SmoothL1(b_l, b_l^*) \\ \ell_{part} &= FocalLoss(p_s, p_s^*) + SmoothL1(p_l, p_l^*) \\ \ell_{refine} &= FocalLoss(\hat{p_s}, p_s^*) \\ \mathcal{L} &= \ell_{det} + \ell_{part} + \ell_{refine} \end{aligned} \tag{1}$$

where b_c^*,b_l^*,p_s^* and p_l^* are corresponding ground truths. Following DETR [1], we adopt bipartite matching algorithm to decide ground truth labels.

3. Video-level Action Recognition

Inspired by the previous work [2], we directly adopt CSN model as the backbone network for video-level feature extraction. In our work, the CSN model is pretrained on IG-65M dataset, and then is finetuned on Kinetics-TPS training set. Therefore, it achieves around 97% accuracy on test set, which is comparable to the result in [2]. Since we focus on the part-level parsing, we fix this trained CSN throughout experiments and do not apply any test-time augmentation such as model ensemble.

4. Experiments

4.1. Experimental Settings

Following official protocols, all models are trained in whole training set with 3809 videos and tested on the official server[1]. Besides, our models are implemented based on OpenMMLab[2] on an Ubuntu server with eight Tesla V100 graphic cards. In general, we use ResNet-50 as the backbone network and train our models for 12 epoch, unless other stated. All parsing models are initialized by COCO pretrained models.

4.2. Main Results

Our experimental results are summarized in Tab.2. From the results, we find that cascaded processing is particular important for single-state part-level action parsing, where it surprisingly improves the baseline model (i.e., DETR) by 6.3% (from 53.2% to 59.5%). Moreover, further refinement with modeling part-person relations achieves 1% gains, reaching a score of 60.5%. In addition, we apply a bigger backbone network and replace the ResNet-50 with Swin Transformer Base, achieving a higher score 64.5%. Finally, adjusting training schedule by extending the training epoch from 12 to 24, leads to a further improvement, attaining to 66.4%.

Table 3. 2021 Kinetics-TPS Challenge results on *test* set.

Ranks	Method	Acc^p
(1)	yuzheming	0.630532
(2)	Sheldong	0.613722
(3)	Josonchan	0.605059
(4)	fangwudi	0.590167
(5)	uestc.wxh	0.536067
(6)	hubincsu	0.490984
(6)	scc1997	0.490984
(7)	KGH	0.486483
(8)	zhao_THU	0.434311
(9)	TerminusBazinga	0.396735
(10)	cjx_AILab	0.370753

Tab. 3 lists final results in 2021 KineticsTPS competition. Compared with them, maybe unfair in some extends (little difference in testing set), our method achieves a higher score than that of them. It is worth noting that all top-3 teams adopt the multi-stage strategy for part-level action parsing. Therefore, the experimental results, at least, indicate exploring single-stage paradigm is a promising direction for part-level action parsing.

5. Conclusion

In this report, we present a simple yet effective single-stage approach for part-level action parsing. We unify person detection, part detection and action state parsing into one model. Following this, we design three major modules, including a backbone network for image-level feature extraction, a transformer based encoder for modeling relations between pixels, and a cascaded transformer based decoder that performs multiple instance-level predictions. Experiments on Kinetics-TPS dataset demonstrate the effectiveness of the proposed method, and it indicates the single-stage parsing pipeline is a promising direction for part-level action parsing.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[2] Xiaodong Chen, Xinchen Liu, Kun Liu, Wu Liu, and Tao Mei. A baseline framework for part-level action parsing and action recognition. *arXiv preprint arXiv:2110.03368*, 2021.

[3] Xuanhan Wang, Xiaojia Chen, Lianli Gao, Lechao Chen, and Jingkuan Song. Technical report: Disentangled action parsing networks for accurate part-level action parsing. *arXiv preprint arXiv:2111.03225*, 2021.

[4] Xuanhan Wang, Jingkuan Song, Xiaojia Chen, Lechao Cheng, Lianli Gao, and Heng Tao Shen. Ke-rcnn: Unifying knowledge based reasoning into part-level attribute parsing. *arXiv preprint arXiv:2206.10146*, 2022.

[1] https://codalab.lisn.upsaclay.fr/competitions/4392

[2] https://github.com/open-mmlab