Temporal Feature Augmented Network for Video Instance Segmentation

Minghui Dong[1,2], Jian Wang[1], Yuanyuan Huang[1], Dongdong Yu[1], Kai Su[1], Kaihui Zhou[1], Jie Shao[1], Shiping Wen[3] and Changhu Wang[1]

[1]ByteDance AI Lab
[2]Huazhong University of Science and Technology
[3]University of Electronic Science and Technology of China

Abstract

In this paper, we propose a temporal feature augmented network for video instance segmentation. Video instance segmentation task can be split into two subtasks: instance segmentation and tracking. Similar to the previous work, a track head is added to an instance segmentation network to track object instances across frames. Then the network can performing detection, segmentation and tracking tasks simultaneously. We choose the Cascade-RCNN as the basic instance segmentation network. Besides, in order to make better use of the rich information contained in the video, a temporal feature augmented module is introduced to the network. When performing instance segmentation task on a single frame, information from other frames in the same video will be included and the performance of instance segmentation task can be effectively improved. Moreover, experiments show that the temporal feature augmented module can effectively alleviate the problem of motion blur and pose variation.

1. Introduction

Detection, segmentation and tracking are three fundamental computer vision tasks, which attracted more and more attention in recent years. In the domain of video analysis, these tasks have wide range of applications such as autonomous driving and video editing. In [1], authors proposed a new task named as video instance segmentation. The task of video instance segmentation aims at performing detection, segmentation and tracking tasks simultaneously in videos. Video instance segmentation integrates three tasks into one framework, and the three tasks can share all the video level information.

The task of video instance segmentation can be split into two subtasks: instance segmentation and tracking. In [1], a tracking head is added to Mask-RCNN [2] to perform the task of instance segmentation and tracking simultaneously. Performance of video instance segmentation is limited by the accuracy of the two subtasks. Instance segmentation performs per-pixel labeling of objects at instance level, which usually implemented in two stages [2]. Besides that, Cascade R-CNN [3] proposed a multi-stage detection architecture to handle the problem of training hypotheses with low quality. The performance of detection task can be improved by a large margin with the cascade architecture. Moreover, Hybrid Task Cascade [4] introduced mask information flow branch and semantic segmentation branch to further improves the performance of instance segmentation, especially the mask head. For the task of video object tracking, there are two scenarios. One is detection-based tracking [5] and the other is detection-free tracking [6]. The detection-based tracking task is closer to the video instance segmentation.

Similar to the previous video instance segmentation work, we first choose the Cascade R-CNN as basic instance segmentation model. Then a tracking head is added to tracking object instances across frames. However, the traditional instance segmentation architecture is designed for still images and there are some gaps between the image instance segmentation and video instance segmentation. Firstly, directly applying the existing image instance segmentation model to video instance segmentation task is difficult due to the motion blur, pose variation and object occlusions. Moreover, video sequences contain more features than still images. These features are not used in existing instance segmentation model and can be used to further improve the performance of instance segmentation. To handle these problems, we introduced a temporal feature augmented module into the existing instance segmentation model for integrating temporal feature. In our model, before the ROI feature feeding into the second stage branch, similarity matching and feature fusion are first performed with ROI feature from

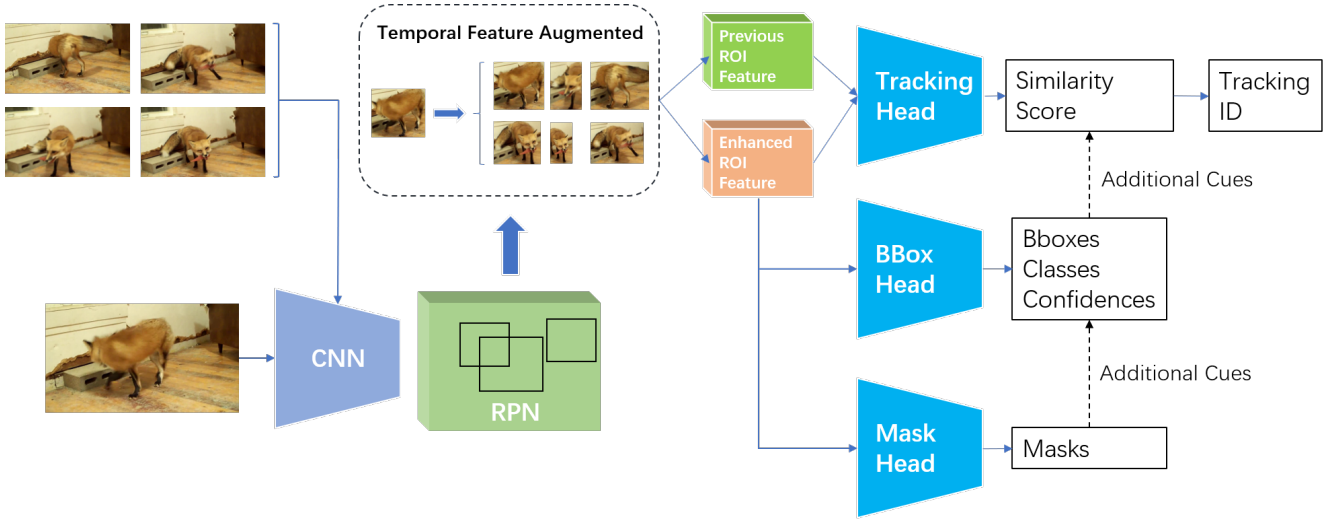

Figure 1. The overall architecture of temporal feature augmented network.

other frames. Then the final inference results (detection, segmentation and tracking) are based on the information from multi-frame, which is more accurate than the results obtained from single-frame. Besides, compared to the original tracking head in [1], we added a deformation factor to further improve the tracking performance.

2. Methods

2.1. Overall architecture

The overall architecture of our model is illustrated in Figures 1. Because the performance of video instance segmentation task is largely dependent on the accuracy of image instance segmentation, we choose the Cascade R-CNN as our basic still image instance segmentation model and ResNext-101 [7] as the backbone. Same to [1], in parallel to the original three branches (classification, bounding box regression, mask segmentation), a tracking head is added to assign object id to each candidate box. Before the ROI features feeding into the four branches, this ROI features are first enhanced by the temporal feature augmented module. Then the detection and segmentation results are obtained in the same way as in Cascade R-CNN, the tracking results are obtained in the same way as in [1].

2.2. Temporal feature augmentation module

The temporal feature augmented module is proposed to mine potential features in video sequence and alleviate the problem of motion blur, pose variation and object occlusions. The structure of temporal feature augmented module is illustrated in Figures 2. During training stage, we first randomly sample N frames $\{x_1, x_2 ... \mathrm{x}_n\}$ from the video sequence. Then one of the N frames is randomly selected as the training sample (suppose as x_k), and the other frames are used to provide temporal features. All sampled frames are first pass through the RPN to get proposals. Suppose $\{r_1^k, r_2^k ... r_m^k\}$ is the m proposals generated from x_k, and $\{r_1^s, r_2^s ... r_p^s\}$ is the p proposals generated from all sampled frames, where $p = N * m$, superscript k indicates proposals from the training sample and superscript s indicates proposals from all sampled frames. Then top t proposals $\{r_1^s, r_2^s ... r_t^s\}$ are selected from the p proposals for temporal feature augmenting according to the score.

Each proposal r_i^k from $\{r_1^k, r_2^k ... r_m^k\}$ is first computed similarity with each proposal r_j^s in $\{r_1^s, r_2^s ... r_t^s\}$. For each pair $\{\mathrm{r}_i^k, \mathrm{r}_j^s\}$, a siamese network f_e which is a fully connected layer embed them into a new feature space $\{\mathrm{e}_i^k, \mathrm{e}_j^s\}$. Then the similarity between the two proposals are measured by cosine similarity between the two embedding:

$$s_{ij} = \cos ine(f_e(r_i^k), f_e(r_j^s)) \tag{1}$$

The s_{ij} is normalized across the t proposals by softmax function. Then the proposal r_i^k is recomputed according to the similarity:

$$\hat{r_i^k} = \sum_{j=1}^{t} s_{ij} r_j^s \tag{2}$$

The recomputed proposal $\hat{r_i^k}$ feature is aggregated by proposal features from all sampled frames and therefore contains more temporal features. The recomputed proposal feature can be more discriminative when used for subsequent branches. For example, it is difficult to distinguish a fox facing away from the camera form a single frame. But if the proposal integrates features from other frames, in which the fox is facing the camera, it will be easier to distinguish its category.

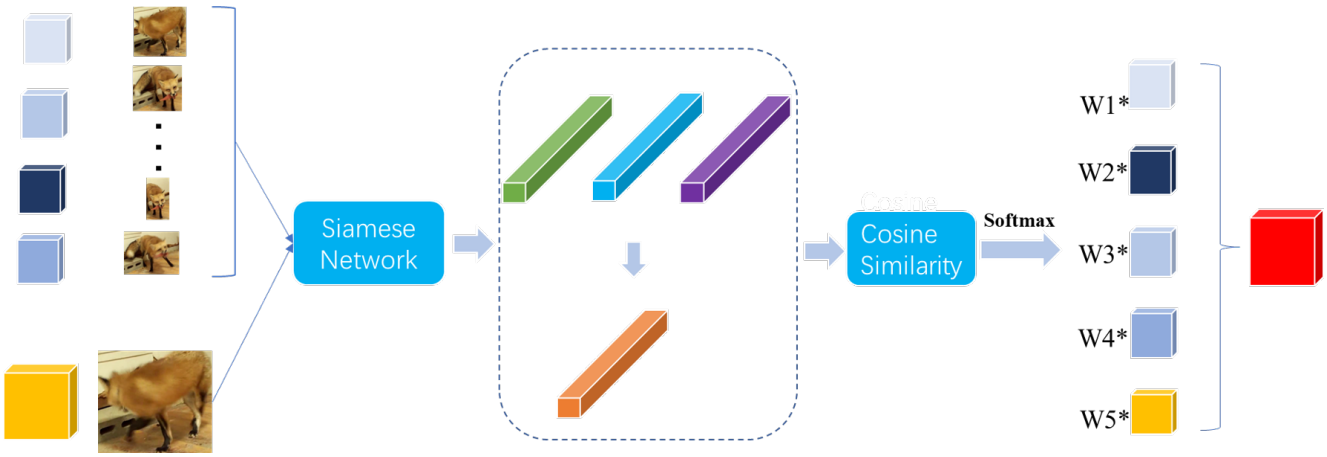

Figure 2. The temporal feature augmented module.

2.3. Tracking head

During training stage, tracking head in our model is same as [1]. During inference stage, we add more additional cues from mask head. Besides the assigning probability, detection score, IOU of bounding boxes and category label consistency, IOU of instance mask is also added as the additional cues. The IOU of instance mask can be viewed as a deformation factor and experiments show that it effectively improves tracking accuracy.

3. Experiments

Implementation details. We choose Cascade-RCNN as our basic image instance segmentation model due to its good performance in our experiments. ResNext101-FPN pretrained on MS-COCO dataset is adopted as the backbone. Overall architecture is implemented base on the public implementation [8]. We further split the YouTube-VIS training set into 1902 offline-training set and 336 offline-validation set. Our model is trained on offline-training set and evaluated on the offline-validation set. The results of YouTube-VIS validation set and test set are evaluated by submitting to the CodaLab site. The original frame sizes are downsampled to 640×360 for both training and evaluation in our model and the model is trained on four Tesla-V100 with batchsize 4.

Results on YouTube-Video Instance Segmentation Challenge. The proposed temporal feature augmented network achieves competitive performance on YouTube-Video Instancce Segmentation Challenge. Table 3 shows the ranking results on YouTube-Video Instance Segmentation Challenge test set. Our model gets 0.444 mAp which ranks fifth in the final leaderboard. Some qualitative results predicted by our model are shown in Figures 3. As shown in the qualitative results, the model can still get satisfactory results when dealing with frames with motion blur, pose variation and multi-objects.

4. Conclusion

In this paper, we proposed a temporal feature augmented network for video instance segmentation. A tracking head and a temporal feature augmented module is introduced to image instance segmentation model to perform the task of video instance segmentation. The temporal feature augmented module aggregated features from multi-frames for single frame inference, which can effectively alleviate the problem such as motion blur and pose variation. In the future, we will explore more architectures to utilize spatial-temporal feature for video instance segmentation.

References

[1] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," *CoRR*, vol. abs/1905.04804, 2019. [Online]. Available: https://arxiv.org/abs/1905.04804

[2] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[3] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[4] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[5] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with

#	Team	mAp		AP50		AP75		AR1		AR10	
1	Jono	0.467	(1)	0.697	(1)	0.509	(1)	0.462	(1)	0.537	(2)
2	foolwood	0.457	(2)	0.674	(3)	0.490	(3)	0.435	(5)	0.507	(4)
3	bellejuillet	0.450	(3)	0.636	(5)	0.502	(2)	0.447	(3)	0.503	(5)
4	linhj	0.449	(4)	0.665	(4)	0.486	(5)	0.453	(2)	0.538	(1)
5	**mingmingdiii(ours)**	0.444	(5)	0.684	(2)	0.487	(4)	0.436	(4)	0.508	(3)
6	xiAaonice	0.400	(6)	0.578	(9)	0.449	(6)	0.396	(9)	0.452	(9)
7	guwop	0.400	(7)	0.608	(7)	0.439	(8)	0.412	(7)	0.491	(6)
8	exing	0.397	(8)	0.621	(6)	0.426	(9)	0.414	(6)	0.461	(8)
9	player1	0.393	(9)	0.606	(8)	0.444	(7)	0.409	(8)	0.472	(7)
10	TeamXu	0.339	(10)	0.549	(13)	0.384	(10)	0.364	(10)	0.404	(11)

Table 1. Ranking results on YouTube-Video Instance Segmentation Challenge test set.

Figure 3. Qualitative results on YouTube-VIS dataset.

long-term dependencies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 300–311.

[6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[7] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[8] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.