Data-Efficient Instance Segmentation with a Single GPU

Pengyu Chen[2,*], Wanhua Li[1,*], Jiwen Lu[1]

[1]Department of Automation, Tsinghua University, China
[2]Beijing University of Posts and Telecommunications

chenpengyu2018@bupt.edu.cn   li-wh17@mails.tsinghua.edu.cn   lujiwen@tsinghua.edu.cn

Abstract

Not everyone is wealthy enough to have hundreds of GPUs or TPUs. Therefore, we've got to find a way out. In this paper, we introduce a data-efficient instance segmentation method we used in the 2021 VIPriors Instance Segmentation Challenge. Our solution is a modified version of Swin Transformer, based on the mmdetection which is a powerful toolbox. To solve the problem of lack of data, we utilize data augmentation including random flip and multiscale training to train our model. During inference, multiscale fusion is used to boost the performance. We only use a single GPU during the whole training and testing stages. In the end, our team named THU_IVG_2018 achieved the result of 0.366 for AP@0.50:0.95 on the test set, which is competitive with other top-ranking methods while only one GPU is used. Besides, our method achieved the AP@0.50:0.95 (medium) of 0.592, which ranks second among all contestants

1. Introduction

Instance segmentation is a popular research field in machine learning and computer vision due to the broad applications. However it is costly to build a large-scale database since a large number of tags on plenty of pictures are needed. Meanwhile, not everyone is wealthy enough to have hundreds of GPUs or TPUs. Therefore, we've got to find a way out. The challenge "2021 VIPriors Instance Segmentation Challenge", hosted at ICCV 2021, is raised which encourages researchers of arbitrary background to participate: no giant GPU clusters are needed, nor will training for a long time.

The dataset used in the challenge of instance segmentation is provided by Synergy Sports, the partner of this challenge, which contains 310 pictures shot during different basketball games, and the mission is to segment the basketball players and the basketball on the images. In the challenge we don't use any other datasets to augment

the dataset and any pre-trained backbones which, of course, utilizes outside database in another way. On the other hand, the state-of-the-art models are often trained with large-scale datasets, including DetectoRS [17], Cascade Eff-B7 NAS-FPN [8], QueryInst [7] and so on. These models may give a poor performance with insufficient labeled data. We believes that the recent proposed Swin Transformer [15] can better leverage the visual inductive priors. Therefore we utilize the Swin-Transformer as the backbone to extract image features and the Cascade Mask R-CNN [1] as the detection and segmentation head. To training the proposed model with only a few samples, data augmentation is a necessary step. Having trained the model, we conducted multiscale fusion to further boost the performance. Note that, all the training is done with only one GPU to simulate the situation of limited computation resources. In the end, our method achieved the AP@0.50:0.95 of 0.366 on the test set, which is competitive with other top-ranking methods while only one GPU is used. Besides, our method achieved the AP@0.50:0.95 (medium) of 0.592, which ranks second among all contestants.

2. Related Work

In this section, we briefly introduce two related topics: instance segmentation and transformer.

2.1. Instance Segmentation

Instance segmentation is essential for a wide variety of applications such as autonomous driving and visual question answering. Liu *et al.* [14] proposed Path Aggregation Network to boost information flow by bottom-up path augmentation and won the COCO 2017 Challenge Object Detection Task. Hybrid Task Cascade [3] is the first successful model to introduce the cascade into the instance segmentation field and improve the training results significantly. Lee *et al.* [10] presented the CenterMask, which is a simple yet efficient anchor-free instance segmentation framework. Cao *et al.* [2] presented a two-stage detection method called D2Det. The dense local regression and a discriminative RoI pooling scheme is introduced to achieve high quality object
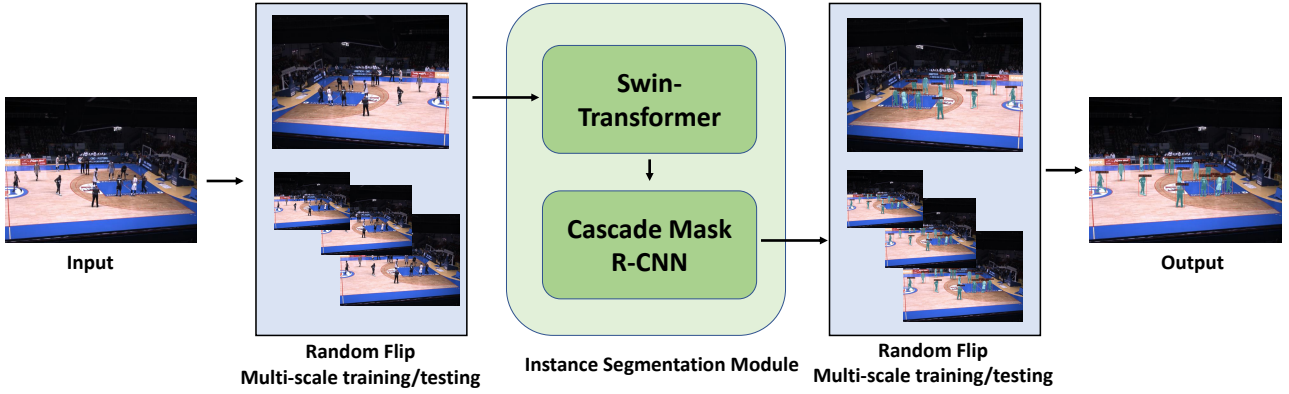

Figure 1. The overall framework.

Table 1. Results of different models on the test set.

Model	AP @0.50:0.95	AP @0.50	AP @0.75	AP@0.50:0.95 (small)	AP@0.50:0.95 (medium)	AP@0.50:0.95 (large)
Swin-S + Cascade Mask R-CNN	0.366	0.559	0.389	0.061	0.592	0.618
Swin-B + Cascade Mask R-CNN	0.362	0.583	0.382	0.063	0.603	0.605
Swin-T + Mask R-CNN	0.343	0.580	0.345	0.104	0.576	0.582
Swin-S + Mask R-CNN	0.352	0.553	0.378	0.058	0.598	0.588

detection and instance segmentation.

2.2. Transformer

Although the Convolutional Neural Networks have achieved great success in many computer vision fields [13, 11, 9, 12, 18], recent researches [5, 15, 20] show promising results of transformers. ViT [5] shows that a pure transformer framework can perform very well on image classification tasks. Swin Transformer [15] is proposed by Liu *et al.*, which presents a hierarchical vision transformer framework using shifted Windows. In this paper, we use Swin Transformer as the backbone to extract image features.

3. Methods

For a given image, we first use Swin-Transformer to extract image features, which can better exploit the visual inductive priors. Then we send the extracted image features to a Cascade Mask R-CNN model to detect and segment the objects. As mentioned earlier, we augment the database during training and testing to boost the performance. The framework s depicted in Figure 1.

3.1. Training Stage

In this part, we follow some ways of data augumentaion in the Swin Transformer, like the random flip of the pictures. And we also make some changes to the resizing strategy to

maximize the quality of the output with the limited memory size and computing resources.

As random flip is a common way of data augmentation to train deep neural networks [6], we adopt it for our method. Image scale matters. With enlarging the size of the images for training and testing, our model achieved a better performance for instance segmentation. Therefore we make the maximum of the image scale as large as possible during training. Besides, we also consider the multi-scale training strategy to augment the data. Instead of using some given values of scale in the training of the Swin Transformer, we consider a range of scales, and the scale of every image will be selected randomly in the range. More details are shown in 4.1.

3.2. Testing Stage

Multi-scale fusion is also a common way to improve the result of models, and it is quite popular in the computer vision field. For example, we can see the significant improvement brought by the fusion in [19]. Here we use multi-scale fusion for validation and testing. With the enlarged scale in the training stage, we can gain a much better outcome for more choices and larger scales.

4. Experiments

In this section, we conducted several experiments on the database provided by Synergy Sports, and we also tested

Table 2. Ablation study of multi-scale fusion for Swin-S + Cascade Mask R-CNN model. Results are reported on the validation set.

multi-scale	flip	AP @0.50:0.95	AP @0.50	AP @0.75	AP@0.50:0.95 (small)	AP@0.50:0.95 (medium)	AP@0.50:0.95 (large)
✗	✗	0.585	0.816	0.680	0.575	0.815	0.823
✓	✗	0.590	0.812	0.679	0.577	0.823	0.830
✗	✓	0.594	0.832	0.611	0.270	0.854	0.850
✓	✓	0.613	0.816	0.688	0.488	0.851	0.850

Table 3. Ablation study of multi-scale fusion for Swin-B + Cascade Mask R-CNN model. Results are reported on the validation set.

multi-scale	flip	AP @0.50:0.95	AP @0.50	AP @0.75	AP@0.50:0.95 (small)	AP@0.50:0.95 (medium)	AP@0.50:0.95 (large)
✗	✗	0.550	0.772	0.649	0.445	0.808	0.811
✓	✗	0.553	0.770	0.630	0.468	0.818	0.822
✗	✓	0.590	0.809	0.652	0.471	0.844	0.839
✓	✓	0.595	0.814	0.673	0.390	0.851	0.841

different parameters to get better results on the testing set.

4.1. Implement Detail

All experiments are conducted with the toolbox called MMDetection [4], based on PyTorch [16]. As we said earlier we only use one GeForce RTX 2080 Ti GPU. We use both the Swin-S + Cascade Mask R-CNN model and the Swin-B + Cascade Mask R-CNN model. In our experiments, they achieve similar results, in which the Swin-S one is slightly better.

For the super-parameters of Swin-B + Cascade Mask R-CNN model training, we use the two kinds of policies of image scales: one is directly resizing images in a range from (1080, 2500) to (1800, 2500), and the other is first resize the image to (900, 2500), (1125, 2500) or (1350, 2500), and then we random crop it into (864, 1400), and finally the image is re-scaled to a range from (1080, 2500) to (1800, 2500). For the validation and the testing stage, we choose two image scales (1400, 2500) and (1800, 2500) for fusion.

For the super-parameters of Swin-S + Cascade Mask R-CNN model training, we use the two kinds of policies of image scales: one is directly resizing images in a range from (1140, 4000) to (1900, 4000), and the other is first resize the image to (1013, 4000), (1267, 4000) or (1520, 4000), and then we random crop it into (912, 1600), and finally the image is re-scaled to a range from (1140, 4000) to (1900, 4000). For the validation and the testing part, we choose two image scales (1600, 4000) and (1900, 4000) for fusion.

The total epochs for both models is 250, and the batch size is 1. The initial learning rates of both models are set at 0.0001. Then they begin to decay at epoch 210 and epoch 240 with decay rate of 0.1. To fully use the database, we merge the training set and the validation set as the new train-ing set, which is allowed by the organizer. We don't use any outside data including the test set.

4.2. Result Analysis

For the Swin-B + Cascade Mask R-CNN model, we finally get an AP@0.50:0.95 of 0.362, and for the Swin-S + Cascade Mask R-CNN model we get an AP@0.50:0.95 of 0.366. The submitted result is achieved by the Swin-S + Cascade Mask R-CNN model, which is competitive with other top-ranking methods with only a single GPU. Besides, the Swin-S + Cascade Mask R-CNN model achieved the AP@0.50:0.95 (medium) of 0.592, which ranks second among all contestants. We report the detailed results on Table 1, which also includes the results of Swin-T + Mask R-CNN model and Swin-S + Mask R-CNN model.

We also conducted the experiments on the Swin-S + Cascade Mask R-CNN model and the Swin-B + Cascade Mask R-CNN model to show the effectiveness of multi-scale fusion during testing. The results are shown in Table 2 and Tabel 3, respectively.

5. Conclusion

In this paper, we have presented the key techniques we utilized for 2021 VIPriors Instance Segmentation Challenge. We use the Swin-Transformer as the backbone and the Cascade Mask R-CNN as the detection and segmentation head. Data augmentation and multi-scale trick are adopted for model training. Multi-scale fusion is employed to further boost the performance during inference. Our method achieves competitive results on the test set with a single GPU.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[2] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11485–11494, 2020.

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] 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 *ICLR*, 2021.

[6] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. On the importance of visual context for data augmentation in scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):2014–2028, 2021.

[7] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. *arXiv preprint arXiv:2105.01928*, 2021.

[8] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. *arXiv preprint arXiv:2012.07177*, 2020.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020.

[11] Wanhua Li, Jiwen Lu, Jianjiang Feng, Chunjing Xu, Jie Zhou, and Qi Tian. Bridgenet: A continuity-aware probabilistic network for age estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1145–1154, 2019.

[12] Wanhua Li, Jiwen Lu, Abudukelimu Wuerkaixi, Jianjiang Feng, and Jie Zhou. Reasoning graph networks for kinship verification: From star-shaped to hierarchical. *IEEE Transactions on Image Processing*, 30:4947–4961, 2021.

[13] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibing Ling. Cb-netv2: A composite backbone network architecture for object detection. *arXiv preprint arXiv:2107.00420*, 2021.

[14] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[17] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[19] Zhifeng Xiao, Yiping Gong, Yang Long, Deren Li, Xiaoying Wang, and Hua Liu. Airport detection based on a multi-scale fusion feature for optical remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1469–1473, 2017.

[20] Bingyao Yu, Wanhua Li, Xiu Li, Jiwen Lu, and Jie Zhou. Frequency-aware spatiotemporal transformers for video inpainting detection. In *ICCV*, 2021.