Delving into High Quality Action Recognition for Low Resolution Videos

Jianye He
DeepBlue Technology (Shanghai) Co., Ltd
hejianye@deepblueai.com

Zhiguang Zhang
DeepBlue Technology (Shanghai) Co., Ltd
zhangzhg@deepblueai.com

Zhenyu Xu
DeepBlue Technology (Shanghai) Co., Ltd
xuzy@deepblueai.com

Zhipeng Luo
DeepBlue Technology (Shanghai) Co., Ltd
luozp@deepblueai.com

Abstract

This paper introduces the solution to the low-resolution action recognition Task of the CVPR 2021 TinyAction Challenge from the DeepBlueAI Team. To realize the multi-label recognition for 26 different actions, we try some classical algorithms. Augmentation methods are adopted in both the training and inference phase, which leads to a comparable result. Cross-Validation and some Post Processing methods improve our score further. Finally, we rank the top one on the leaderboard.

1. Introduction

Recognizing tiny actions in videos is an important topic in the research of action recognition. The existing approaches addressing this issue perform their experiments on artificially created datasets where the high-resolution videos are down-scaled to a smaller resolution to create a low-resolution sample. However, real-world low-resolution videos suffer from grain, camera sensor noise, and other factors, which are not present in the down-scaled videos. The dataset of the CVPR 2021 TinyAction Challenge, TinyVIRAT[1], can serve as a benchmark dataset for tiny action recognition which contains natural low-resolution activities.

In this paper, we will share our solution to TinyAction Challenge. The key points of our solution can be summarized as follows:
1. Extensive experiments of different algorithms.
2. Suitable augmentation in the training and testing phase.
3. The usage of ensemble methods.
4. The tricky post-processing process.

2. Our solution

2.1. models

For the action recognition algorithm, we try TSM[3], TPN-Slowonly[6] and CSN[4]. TSM moves part of the channel along the temporal dimension; Thus facilitating the exchange of information between adjacent frames and achieving high efficiency and high performance simultaneously. TPN models the visual tempos of different actions effectively and can be flexibly integrated into 2D or 3D backbone networks in a plug-and-play manner. CSN explores the importance of interaction between channels and achieves a balance between saving calculation parameters and interaction between channels. The network has the advantages of simple structure, small computation, fast speed, good accuracy, and some regularization ability. Among them, CSN performs better than the other two. Our best single model result is got from interaction-reduced CSN(ir-CSN) with ResNet152[2] as the backbone.

2.2. Augmentation

In the training phase, we first apply RandomResized-Crop and then resize the frame sampled to the scale of 128x128. Another random flip is followed. Experiments show that scale is a factor that has a great impact on the result. We only try two kinds of scale, 128x128 and 70x70. The former has a much better score. Some augmentation methods, such as Mixup[7] and ColorJitter show no improvement. And we apply the non-local[5] module in the corresponding block, which doesn't work either.

In the testing phase, we use TenCrop as a kind of test time Augmentation(TTA), that is, crop the four corners and the center part of the image with the same given crop-size, and flip it horizontally. The final result is the average of the ten crops. We also try other TTA methods, like flip and

Model	F1 score
ir-CSN(**70x70**)	0.3942
ir-CSN(**128x128**)	0.4447

Table 1. Ablation experiments for scales

Model	F1 score
ir-CSN(**baseline**)	0.4447
ir-CSN + Mixup	0.4304
ir-CSN + ColorJitter	0.4222
ir-CSN + NonLocal	0.4352

Table 2. Ablation experiments

multi-scale, but get no improvement.

2.3. Ensemble

The popular ensemble method, five fold cross-validation(CV) is also adopted. We just combine the videos in both the training dataset and the validation dataset. Divide the data randomly into five equal parts. Use four of them to train while using the rest part for validation. Repeat the process four times to get four models trained by different data.

2.4. Post Processing

For the post-processing(PP), we try many different thresholds and choose the one which got the highest score. If there is no confidence score above the threshold after the sigmoid operation, we just keep the categories with the top2 scores. We can see that the 26 categories in the dataset can be grouped into 4 parts, activity, vehicle, specialized, and others. Within the same group, we think it's reasonable only one class can be retained at most. So we just keep the one with the highest score above the threshold within the same group.

3. Experiments

We deal with this competition as a multi-label task. So the BCE loss is adopted. We choose SGD as our optimizer and use warmup at the start phase of training. The total number of training epoch is 58. The initial learning rate is 1.25e-4 and decays by 0.1 at epoch 32&48. The weight decay is set as 1e-4 and the dropout ratio is 0.5. Sample 32 frames at a rate of one frame every two for every clip and only one clip every video segment. Based on ir-CSN, the performance on the test dataset of two different scales is shown in Table 1. Other ablation experiment results are shown in Table 2. And how we boost the performance step by step is listed in Table 3

Model	F1 score
ir-CSN(***1**)	0.4447
*1 + TenCrop(***2**)	0.4604
*2 + CV(***3**)	0.4722
*3 + PP	0.4782

Table 3. Milestone of score enhancement

4. Conclusion

We treat this competition as a multi-label action recognition task. The base algorithm, ir-CSN with ResNet152 as the backbone, can model different actions efficiently while mitigating overfitting. After the careful training of models, we use TenCrop to do a test time augmentation. Ensemble methods, 5fold cross-validation, is applied, coupled with a suitable threshold. Finally, some tricky post-processing methods lead to our results.

References

[1] Ugur Demir, Yogesh S Rawat, and Mubarak Shah. Tinyvirat: low-resolution video action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7387–7394. IEEE, 2021. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 1

[4] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 1

[5] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1

[6] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020. 1

[7] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1