Team Efficient Multi-Moments in Time Challenge 2019 Technical Report

Manyuan Zhang[1*] Hao Shao[1*] Guanglu Song[1] Yu Liu [2] Junjie Yan[1]
[1]SenseTime X-Lab
[2]Multimedia Laboratory, The Chinese University of Hong Kong
{zhangmanyuan,shaohao}@sensetime.com

Abstract

In this technical report, we briefly introduce the solutions of our team 'Efficient' for the Multi-Moments in Time challenge in ICCV 2019. We first conduct several experiments with popular Image-Based action recognition methods TRN, TSN and TSM. Then a novel temporal interlacing network is proposed towards fast and accurate recognition. Besides, the SlowFast network and it's variants are explored. Finally, we ensemble all the above models and achieve 67.22% on the validation set and 60.77% on the test set, which ranks 1st on the final leaderboard.

1. Image-Based Models

In this work, we have experimented with different 2D models including TSN [13], TSM [9], TRN [15]and TIN. These methods all use 2D convolution kernels instead of 3D convolution kernels to capture the temporal information. The number of their parameters and FLOPs are small compared to 3D-Based Models, but most of them don't have better performance than those 3D Networks.

1.1. Temporal Segment Network

Temporal segment network (TSN [13]) is a framework for video-based action recognition. TSN takes a strategy of sampling a fixed number of sparse segments from one video to model long-term temporal structure. The final prediction of video is averaged by the logits of each chip. In this competition, we experimented with the temporal segment network with evenly sampling 5 segments form one video.

1.2. Temporal Relational Network

Temporal relational network (TRN [15]) is a recognition framework that can model and reason about temporal dependencies between different segments of one video. The model is also designed to reason at multiple time scales. However, it doesn't work well in our attempt.

1.3. Temporal Shift Module

Temporal Shift Module (TSM [9]) proposes an operator that shifts part of the channels along the temporal dimension. The operator can help the network fuse the temporal information among neighboring frames. We experimented the model with different backbones and input sequence lengths T.

1.4. Temporal Interlacing Network

In this work, we proposed a Temporal Interlacing Network (TIN) which uses a network to model the relation between the shift distance and the specific input data. While TSM can only shift the channels along the temporal dimension by +1 or -1. The differentiable module we designed can infer the suitable displacement length according to different groups and suitable weight for the feature-map along the temporal dimension. Our proposed module has almost the same FLOPs and parameters as the origin TSM model. Moreover, in our experiments between TSM and TIN, TIN obtained about 1% - 2% better performance with the same train and test configure.

2. SlowFast-based Models

In this part, we conduct experiments on SlowFast [3] network. SlowFast network has two paths, a slow path to capture appearance content while a fast path to capture motion information. For details about the architecture, please refer to its raw publication [3].

For this challenge, we train several SlowFast models and its variants. Note that only RGB input is used, for the reason that flow extraction costs too much computation and storage. The models we select are (a) SlowFast, Slow path 8×8 and Fast path 32×2, with an input clip of consecutive 64 frames (b) only Fast path 32×2, with no channel reduction. This model is pretty heavy. It has above 4x computation consumption than the standard SlowFast network.

*They contributed equally to this work.

(c) only Slow path 8×8, which only keeps the slow path to capture appearance content. (d) SlowFast, Slow path 11×8 and Fast path 44×2. Due to most videos has around 90 frames, this model designs to capture the whole video information.

3. Experiments

Dataset. The Multi-Moments in Time datasets [11] consists of more than one million three-second videos with two million action labels. This dataset focus on multi-label action recognition in videos. There are 1,025,862 training , 10,000 validation and 10,000 test videos segments. We use FFmpeg to extract frames with 30 fps and resize the short side to 320 pixels. Note that 737 training videos, 8 validation videos, and 6 test videos can't extract frames. The performance metric is mean Average Precision (mAP) under sample level.

Loss function. For the multi-label action recognition problem, there exist high relationships between class labels. Proper loss function will capture the relationship and leading to better convergence. We try several multi-label classification loss functions including Warp [8], Lsep [14], and most used binary cross-entropy loss. During the experiments, we find that simply scale up BCE loss achieves a huge performance gain. But scale up learning rate with the same ratio can not have the same performance. Scale-up BCE loss which uses 'mean' as a reduction operator in PyTorch with factor 160 achieves the best performance and we use it as our loss function.

Class unbalance. we count the sample number for each class, the max number of videos in one class is 48060 while the min is 504. To tackle the training sample unbalance problem, we give each class a loss weight. Denotes the mean number of videos in a class as N_{mean}, number videos in class i as N_i. We have two strategies, weighted by $\sqrt{\frac{N_i}{N_{mean}}}$ and $\frac{N_i}{N_{mean}}$. Both of them achieve slightly performance gain, so we directly use original data to train, without handling the class sample unbalance problem.

Model pre-train. There are several works claims that pre-train is important for action recognition [2, 4]. We pre-train models on Kinetics dataset, which is the largest scale well-label action recognition dataset. Pre-train improves Image-based methods but it is interesting that pre-train on Kinetics hurt recognition performance for SlowFast-based methods (1.1% drop). The reason may be that Kinetics focuses on human activities but Multi-Moments In Time dataset includes a wider range of actions like animal and nature motion.

Training. For the training of Image-Based Models, we evenly select 8 or 16 frames from a video and the spatial size of the input image is 224×224 pixels. About the data augmentation, we use the technique of spatial jittering and horizontal flipping to alleviate the problem of overfitting. We set the dropout rate [12] to 0.5 and set weight decay to 5e-4. Meantime, we use the algorithm of mini-batch stochastic gradient descent with a momentum of 0.9. The initial learning rate is set to 0.01 and divided by 10, 20 epochs.

Table 1. mAP on the validation set of the Multi Moments in Time Dataset.

Model	Frames	Backbone	mAP
TSM	8	ResNet-50	59.85
TSM	16	ResNet-50	61.12
TSM	8	ResNet-101	61.06
TIN	8	ResNet-50	62.23
TIN	8	ResNet-101	62.22
TIN	16	ResNet-101	62.49
TSN	5	ResNet-101	58.92
TSN	5	IC-ResNet-v2	57.45
TSN	5	SeNet-154	53.19

For training the SlowFast-based model, we use the half-period cosine learning rate [10] for training. Learning rate at k-th iteration is $\eta \cdot 0.5 \left[\cos\left(\frac{\dot{n}}{n_{\max}}\pi\right) + 1\right]$, the base learning rate is 0.2 and the max iteration number is 180k. We also adopt linear warm up [5] for stable convergence. The momentum is 0.9 and weight decay is 10^{-4}. Dropout [7] is set to 0.5 to prevent over-fitting. 32 Nvidia V100 GPUs with synchronized SGD and synchronized BatchNorm is used. The mini-batch size in one GPU is 32, 1024 in total. The training process usually lasts for a week. As for data augmentation, we first random sample video clips with consecutive frames. For the spatial domain, we randomly resize the video frame short side to [128,160] and then randomly crop 112×112 pixels. Randomly horizontal flip is also used.

3.1. Results for Image-Based Models

In this competition, we use ResNet-50 [6], ResNet-101, Inception-ResNet-v2, Senet-151 as our backbone models, which are pretrained on Kinetics-600 [1]. The mAP on the validation set of the Multi Moments in Time are shown in Table. 1 We can find that models with more input frames have better performance, and TIN achieved 2.37% better performance compared to TSM with the same 8 frames and backbone network. From Table. 1, we can also find that TSN still obtained a not bad performance with ResNet-101 backbone networks.

We also try to sample more chips from one video, it can also help the model get higher mAP on the validation set. In detail, we sample 5 random chips along the temporal dimension. With respect to the spatial dimension, we also random crop 3 chips which the size of it is 256×256 pixels. These tricks can help our models achieve about 0.5%-1.1% improvement compared with using only one chip to test.

	mAP@clip	mAP@video (dense testing)	mAP@video (multi-scale & dense testing)
SlowFast, 8×8	59.89	61.52	63.52
SlowFast, 11×8	60.41	61.60	64.00
Fast,16×2	59.05	61.55	63.45
Fast, 32×2	62.03	63.59	**64.81**
Slow, 8×4	59.02	61.68	62.99
Ensemble	66.71 (val)		59.80 (test)
Ensemble+Image-based methods	**67.22** (val)		**60.77** (test)

Table 2. Results for SlowFast-based networks on the validation set. By ensemble with Image-based methods, we achieve 67.22% on validation set and 60.77% on test set, which ranks 1st on the final leaderboard.

3.2. Results for SlowFast-Based Methods

In this competition, we select SlowFast-101 as base model. Deeper network SlowFast-152 and SlowFast-200 is also explored, but they achieve near the same performance as SlowFast-101, so we finally conduct experiments base on SlowFast-101. The results for SlowFast network and its variants are shown in Table. 2. From Table. 2, we can find that only Fast path with 64 consecutive frames input achieves the best performance, due to its huge computation.

We also conduct multi-scale and multi-crop inference. For each test video, we uniformly sample 10 clips along its temporal axis. For each clip, we scale the shorter size to 128 and random crop 3 clips with 128×128 pixels from the spatial domain. A multi-scale crop like 144 and 160 is also applied for the robust test. For all crop clips from a video, class prediction probabilities are average. Results are list in the Table. 2. The multi-scale and dense inference strategy leading to around 1.5% 2.5 performance gain.

4. Conclusion

In our submission to the Multi-Moments in Time Challenge 2019, we experiment with many popular video action methods including TSN, TRN, TSM, and SlowFast. A novel Image-based recognition model TIN is proposed to achieve fast and accuracy trade-off. For this work, only RGB input is used, multimodal input including flow and audio will be explored in the further work. A more sophisticated loss function can also be studied for multi-label action recognition problem.

References

[1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

[4] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019.

[5] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[6] 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.

[7] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[8] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3617–3625, 2017.

[9] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *arXiv preprint arXiv:1811.08383*, 2018.

[10] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[11] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[13] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[14] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[15] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018.