Few-Shot Learning of Video Action Recognition Only Based on Video Contents Supplementary Material

Yang Bo Yangdi Lu Wenbo He
McMaster University
boy2@mcmaster.ca

1. Introduction

We provide the hyper-parameters evaluation of Temporal Attention Vectors (TAVs) on UCF101. We also evaluate the TAVs on the Diving48 dataset with two backbones (frame/clip features extractor), ImageNet pre-trained resNet-152 [2] and Kinetics 400 pre-trained I3D [1] networks. Both backbones are not trained with Diving48 during the entire evaluation.

2. Diving48 Dataset

The newly released Diving48 dataset [3] which contains $15,943$ training and 2096 testing videos of professional divers performing 48 types of dives. We choose this dataset because unlike datasets such as Kinetics or UCF101, Diving48 is designed to minimize the bias towards particular scenes or objects.

3. Implementation details

During the following evaluations, we use the Short-long dynamic Gaussian (SLDG) to initialize the TAVs and use the Short-long term Fusion to fuse the TAVs for different temporal terms. The training and testing procedures are the same as we explained Sec. 4.3 in the paper. All evaluations of hyper-parameters are performed with both RGB and optical flow as input and the evaluation on Diving48 are performed with only RGB frames as input.

resNet-152 backbone. The input configuration is the same as we explained in Sec. 4.1.

I3D backbone. The videos are resized preserving aspect ratio so that the smallest dimension is 256 pixels, with bilinear interpolation. We use $10\times$ data argumentation with randomly select an 224×224 image crop. Pixel values are then rescaled between -1 and 1. We use video clip with 8 continuous frames as input. If the video length is not divisible by 8, we just simple discard the rest frames. We modify the kernel size of 3D average pooling layer to 1×7 and use the output as clip features (1024D).

Base I3D. The input configuration is the same as the I3D backbone except we use 64 frames video clip as the input. We follow the frame selection approach in [5]. First split the whole video into 64 chunks. If the video which has insufficient frames we just simply repeat the video multiple times. Then from each chunk, we uniformly choose a frame to construct the video clip. We only retrain the last layer on Diving48 dataset.

Importance Score Learner We use the inflate-shrink during the evaluations. The first convolution layer has 16 1 filters and others remain the same as indicate in the paper.

# vectors K	# iteration N	# chunks m_n	Accuracy
7	2	6, 1	78.4
8	2	6, 2	78.3
9	2	6, 3	77.7
9	3	6, 2, 1	75.4
9	2	8, 1	77.4
10	2	8, 2	77.1
12	2	8, 4	76.7
14	3	8, 4, 2	76.5
15	4	8, 4, 2, 1	75.7
17	4	8, 6, 2, 1	75.1
17	2	16, 1	76.9
18	2	16, 2	76.5
20	2	16, 4	76.1
21	2	16, 5	76
21	3	16, 4, 1	76.6
24	2	16, 8	74.8
28	3	16, 8, 4	75.9

Table 1: Average accuracy (%) on the UCF101 split 1 (the experiments are repeated 10 times and each time 10 training videos are uniformly chosen for each class) of SLDG with different parameters. The short-long term fusion is applied.

3.1. Analysis of the SLDG Vector

We introduce the SLDG initialized TAVs in Section 3 which is generated by repeating the DG initialization N times and each time the number of chunks is set to m_n. As the number of iteration increases, we highlight more information on different temporal range. In this section, we

focus on finding a good combination between the iteration number N and the set of chunk numbers m.

The results are shown in Table 1. We first focus on the top and middle parts of the table. We notice that i). When the iteration number N does not change, as the value of K increases, the accuracy actually decreases. For example, the first three rows in the top and middle parts of Table 1 indicate the accuracy drops along with an increase of K. This is because the long-term information is not highlighted (when $m_n = 1$ or 2). ii). There is a negative correlation between the iteration number N and the accuracy when the number of SLDG K remains the same. For example, row 3 and row 4 show the accuracy decreases by 2.3% as N goes up to 3 from 2 when $K = 9$. Then we move to the bottom part of Table 1. However, the second observation is not valid when $m_1 = 16$. The fourth and fifth rows in the bottom part of the table show an increase in accuracy as the iteration number N goes up when $K = 21$. This is because the SLDG vectors with chunk numbers $(16, 5)$ only highlight the short-term temporal information but ignore the long-term one.

3.2. Analysis of Different Initialized TAVs with Different Fusion Approaches

In Section 3 and Section 4.2, we discuss four initialization approaches of TAVs to encode the frame features and three two-stream fusion approaches. The goal here is, given few labeled training data (*e.g.* 10 training video per class), finding the best-performed combination of the TAVs number and the fusion approach for the TAVs initialization approaches. Table 2 lists the fusion approaches (early concatenation, late fusion or short-long term fusion), the TAVs initialization approaches (Random, SSD, DG or SLDG) and the number of the TAVs that is used. The value of θ is set to 2 for DG TAVs and we sort the SSD TAVs with ascending order since we find the order does not impact their performance. Note that, the short-long fusion is only applicable to the SLDG TAVs.

We first focus on the performance of the TAVs with different fusion approaches. Not surprisingly, the simple early concatenation outperforms the late fusion over all TAVs initialization approaches since the feature extractor is only trained with RGB images (*e.g.* ImageNet), the imprecise optical flow features result misprediction. For the SLDG initialization, the short-long term fusion provides marginal improvements (about 1%) than the early concatenation when $m = (2, 1)$ and $m = (4, 2)$. Then we investigate the effectiveness of using different number of the TAVs. Increasing the number of them leads to a small improvement (0.6%) for the random initialized TAVs and a significant boost for the DG TAVs (3.6%). However, as the number of TAVs increases, the accuracy slightly drop for the SSD and SLDG TAVs. We also perform the same experiments on UCF101 split 1 with 20 training video per class and we get the same results.

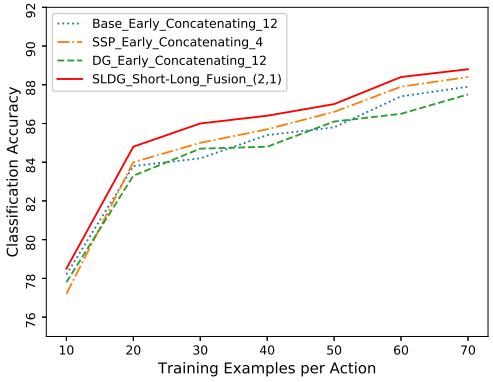


Figure 1: Accuracies (%) on UCF101 split 1 with change in the size of the labeled training set. The experiments are repeated 10 times and each time the training videos are uniformly chosen for each class.

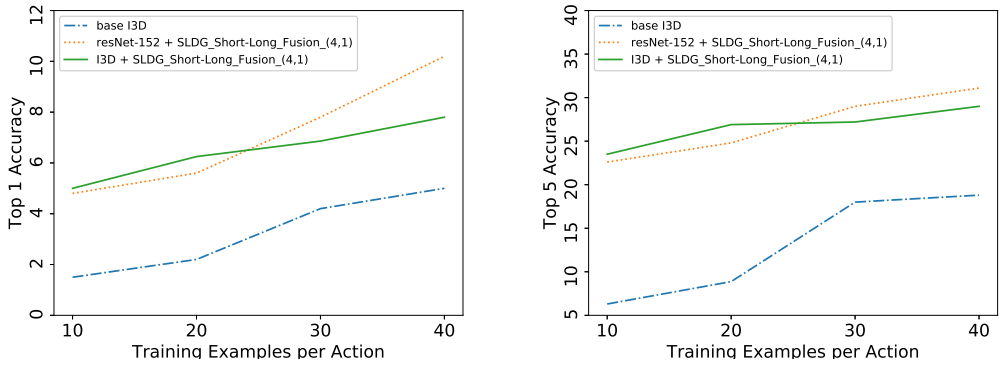
We choose the best-performed combination of the TAVs number K and fusion approaches for each type of the TAVs and evaluate their performance with various size of the training set. The results are shown in Figure 1. Using SLDG TAVs with $m = (2, 1)$ and short-long term fusion outperforms other combinations over all training sets.

4. Evaluation of TAVs with Very Few Training Data of Diving48

In this section, we compare TAVs using resNet-152 and I3D as backbones with base I3D on Diving48 with change in the size of the labeled training set. All experiments are done with the same training sets. The top 1 and top 5 average accuracy are shown in Figure 2a and Figure 2b, respectively. Clearly, the accuracy has a significant gap between the base I3D and the I3D+TAVs. For example, with only 10 labeled training video per class, the top 1 accuracy of I3D+TAVs achieves 5.5% on Diving48 which are 3.3% higher than the accuracy of base I3D network. If we compare the top 5 accuracy, the gap even becomes larger (17.13% with 10 labeled training videos). Overall, I3D+TAVs outperforms the base I3D network with very few training videos. The interesting thing is, even we use a less powerful backbone (resNet-152), the TAVs still outperform the base I3D network with very few training data. Also, the performances of resNet-152+TAVs and I3D+TAVs are really close which shows the TAVs could effectively encode the temporal information of videos with both 2D or 3D backbones.

Temp. Vectors / Fusion	Base			SSD			DG			SLDG		
	K=4	K=8	K=12	K=4	K=8	K=12	K=4	K=8	K=12	$K=3$, m=(2,1)	$K=6$, m=(4,2)	$K=12$, m=(10,2)
Early Concat.	77.5	77.8	**78.1**	**77.2**	76.9	76.2	74.2	73.1	**77.8**	77.6	77.3	77.2
Late Fusion	75.4	71.4	72.4	74.3	68.2	74.2	72	71	72.4	72.8	72.9	72.7
Short-Long term Fusion	-			-			-			**78.5**	78.4	76.8

Table 2: Average accuracy (%) on the UCF101 split 1 (the experiments are repeated 10 times and each time 10 training videos are uniformly chosen for each class) of different combinations of TAVs with different numbers and fusion approaches.



(a) Top 1 Accuracy, RGB

(b) Top 5 Accuracy, RGB

Figure 2: Comparisons of TAVs using different backbones with I3D for action recognition with change in the size of the labeled training set on Diving48. The experiments are repeated 10 times and each time the training videos are uniformly chosen for each action class. The average accuracy (%) are reported.

Methods	Framework	Input	Pre-train	Full-FT	Accuracy
TSN [5]	2D	RGB	ImageNet (objects)	Yes	16.8
TSN [5]	2D	RGB+FLOW	ImageNet (objects)	Yes	20.3
TSN+TAVs (Ours)	2D	RGB	ImageNet (objects)	No	22.1
I3D [1]	3D	RGB	Kinetics (actions)	No	12.2
R(2+1)D [4]	3D	RGB	Kinetics (actions)	Yes	28.9
I3D+TAVs (Ours)	3D	RGB	Kinetics (actions)	No	20.5

Table 3: Comparison with the state-of-the-art methods on the Diving48 dataset. The full-FT: "yes" indicates end-to-end fine-tuning on Diving 48, "no" means only train the last layer for I3D or the importance score learner of TAVs.

5. Comparison with the state-of-the-art on Full Diving48

Finally, we compare TAVs to the state-of-the-art action recognition results on Diving48 datasets. The performance is summarized in Table 3. The accuracy shows significant improvement when using TAVs with 2D framework. For example, the Temporal Segment Network(TSN)+TAVs outperform 5.7% than the end-to-end fine-tuning TSN network when using only RGB as input, even the backbone network never see the video in Diving48. The end-to-end fine-tuning seems much important for 3D framework than 2D framework. For example, the end-to-end fine-tuning R(2+1)D [4] network gives 28.9% classification accuracy which is higher than I3D+TAVs. We believe the reason is that the I3D backbone are not trained on Diving48 since the I3D and R(2+1)D networks show similar performance on other video action recognition datasets (*e.g.* Kinetics). However, when using TAVs on top of I3D, 8.3% accuracy improvement is still achieved comparing with the base I3D network.

6. Conclusions

In this supplementary material, we provide the detailed evaluation of TAVs on Diving48 dataset. We build the TAVs on both 2D (resNet-152) and 3D (I3D) backbones. The significant improvement is achieved on both backbones. The TAVs show the best performance with very few training videos. Also, compared to other state-of-the-art frameworks on full diving48 dataset, the TAVs still shows competitive results, especially with 2D backbone.

References

[1] J. Carreira and A. 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.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Y. Li, Y. Li, and N. Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018.

[4] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[5] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.