Learning Spatiotemporal 3D Convolution with Video Order Self-supervision

Tomoyuki Suzuki[1(✉)], Takahiro Itazuri[2], Kensho Hara[1], and Hirokatsu Kataoka[1]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
{suzuki-tomo,kensho.hara,hirokatsu.kataoka}@aist.go.jp
[2] Waseda University, Tokyo, Japan
s132800732@fuji.waseda.jp

Abstract. The purpose of this work is to explore self-supervised learning (SSL) strategy to capture a better feature with spatiotemporal 3D convolution. Although one of the next frontier in video recognition must be spatiotemporal 3D CNN, the convergence of the 3D convolutions is really difficult because of their enormous parameters or missing temporal(motion) feature. One of the effective solutions is to collect a 10^5-order video database such as Kinetics/Moments in Time. However, this is not an efficient with burden of manual annotations. In the paper, we train 3D CNN on wrong video-sequence detection tasks in a self-supervised manner (without any manual annotation). The shuffling and verification of consecutive video-frame-order is effective for 3D CNN to capture temporal feature and get a good start point of parameters to be fine-tuned. In the experimental section, we verify that our pretrained 3D CNN on wrong clip detection improves the level of performance on UCF101 (+3.99% better than baseline, namely training 3D convolution from scratch).

Keywords: 3D Convolutional Neural Network · Self-supervised learning · Motion feature · Human action recognition

1 Introduction

Spatiotemporal 3D Convolutional Neural Network (3DCNN) have been successful in video understanding and it is expected to further develop [4]. Recent researches [1,4] have shown that 3DCNN have beat 2DCNN [11], which are conventional state-of-the-art methods in video recognition task. Against to the difficulty of 3D conv optimization, there are mainly two factors as follows: (i) Carreira *et al.* proposed a parameter inflation [1] which is a method for knowledge transfer from 2D pretrained model into 3D initialized parameters, (ii) large-scale (over 10^5-order) and clearly annotated video datasets have been released [6,10].

However, while inflation makes *appearance* feature easier to capture for 3DCNN, the suspicion that 3DCNN does not capture the *motion* (temporal)

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11130, pp. 590–598, 2019.
https://doi.org/10.1007/978-3-030-11012-3_45

feature effectively. In the recent study, Huang *et al.* pointed out that the 3DCNN with consecutive frames only selects effective frame(s) to contribute the video classification [5].

On one hand, Two-Stream 3DCNN, which is ensemble method of RGB- and flow-input achieved significantly better performance than single RGB-input in action recognition [1] (Two-stream 98.0 vs. RGB-stream 95.6 on UCF101). This result indicates that a 3DCNN with consecutive RGB frames cannot completely capture a motion feature like optical flows. We believe that acquiring motion feature like optical flows with 3DCNN from only RGB-input is the key to further progress of 3DCNN in action recognition.

Also, manual annotations are time-consuming. To effectively optimize 3DCNN, we require over 10^5-order video database such as Kinetics. The burden of human labeling has contributed to the advanced video recognition model, however, any further annotations are obstacles of training of video understanding. Namely, we must consider an effective learning method for 3D convolution without any human supervisions.

In this study, we focus on self-supervised learning (SSL) with video order to optimize 3D convolution without any manual annotations on pretrained 3DCNN. We here propose the shuffling and detecting of wrong video order to learn a 3D convolution. The video order is a high-confidence context to enoughly train 3D convolutional filters. According to the experimental results, we confirm that easily solvable SSL tasks can improve an optimization of 3DCNN. The performance rate with 3DCNN on UCF101 is improved with our self-supervision. Our self-supervision is +3.99% better than a baseline, training 3DCNN from scratch.

2 Related Works

2.1 CNN Based Action Recognition on Videos

One of the most popular approach to action recognition is Two-Stream 2DCNN [11]. This is the ensemble of spatial-stream which takes RGB frame as input and temporal-stream which takes stacked optical flow frames as input. Spatial-stream can get the benefits from ImageNet pretraining which captures strong appearance feature, while temporal-stream captures motion feature by hand-craft optical flow and temporal stacked input. Also, the variants of this model were proposed.

One of the recent trends of action recognition is the use of 3D convolution to capture spatio-temporal feature. Tran et al. trained (relatively shallow) 3DCNN on Sports 1M dataset to extract spatio-temporal feature from videos [12]. However in several years ago, unlike 2DCNN which has several large-scale image datasets, optimization of deeper architecture of 3DCNN was difficult due to the luck of the dataset including sufficient *video* instances. In recent study, several huge datasets [6,10] were released. In addition, Inflation [1], which is the method to transfer appearance knowledge from pretrained 2DCNN by expanding 2D kernels into 3D was proposed. Under favor of these factors, as deep 3D architectures as 2D became able to provide powerful performances.

On the other hand, Huang et al. provided interesting discussion that 3DCNN does not necessarily capture *temporal* information well in action recognition [5]. From the results of their experiments, it can be considered that 3DCNN focuses on key frame selection from *set* of images rather than motion information from video clip, namely relationship between frames. Moreover, Two-Stream 3DCNN, which is ensemble method of RGB input and optical flow input achieved significantly better performance than only RGB input [1]. This result indicates that from consecutive RGB frames 3DCNN can not capture complete motion which contributes to classify actions.

Based on these insights, we believe that motion information can be still disputable to explore in contrast to appearance one and contribute to further progress of 3DCNN. In this research, we provide experiments about influence of pretraining procedures which encourage 3DCNN to consider relationship between frames.

2.2 Self-supervised Representation Learning

The main goal of self-supervised learning (SSL) is to transfer a model trained on a pretext task which is defined without manual annotation to a target task. Because of no cost of annotation and fine-tuning which is compatible transfer strategy, self-supervised learning is one of the focuses of attention in computer vision. Many of pretext tasks are defined using image data. For instance, colorization, inpainting, solving puzzles and counting. Another direction is the use of video-sequence. Misra et al. trained SiameseNet to verify the order of video frames [9] and Lee et al. trained to sort frames in correct order [8]. While their approaches focus on capturing appearance feature from single frame as a result of video-sequence based pretext task, in order to capture motion feature Fernando et al. trained a model to verify video-order taking multiple subtraction images of consecutive frames as input [2]. While they used 2DCNN, currently it is said that 3DCNN is more suitable for capturing spatio-temporal feature [5]. In this research, we focus on exploring the potential of SSL using video-sequence towards motion feature on 3DCNN.

3 SSL for 3D CNN

In order to explore self-supervised learning strategy towards motion feature, we first trained a 3DCNN model on pretext task and fine-tune it on action recognition task without freezing any weights of layers. This means that in this experiments we assume capturing efficient representation provides good start point of optimization.

3.1 Pretext Tasks

According to previous works [2,8,9], we define classification task based on SiameseNet which is consist of several branch networks with shared weights and one

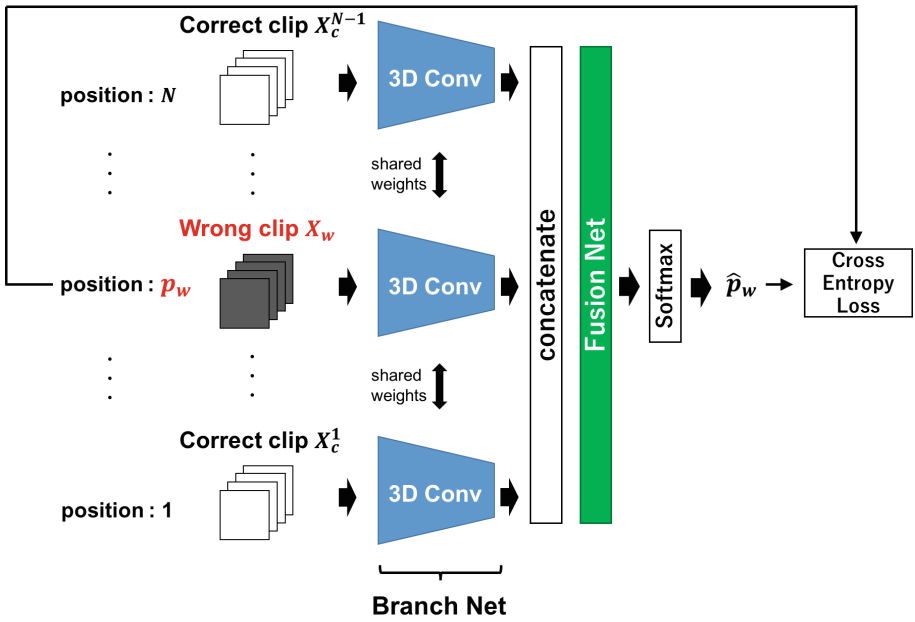

Fig. 1. Overview of pretraining strategy: Each branch network (3DCNN) takes a clip from a set of wrong one and several correct ones as input and their outputs are concatenated as input of fusion network. Here, N represents total number of input clips as well as branch network. And then, fusion network outputs estimated wrong position $\hat{p_w}$ In each question we permutate branch positions where each clip is input. Once we define the scheme of making a wrong clip, all of this task can be designed without any manual annotation.

fusion network (see Fig. 1). In this research, each branch network takes a different clip from wrong one (X_w) and several correct ones ($\{X_c^i\}_{i=1}^{N-1}$) as input and their outputs are concatenated as input of fusion network. Here, N represents total number of input clips as well as branch network. To establish problem, in each question we permutate branch positions where each clip is input, and obtain the position of a wrong clip $p_w \in \{1, ..., N\}$. And then, fusion network predicts p_w, as N-way classification problem. Note that while a wrong clip is modified in some way on temporal order (discussed below), a correct one is not modified at all.

Finally, we train a model by standard maximum likelihood estimation. Given X^j and p_w^j are j-th samples of permutation of clip set $\{X_w, X_1, ..., X_{N-1}\}$ and corresponding wrong position,

$$\theta^* = \arg\max_{\theta} \sum_j \mathcal{L}(f_\theta; X^j, p_w^j) \tag{1}$$

where $\mathcal{L}$ is likelihood function (cross entropy), and f_θ is our model parameterized by θ. Since we focus on motion feature, pretext tasks should be solved by *motion*

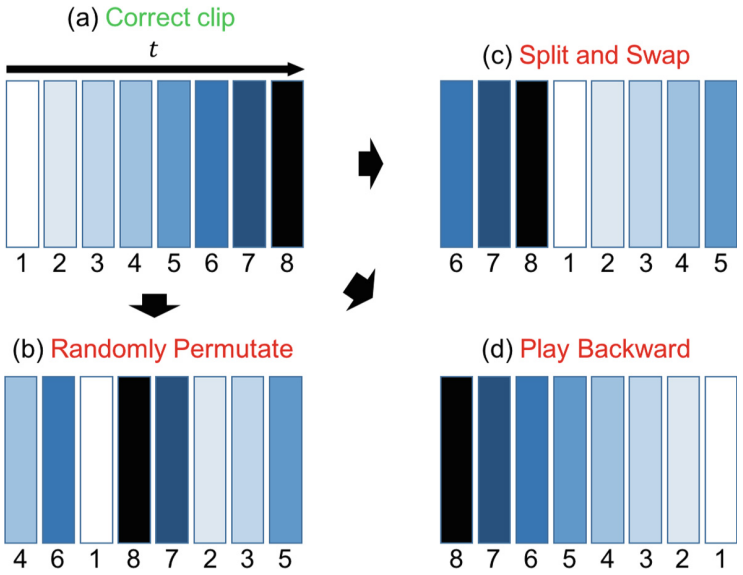

Fig. 2. The settings of wrong clip: The rectangles denote the frames in a clip ($D = 8$) and rectangles which have same color and index indicates that each of them is identical frame. We make three types of wrong clips from a correct ordered clip (a). (b) Randomly Permutate. (c) Split and Swap where p_s is set to 6. (d) Play Backward.

cue in input clips. With this in mind, we define several settings of wrong clip using unlabeled videos as below (see also Fig. 2).

Randomly Permutate (RP). This setting is similar to Fernando's [2], where they aimed at capturing video representation by 2DCNN. We first select a video clip of size W as a constrained window, then we sample different N clips including D consecutive frames from this window. We randomly choose one of the clips as a candidate of wrong clip $X'_w = \langle x_1, ..., x_i, ..., x_D \rangle$ where x_i denote the i-th frame of the clip. Then we permutate its frames randomly as a wrong clip X_w, while the rest of clips are correct clips. Note that since wrong clips differ in only temporal order, a model is required to utilize temporal information in other to detect wrong one. In this setting, our major concern is that a model captures relatively low-level temporal feature (e.g. detecting temporal blur) only, not high-level (semantic) feature which is intuitively effective for action recognition.

Split and Swap (SS). As with RP setting, we sample one candidate of wrong clip X'_w and $N - 1$ correct clips from constrained window. In this setting, we randomly choose a split position $p_s \in \{2, ..., D - 1\}$, and swap two split clips to make a wrong clip.

$$\begin{aligned} X_w &= \mathrm{SS}(X'_w, p_s) \\ &= \langle x_{p_s}, ..., x_D, x_1, ..., x_{p_s - 1} \rangle \end{aligned} \tag{2}$$

In contrast to RP setting, a wrong clip has only one temporal inconsistency, and it is considered that this makes pretext task somewhat more difficult than RP.

Play Backward (PB). In this setting, we play wrong clip backward, in other words apply temporal-flip.

$$\begin{aligned} X_w &= \mathrm{PB}(X'_w) \\ &= \langle x_D, x_{D-1}, ..., x_2, x_1 \rangle \end{aligned} \tag{3}$$

This is inspired by Gidaris's research [3], where they defined classification of image rotation as pretext task. The intuition behind this setting is related to the fact that in other to detect temporal-flipped videos a model is forced to capture more semantic information. In more detail, a model localizes moving subject (e.g. human, car, animal), recognizes the type and pose (e.g. where is it facing) of them, detects the direction they are moving to, and then results in successful classification. Specifically, since we use the videos for human action recognition, the subject a model detects may be often human. The most important concern of this setting is possibility of ambiguity. The distributions of videos belonging to some classes (e.g. jumping on the spot) is much the same as that of temporal-flipped videos and this means it is difficult for a model to solve the task.

3.2 Implementation

Following previous experiments of SSL [2,3,8,9] where AlexNet [7] is used as a baseline model, we construct 3D-AlexNet. The detail of architecture is shown in Table 1. Based on original architecture, 2D convolutional and pooling kernels are expanded to 3D spatio-temporal kernels, and dropout layers are replaced by batch normalization layers. This model takes 8 frames as input. During self-supervised learning, to construct SiameseNet, we use layers from `conv1` to `pool3` (denoted by `conv1-pool3`) of 3D-AlexNet followed by `fc4` as branch model, and `fc5`, `fc6` as fusion model. Training is run with $W = 20$, $D = 8$ and $N = 6$ on UCF101 train set of split1. The initial learning rate is 0.0001 and batch size is 64.

When fine-tuning on action recognition, we initialize 3D-AlexNet with pre-trained `conv1-pool3` and random initialized `fc1-fc3` followed by softmax function. We fine-tune the 3D-AlexNet on UCF101 train set of split1 and set the initial learning rate to 0.001 on `fc6-fc8` and 0.0001 on `conv1-pool6`, and batch-size to 64.

During final evaluation, we sample all non-overlapping clips from UCF101 test set of split1 as input and compute accuracy for each clip (clip-accuracy) from the maximum conditional probabilistic estimate. Also, to compute accuracy for each video (video-accuracy), we calculate arithmetic mean of the output from all clips in the video, as conditional probabilistic.

Table 1. Network architecture: In our experiments, $C = 101$ and $N = 6$.

Model	Layer name	Kernel		Stride
		Size	Channel	
3D-AlexNet	`conv1`	$(5, 11, 11)$	$3 \rightarrow 64$	$(1, 4, 4)$
	`pool1`	$(2, 3, 3)$	–	$(2, 2, 2)$
	`conv2`	$(3, 5, 5)$	$64 \rightarrow 192$	$(1, 2, 2)$
	`pool2`	$(2, 3, 3)$	–	$(2, 2, 2)$
	`conv3`	$(3, 3, 3)$	$192 \rightarrow 384$	$(1, 1, 1)$
	`conv4`	$(3, 3, 3)$	$384 \rightarrow 256$	$(1, 1, 1)$
	`conv5`	$(3, 3, 3)$	$256 \rightarrow 256$	$(1, 1, 1)$
	`pool3`	$(2, 3, 3)$	–	$(2, 2, 2)$
	`fc1`	–	$9216 \rightarrow 4096$	–
	`fc2`	–	$9216 \rightarrow 4096$	–
	`fc3`	–	$4096 \rightarrow C$	–
for SiameseNet	`fc4`	–	$9216 \rightarrow 128$	–
	`fc5`	–	$128N \rightarrow 128N$	–
	`fc6`	–	$128N \rightarrow N$	–

4 Results and Discussion

The results are reported in Table 2, where "Scratch" means training 3D-AlexNet on UCF101 without any pretraining and "Inflated" represents inflating from 2D version pretrained on ImageNet as a guide.

Table 2. Results.

Setting	Pretrain acc.(%)	Fine-tune acc.(%)	
		Clip	Video
Scratch	–	40.98	45.21
RP	99.21	**44.96**	**49.20**
SS	98.00	44.75	49.07
PB	17.65	40.01	44.97
Inflated	–	55.12	60.13

First, focusing on accuracy of pretext tasks, RP and SS setting are performed in high accuracy and seem to be solved almost perfectly. From this observation, it is expected that these settings is too easy for 3DCNN to capture efficient motion feature. On the other hand, PB is performed in near chance rate ($1/6 \approx 16.67\%$), and this implies that it is too difficult because of abovementioned ambiguity.

In spite of this expectation, as we can see in results of fine-tuning, pretraining on RP and SS improves performance of action recognition compared with Scratch, even though still can not compete against Inflated. This improvement implies that low-level and easily available temporal feature can contributes to better optimization of 3DCNN, and conversely this also means it is possible that ordinary training strategy of action recognition misses this feature. Meanwhile, we can not observe improvement by PB. The cause of this is considered to be the fact that 3DCNN can not detect wrong clips. We confirm that certain performance on pretext task is a necessary condition for improvement on target task. Throughout the whole, our strategies still can not compete against Inflated.

5 Conclusion and Future Works

In this study, we evaluate the effectiveness of self-supervised representation learning methods using temporal-sequence for 3DCNN. From results of experiments, we obtain the knowledge that low-level and easily available temporal feature from RP and SS improves optimization for 3DCNN and too difficult task to solve can not contribute to improvement.

We discuss the future research below:

Analyzing effect of motion feature: Although we show the effectiveness of SSL on 3DCNN in the form of improvement on accuracy, we should also explore how captured motion feature affects 3DCNN and action recognition on other experiments such as Huang et al.'s [5].

Removing ambiguity of PB: Despite intuition that PB setting can give semantic information to 3DCNN, this setting did not make an improvement. However, removing ambiguity of this task may lead to better result. For example, limiting dataset, action class or spatial region to ones including more intense motions.

Training on more large-scale DB: Since we set self-supervised manner, we can assume getting the benefit of more large-scale data without manual annotation. In previous work [4], training on immense number of *video* is significantly effective for 3DCNN. This is worth trying on SSL for novel knowledge of motion feature.

Fusing benefits of two features: Although we focus on motion feature, appearance feature obtained by Inflation is also strong. While we consider temporal SSL and Inflation separately now, there may be an effective method to fuse benefits of both initialization strategies.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733. IEEE (2017)
2. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5729–5738. IEEE (2017)

3. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
4. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 18–22 (2018)
5. Huang, D.A., et al.: What makes a video a video: analyzing temporal information in video understanding models and datasets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7366–7375 (2018)
6. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
8. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 667–676. IEEE (2017)
9. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 527–544. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_32
10. Monfort, M., et al.: Moments in time dataset: one million videos for event understanding. arXiv preprint arXiv:1801.03150 (2018)
11. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
12. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)