Deep Temporal Linear Encoding Networks

Hongzhi Liu

Jun 8, 2018

Abstract

Nowadays, human action recognition in videos has attracted quite some attention, due to the potential applications in video surveillance, behavior analysis and video retrieval. The CNN-encoding of features from entire videos for the representation of human actions has rarely been addressed. Instead, CNN work has focused on approaches to fuse spatial and temporal networks, but these were typically limited to processing shorter sequences. Today, I read a thesis written by Vivek Sharma, who is from University of Leuven. His team propose a new video representation, called temporal linear encoding (TLE) and embedded inside of CNNs as a new layer, which captures the appearance and motion throughout entire videos. It encodes this aggregated information into a robust video feature representation, via end-to-end learning. The experiments show that TLE outperforms current state-of-the-art methods on both datasets.

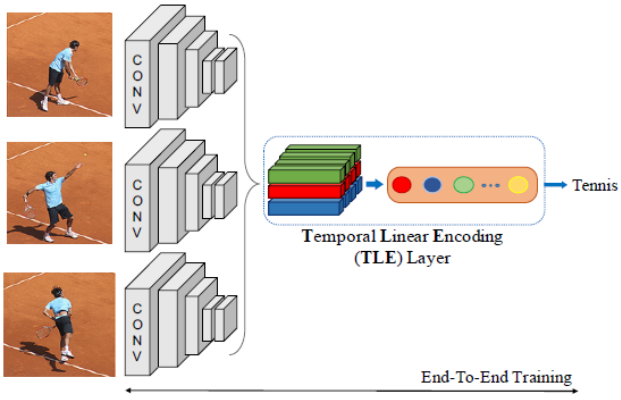

Figure 1. Temporal linear encoding for video classification. Given several segments of an entire video, be it either a number of frames or a number of clips, the model builds a compact video representation from the spatial and temporal cues they contain, through end-to-end learning. The ConvNets applied to different segments share the same weights.

1. Overview of Video Representation

Over the last two decades, several action recognition techniques in videos have been proposed by the vision community. The performance of computer vision systems still falls behind that of people even if considerable progress was made. On top of the challenges that make object class recognition hard, there are issues like camera motion and the continously changing viewpoints that come with it. Whereas Convolutional Networks have caused several sub-fields of vision to leap forward, they still lack the capacity to exploit long-range temporal information. Neural networks for action recognition can be categorized into two types, namely *one-stream* ConvNets [3] and *two-stream* ConvNets [5].

As to the *one-stream* ConvNets, spatial networks perform action recognition from individual video frames. They lack any form of motion modeling. Besides, temporal networks typically get their motion information from dense optical flow. This reliance on dense temporal sampling leads to excessive computational costs for longer videos. The *two-stream* ConvNets have shown to outperform *one-stream* ConvNets. They exploit fusion techniques like trajectory-constrained pooling and consensus pooling [8]. The fusion methods of spatial and motion information lie at the heart of the state-of-the-art *two-stream* ConvNets.

Motivated by the above observations, Sharma proposes the new spatio-temporal encoding [1] illustrated in Fig. 1. The design of the spatio-temporal deep feature encoding aims to aggregate multiple video segments over longer time ranges. To that end, the team use their temporal linear encoding (TLE), which is inspired by previous works on video representations [7] and feature encoding methods. TLE is a form of temporal aggregation of features sparsely sampled over the whole video using feature map aggregation techniques and then projected to a lower dimensional feature space using encoding methods powered by end-to-end learning of deep networks.

2. Deep Temporal Linear Encoding

In a video, the motion between consecutive frames tends to be small. This suggests that the team need a video rep-

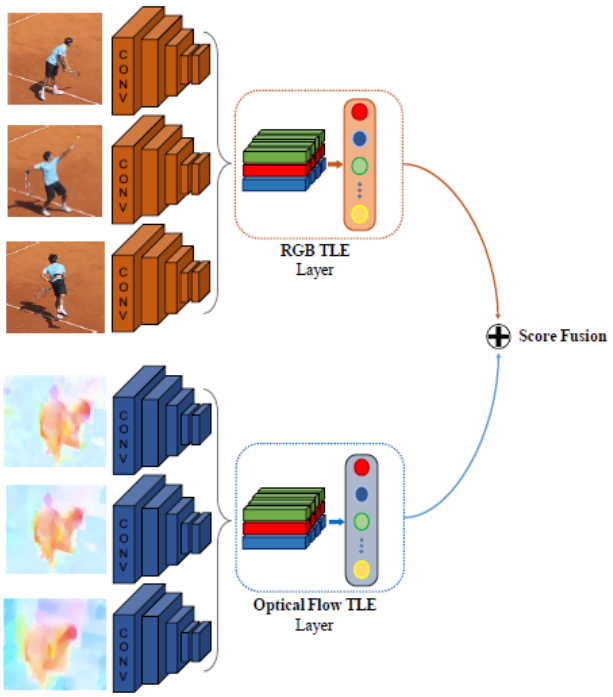

Figure 2. The temporal linear encoding applied to the design of two-stream ConvNets [5]: spatial and temporal networks. The spatial network operates on RGB frames and the temporal network operates on optical flow fields.

resentation that encodes all the frames together, in order to also capture long-range dynamics. Their goal is to create a single feature space in which to represent each video using all its selected frames or clips, rather than scoring separate frames with classifiers and label the video based on scores aggregation. They propose temporal linear encoding (TLE) to aggregate spatial and temporal information from an entire video and to encode it into a robust and compact representation using end-to-end learning as shown in Fig. 2.

Consider the output feature maps of CNNs truncated at a convolutional layer for K segments extracted from a video V. The feature maps are matrices $S1, S2, \ldots, S_K$ of size $S \in \mathbb{R}^{h \times w \times c}$, where h, w and c denote the height, width, and number of channels of the CNN feature maps. A temporal aggregation function T: $S1, S2, \ldots, S_K \rightarrow X$, aggregates K temporal feature maps to output an encoded feature map X.

A bilinear model computes the outer product of two feature maps as Equation 1:

$$y = W[X \otimes X']. \tag{1}$$

Where $X \in \mathbb{R}^{(hw) \times c}$, $X' \in \mathbb{R}^{(hw) \times c'}$ are input featuremaps, $y \in \mathbb{R}^{c \times c'}$ are bilinear features, $\otimes$ denotes the outer product, [ ] turns the matrix into a vector byconcatenating the columns, and W represents model parameters to be learned (here linear). In their case, $X = X'$. The resulting bilinear features capture theinteraction of features with each other at all spatial locations,hence leading to a high-dimensional representation.

Compared to the fully-connected pooling method, bilinear models projects the high dimensional feature space to a lower dimensional space, which is far fewer in parameters and still perform better than fully-connected layers in performance, apart from computational efficiency.

One can readily employ other encoding methods like deep fisher encoding or VLAD, instead of bilinear models or fully connected pooling. When bilinear models are used the features are passed through a signed squared root and L2-normalization. In either case, the team use softmax as a classifier.

Table 1. Accuracy (%) performance comparison of the aggregation functions in TLE BN-Inception network for two-stream ConvNets using 3 segments on UCF101 and HMDB51 datasets.

Aggregation Function	UCF101/HMDB51
Element-wise Maximum	91.3/67.4
Element-wise Average	92.6/68.1
Element-wise Multiplication	94.8/70.4

Table 2. Different architecture accuracy (%) performance comparison of spatial and temporal ConvNets using 3 segments on the UCF101 and HMDB51 datasets.

	UCF101/HMDB51	UCF101/HMDB51
Method	Spatial ConvNets	Temporal ConvNets
AlexNet	74.4/50.8	82.7/52.4
VGG-16	81.5/60.9	86.8/61.5
BN-Inception	86.9/63.2	89.1/66.4

3. Evaluation of TLE

The team explore not only different aggregation functions to linearly aggregate the segments into a compact intermediate representation for encoding but also different ConvNet architectures for both two-stream. For this evaluation, they report the accuracy of split1 on UCF101 and HMDB51. The reported performance is for TLE with bilinear models using the tensor sketch algorithm.

Sharma reports the performance of the different aggregation strategies in Table 1. He observes that the element-wise multiplication performs the best. Therefore, they choose element-wise multiplication as a default aggregation function. They believe combining the feature maps in this way allows them to aggregate the appearance and motion information accurately, hence leading to better results.

Furthermore, the team compare the different ConvNet architectures for TLE. Specifically, they compare AlexNet [4], VGG-16 [6], and BNInception [2]. Among all architectures shown in Table 2, BN-Inception achieves the best performance, better than the AlexNet and VGG-16 architectures. BN-Inception is 5.4/2.3% (spatial ConvNets) and 2.3/4.9% (temporal ConvNets) better than VGG-16 on UCF101/HMDB51. Therefore, they choose BN-Inception as a default ConvNet architecture for TLE.

References

[1] A. Diba, V. Sharma, and L. Van Gool. Deep temporal linear encoding networks. In *CVPR*, 2017. 1

[2] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[7] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1

[8] 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 *ECCV*, 2016. 1