Temporal Segment Networks: Towards Good Practices for Deep Action Recognition - Summary

Liming Wang, Yuanjun Xiong, Zhe Wang. Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool

Introduction

This paper starts from the idea that applying CNNs in video-based tasks such as action recognition comes with two major obstacles:

- "Long-range temporal structure plays an important role in understanding the dynamics in action videos", but the authors argue that mainstream ConvNets are focusing on appearance and short-term motions, thus lacking to model properly the long-range temporal structure.
- Training CNNs requires a large volume of data, which is difficult to obtain in video-based tasks, due to expensive data-collection and annotation, thus resulting in limited public datasets.

Starting from these two obstacles, this paper proposes a video-level framework built on top of the two-stream architecture, that aims to model the long-range temporal structure needed for action recognition tasks, and also to learn a convolutional network given limited training samples.
The proposed method is to extract short snippets over a long video sequence with a sparse sampling scheme, where the samples distribute uniformly along the temporal axis.

Temporal Segment Networks

As described previously, instead of working with single frames, TSN operates on a sequence of short snippets sparsely sampled from the entire video. After the snippets are extracted, each one of them will produce its own preliminary prediction, and a consensus among the snippets will be derived as the video-level prediction.
Specifically, given a video V, the video is divided into K segments *{S1,S2,…,Sk}* of equal durations. Then this sequence is modelled as follows:

$$\mathrm{TSN}(T_1, T_2, \cdots, T_K) = \mathcal{H}(\mathcal{G}(\mathcal{F}(T_1; \mathbf{W}), \mathcal{F}(T_2; \mathbf{W}), \cdots, \mathcal{F}(T_K; \mathbf{W}))).$$

Where *(T1,T2,…,Tk)* is the sequence of snippets. Each of these snippets is randomly sampled from its corresponding segment. *F(Tk; **W**)* represents the convolutional network, where W is the parameters matrix. The segmental consensus function *G* combines the outputs from multiple short snippets to obtain a consensus of class hypothesis among them. Based on this consensus, the prediction function (Softmax) predicts the probability of each action class for the whole video.

Loss Function

$$\mathcal{L}(y, \mathbf{G}) = -\sum_{i=1}^{C} y_i \left(G_i - \log \sum_{j=1}^{C} \exp G_j \right)$$

In the Loss Function formula, C represents the number of action classes and yi is the ground truth label for class I. The authors note that in their experiments, K set to 3 works best. The consensus function used is the averaging aggregation function. In the back propagation process, the gradients of the parameters with respect to the Loss value are derived as:

$$\frac{\partial \mathcal{L}(y, \mathbf{G})}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \mathbf{G}} \sum_{k=1}^{K} \frac{\partial \mathcal{G}}{\partial \mathcal{F}(T_k)} \frac{\partial \mathcal{F}(T_k)}{\partial \mathbf{W}},$$

Architecture: