



Hierarchical Video Frame Sequence Representation with Deep Convolutional Graph Network

Feng Mao^(✉), Xiang Wu, Hui Xue, and Rong Zhang

Alibaba Group, Hangzhou, China
{maofeng.mf,weiyi.wx}@alibaba-inc.com

Abstract. High accuracy video label prediction (classification) models are attributed to large scale data. These data could be frame feature sequences extracted by a pre-trained convolutional-neural-network, which promote the efficiency for creating models. Unsupervised solutions such as feature average pooling, as a simple label-independent parameter-free based method, has limited ability to represent the video. While the supervised methods, like RNN, can greatly improve the recognition accuracy. However, the video length is usually long, and there are hierarchical relationships between frames across events in the video, the performance of RNN based models are decreased. In this paper, we propose a novel video classification method based on a deep convolutional graph neural network (DCGN). The proposed method utilizes the characteristics of the hierarchical structure of the video, and performed multi-level feature extraction on the video frame sequence through the graph network, obtained a video representation reflecting the event semantics hierarchically. We test our model on YouTube-8M Large-Scale Video Understanding dataset, and the result outperforms RNN based benchmarks.

Keywords: Video classification · Sequence representation
Graph neural network · Deep convolutional neural network

1 Introduction

Nearly 80% of the data on the Internet are images and videos. Research on analyzing, understanding and mining these multimedia data has drawn a significant amount of attention from both academia and industry. Labeling or classifying videos is one of the most important requirements, and challenges remained to be solved. Google's YouTube-8M team introduced a large multi-label video classification dataset which composed of nearly 8 million videos-500K hours of video-annotated with a vocabulary of 4800 visual entities [1]. They used the state-of-the-art Inception-v3 network [2] pre-trained on ImageNet to extract frame features at one-frame-per-second, providing reliable data supporting for large-scale video understanding. The key task is modeling the long sequence of frame features. The popular methods are LSTM (Long Short-Term Memory Networks) [3], GRU (Gated recurrent units) [4], DBoF (Deep Bag of Frame Pooling) [1], etc.

In this work, a novel deep convolutional graph based frame feature sequence modeling method is proposed, which aims to mine the complex relationship between video frames and shots, and perform hierarchical semantic abstraction across video. Evaluations are made based on the YouTube8M-2018 dataset, which containing about 5 million videos and 3862 labels. We use the provided frame level Inception-v3 features to training the model, the results show that our model out-performs the RNN based methods.

2 Related Works

Video feature sequence classification is essentially the task of aggregating video features, that is, to aggregate N D -dimensional features into one D' -dimensional feature by mining statistical relationships between these N features. The aggregated D' -dimensional feature is a highly concentrated embedding, making the classifier easy to mapping the visual embedding space into the label semantic space. It is common using recurrent neural networks, such as LSTM (Long Short-Term Memory Networks) [3,5,6] and GRU (Gated recurrent units) [4,7], both are the state-of-the-art approaches for many sequence modeling tasks. However, the hidden state of RNN is dependent on previous steps, which prevent parallel computations. Moreover, LSTM or GRU use gate to solve RNN gradient vanish problem, but the sigmoid in the gates still cause gradient decay over layers in depth. It has been shown that LSTM has difficulties in converging when sequence length increase [8]. There also exist end-to-end trainable order-less aggregation methods, such as DBoF(Deep Bag of Frame Pooling) [1].

3 Proposed Method

In this paper, we propose a convolutional graph based video representation method for a sequence of video frame features. The main idea is that video is a hierarchical data structure, composed of events, scenes, shots, super-frames and frames. Additionally, the relations between frames, and the relations between shots are more complex than the order of a sequence. Consider an example sequence of “cooking show” as shown in Fig.1, frames containing same targets are not continuous, and distributed in various timestamps, as well as the events and shots. We model the video frame sequence, shot, and event hierarchically by a deep convolution graph Network (DCGN). It gradually abstracts information from frame level to video level by convolution and information propagating through graph, and finally generates a global representation for further classification. The method is shown in Fig. 2.

3.1 Graph Network with Deep Convolution

We use $\mathbf{F} = \{f_i^D, 0 \leq i \leq N\}$ to denote video frame feature sequence, where D is the dimension of the frame feature and N is the number of video frames.



Fig. 1. “cooking show” video frame sequence. Frames with same color border contain similar targets.

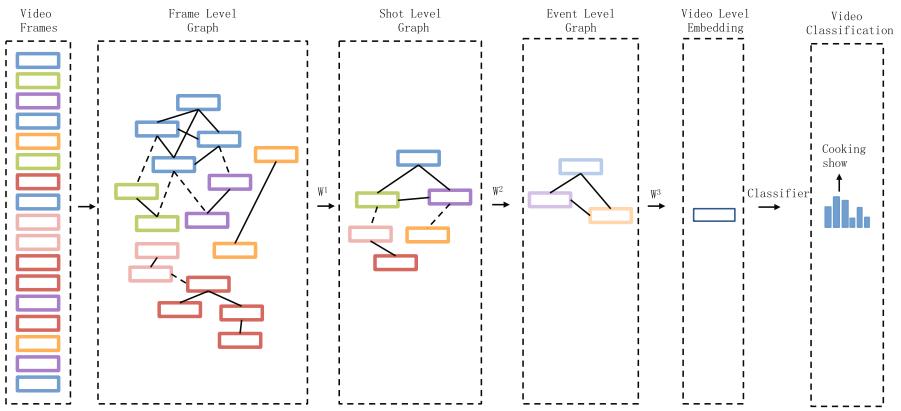


Fig. 2. High-level illustration of our proposed method DCGN. This is an example with label of “cooking show”, containing shots of chef cooking, host chatting, food, audience, etc. Rectangle with different color represent different shots or events. We use graph network to represent the relations between frames, shots, or events, that similar ones (nodes) have edge connected. The graph is gradually aggregated that represent frame, shot, event and video hierarchically.

We treat each frame, shot or event as a node of a graph, and the graph is densely connected (each pair of the nodes has an edge). Two nodes are connected weighted by their feature vector similarity. Let A be the adjacency matrix of F , and the elements is calculated by cosine similarity:

$$A(i, j) = \frac{\sum_{d=0}^{D-1} (f[i][d] \times f[j][d])}{\sqrt{\sum_{d=0}^{D-1} f[i][d]^2} \times \sqrt{\sum_{d=0}^{D-1} f[j][d]^2}} \quad (1)$$

Graph neural network uses a differentiable aggregation function 2 to perform “message passing”. It is an end-to-end learning model, which can learn node and edge representations simultaneously.

$$h^l = G(A, h^{l-1}, W^l) \quad (2)$$

where h^l is the node representation (messages) after l -th iteration, W^l is the parameters for the l -th iteration. G is the message propagation function. GCN [9] is a popular graph model whose G is:

$$G = \sigma(\sqrt{\bar{D}}A\sqrt{\bar{D}}h^{l-1}W^l) \quad (3)$$

where \bar{D} is the diagonal node degree matrix that normalizing A such that all rows sum to one.

Graph Pooling. In the Eq. 3 the adjacency matrix A is unchanged during iteration, so the topology of the graph is static. However, video frame sequence is a hierarchical structure, hence the graph topology should be abstracted to higher level gradually. We use two pooling methods to aggregate graph.

Average Pooling. This method applies the center of K consecutive nodes as the node of the next level graph:

$$p^l[i][d] = \frac{\sum_{k=0}^{K-1} h^{l-1}[i \times K + k][d]}{K}, d \in [0, D] \quad (4)$$

where K is the pooling kernel size, p^l is the l -th pooled graph node feature vector. After l -th iteration, the graph size is $\frac{1}{K^l}$ of the original.

Self-attention Based Pooling. This method performs a local self-attention to obtain a weight α for each feature of the local consecutive sequence, thereby obtaining a locally weighted and fused output of the feature sequence. Comparing to average pooling, it can better obtain the topology of the next layer graph, which is beneficial to the propagation of feature information. We formulate it as:

$$\begin{aligned} p_{att}^l[i] &= \alpha^l[i] \otimes h^{l-1}[i \times K : (i+1) \times K], \\ \alpha^l[i] &= \text{softmax}(h^{l-1}[i \times K : (i+1) \times K]W_{att}^l + b^l) \end{aligned} \quad (5)$$

where K is the number of local features to perform self-attention, and W_{att} and b are the parameters to learn.

Nodes Convolution. Different from GCN [9] which use fully connected layers to represent the nodes, in order to represent frame sequence hierarchically and maintain the local sequence order, we represent the nodes by convolution as follows:

$$c^l[i] = h^{l-1}[i \times K : (i+1) \times K]W^l \quad (6)$$

where c^l is the l -th graph node embedding, and W^l is the convolution kernel weights with size of $K \times D$.

Pooling and convolution are shown in Fig. 3.

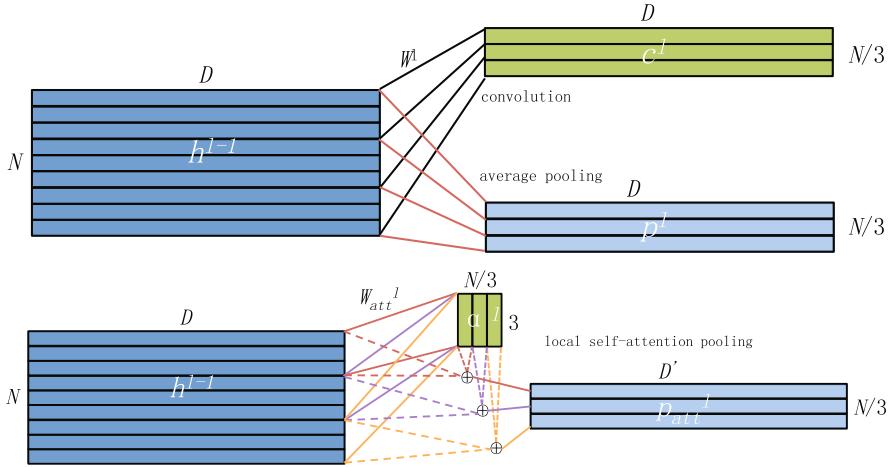


Fig. 3. Convolution and pooling ($K=3$) of N nodes with D -dimension, outputs new $\frac{N}{3}$ nodes with D -dimension.

Node Feature Propagation. Now, we have obtained a new graph topology and new feature vector for each node. In order to obtain a more complete representation in higher level, we perform the “message passing” across the entire graph, so that the fused feature of each node is generated from the global perspective. We use the similar form as Eq. 3:

$$h^l = \sigma(\sqrt{D^{l-1}} A^{l-1} \sqrt{D^{l-1}} c^{l-1} W^l) \quad (7)$$

where A^{l-1} is calculated using Eq. 1 in which f is replaced with p^{l-1} . Figure 4 shows the network described above.

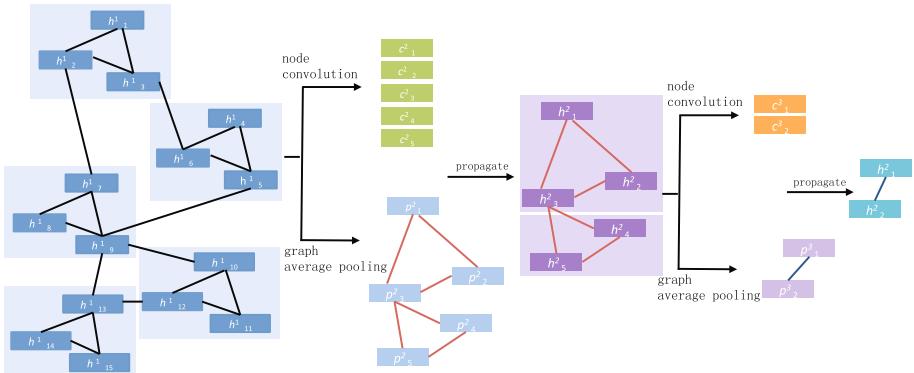


Fig. 4. Illustration of two layer convolutional graph network. 15 input feature vectors are aggregated to 2 output feature vectors.

3.2 Shot Segmentation Aided Graph Pooling

Shot is a basic temporal unit, which is a series of interrelated consecutive pictures taken contiguously by a single camera and representing a continuous action in time and space. It is independent over video category, and can be obtained by unsupervised nonparametric way. Inspired by the kernel temporal segmentation (KTS) algorithm [10], the proposed method applies the deep convolutional neural network features to calculate the matrix of frame-to-frame similarities. The algorithm uses dynamic programming to minimize within segment kernel variances, and get the best shot boundaries under given number of shots, the object function is:

$$\min_{m; t_0, \dots, t_{m-1}} J_{m,n} := L_{m,n} + Cg(m, n) \quad (8)$$

where

$$L_{m,n} = \sum_{i=0}^m v_{t_{i-1}, t_i}, g(m, n) = m(\log(\frac{n}{m}) + 1),$$

$$v_{t_{i-1}, t_i} = \sum_{t=t_i}^{t_{i+1}-1} \|f_t - \mu_i\|^2, u_i = \frac{\sum_{t=t_i}^{t_{i+1}-1} f_t}{t_{i+1} - t_i} \quad (9)$$

where m is the number of shots and $g(m, n)$ is a penalty term. Figure 5 shows the segmented shots together with their positions in frame-to-frame similarity matrix.

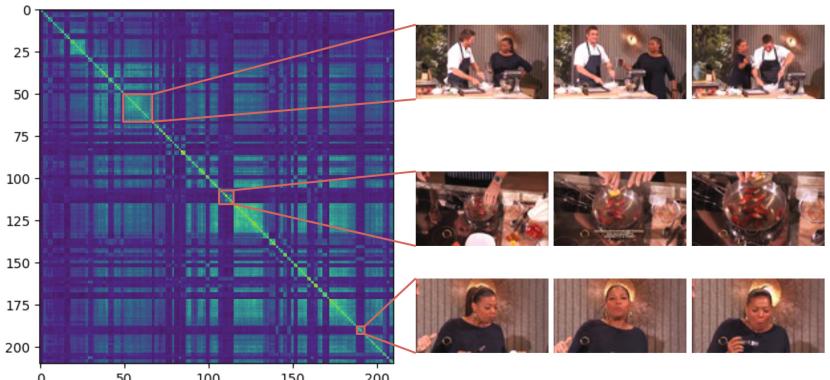


Fig. 5. The left is a matrix of frame-to-frame similarities, which brighter (larger value) points indicate more similar frames. Local bright squares are treated as a shot.

We apply this shot segmentation algorithm to the frame level graph modeling, and the formulation becomes:

$$p^1[t][d] = \frac{\sum_{k=s_t}^{s_{t+1}} f[k][d]}{s_{t+1} - s_t}, d \in [0, D] \quad (10)$$

$$c^1[t] = f[s_t : s_{t+1}]W^1 \quad (11)$$

where t is the shot number, s_t is frame index of the t -th shot boundary.

Figure 6 shows the network aided with shot segmentation: first, a fixed number m of shots are segmented, and a frame level convolution graph network is applied to get the shot level features H^1 . Then deep convolution graph network is applied to modeling higher-level features. Finally, the last level feature vectors are concatenated and inputted to a mixture of experts (MoE) [11] model to get the classification score. The model is trained end-to-end.

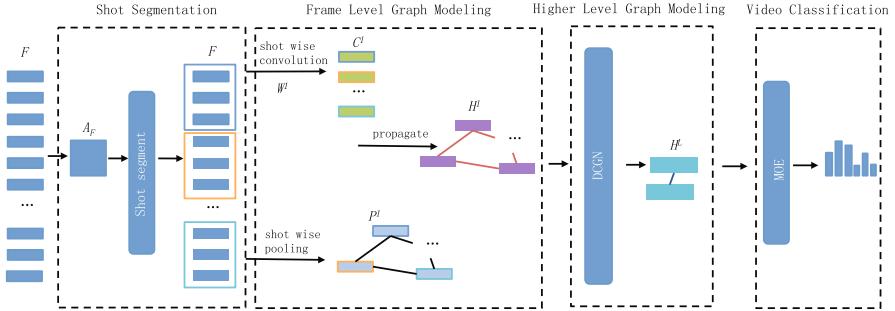


Fig. 6. Overall architecture. Shot-wise convolution and pooling performs at the frame level. The subsequent levels perform DCGN and obtain a video level feature H^L . MoE classifier use H^L as the input to provide the final label predictions.

4 Experiment

In this section, we evaluate the proposed method with other alternatives, including average pooling, LSTM, GRU, DBoF. We train these models on YouTube-8m-2018 frame-level training set and evaluate on validation set. There are 3.9 million videos in the training set and 1.1 million videos in the validation set. The dataset is a multi-class task with 3862 classes in total, and we use cross entropy loss to train the model:

$$Loss = - \sum_{c=1}^C y_c \log(p_c) \quad (12)$$

where C is the number of classes, p_c is the prediction for class c .

We use GAP and Hit@1 [1, 12] as the evaluation metrics:

$$GAP = \sum_{i=1}^N p(i) \Delta r(i) \quad (13)$$

where N is the number of final predictions, we set $N = 20$ here, $p(i)$ is the precision, and $r(i)$ is the recall.

$$Hit@1 = \frac{1}{|V|} \sum_{v \in V} \vee_{e \in G_v} \mathbb{I}(rank_{v,e} \leq 1) \quad (14)$$

where G_v is the set of ground-truth entities for v . This is the fraction of test samples that contain at least one of the ground truth labels in the best prediction.

The network architecture parameters and hyper-parameters for training these models are given in Table 1.

Table 1. Training settings

	Ours	Average Pooling	LSTM	GRU	DBoF
network parameters	layers: 5, filter size: 1024 (same for each layer)		cells: 1024, layers: 2	cells: 1024, layers: 2	cluster size: 8192, hidden size: 1024, pooling method: max
base learning rate			0.001		
learning rate decay			0.8		
learning rate decay examples			4M		
mini batch size			1024		
number of epochs			5		
number of mixtures in MoE			2		

Table 2 shows results for all methods. Simple averaging of features across all frames, perform poorly on this dataset. The order-less aggregation methods DBoF perform worse than the RNN based models. Our model outperforms all other models, and the self-attention based graph pooling got the best result.

Table 2. Evaluation results. All values in this table are averaged results and reported in percentage(%). Top-2 of each metrics are bolded.

Methods	GAP	Hit@1	Loss
Average pooling	76.5	83.5	5.49
LSTM	83.6	87.2	4.12
GRU	83.9	87.9	4.03
DBoF	81.1	86.2	6.02
ours + average graph pooling	84.1	87.4	4.01
ours + self-attention graph pooling	84.5	87.7	3.98

5 Conclusions

In this work, we propose a novel sequence representation method DCGN for video classification. One layer in DCGN is composed of graph pooling, nodes convolution and nodes feature propagation. The problem of representing complex relationships between frames or shots is addressed by applying this graph based network hierarchically across the frame sequence. Based on the quantitative results in Sect. 4, the proposed method DCGN outperform the other alternatives such as LSTM and GRU.

References

1. Abu-El-Haija, S., et al.: Youtube-8m: a large-scale video classification benchmark. CoRR abs/1609.08675 (2016)
2. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR abs/1512.00567 (2015)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(9), 1735–1780 (1997)
4. Cho, K., van Merriënboer, B., Gülcehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014)
5. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMs. In: International Conference on Machine Learning, pp. 843–852 (2015)
6. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4694–4702 (2015)
7. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. arXiv preprint [arXiv:1511.06432](https://arxiv.org/abs/1511.06432) (2015)
8. Li, S., Li, W., Cook, C., Zhu, C., Gao, Y.: Independently recurrent neural network (IndRNN): building a longer and deeper RNN. CoRR abs/1803.04831 (2018)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907 (2016)
10. Potapov, D., Douze, M., Harchaoui, Z., Schmid, C.: Category-specific video summarization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 540–555. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_35
11. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Comput. **6**(2), 181–214 (1994)
12. Google: Youtube-8m starter code (2018). <https://github.com/google/youtube-8m>