HIERARCHICAL SEQUENCE REPRESENTATION WITH GRAPH NETWORK

Da Chen[1*], *Xiang Wu*[1*], *Jianfeng Dong*[2,3,†], *Yuan He*[1], *Hui Xue*[1], *Feng Mao*[1]

[1]Alibaba Group
[2]Alibaba-Zhejiang University Joint Institute of Frontier Technologies
[3]Zhejiang Gongshang University

ABSTRACT

Video classification problem is a challenging task in computer vision. The performance of this task is highly relied on the scale of training data and the effectiveness of video embedding via a robust embedding network. Unsupervised solutions such as feature average pooling technique, as a simple label-independent and parameter-free based method, cannot efficiently represent the video sequences. While supervised methods, such as RNN, can improve the recognition accuracy. The performance of RNN based methods, however, is decreased with the increasing length of the videos and the hierarchical relationships between frames across events in the video. 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 embedding feature extraction on the video frame sequence through the graph network, and obtained a video representation which reflects the event semantics hierarchically. Experiments on YouTube-8M Large-Scale Video Understanding dataset show that our proposed model outperforms the commonly used RNN based models, verifying its effectiveness for video classification.

Index Terms— Video Classification, Sequence Representation, Graph Neural Network, Deep Convolutional Neural Network

1. INTRODUCTION

With the significant increment of the number of videos posted everyday, understanding video content becomes a challenging task in computer vision. Research on analyzing, understanding and recognizing these multimedia data has drawn attention from both academia and industry. In this paper, we address the problem of video classification on a large scale video dataset where the videos are from the Internet. It is challenging as it contains a huge amount of videos which have great variability even for videos in the same class. The content and the quality is also inconsistent due to the biased manual label and the source quality. Hence classical video classification methods such as LSTM (Long Short-Term Memory Networks) [2], GRU (Gated recurrent units) [3], DBoF (Deep Bag of Frame Pooling) [1], etc. are not able to handle video classification task under this complex dataset.

Fig. 1. Example in YouTube-8M dataset [1]: "Cooking show" video frames crop from every 10 seconds. Under the same label, the frames can be different from each other. Frames with same color border contain similar objects or scenes and indicate one of the difficulties of this dataset: complex scenes distributed temporally.

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 the 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 the graph, and finally generates a global representation for further classification.

Evaluations are made based on the YouTube8M-2018 dataset, which contains about 5 million videos and 3862 labels. We use the provided frame level Inception-v3 features to training the model. The quantitative results show that our model outperforms the baselines including RNN based method, GRU based method, *etc*.

∗ Equal Contribution.
† Corresponding Author.

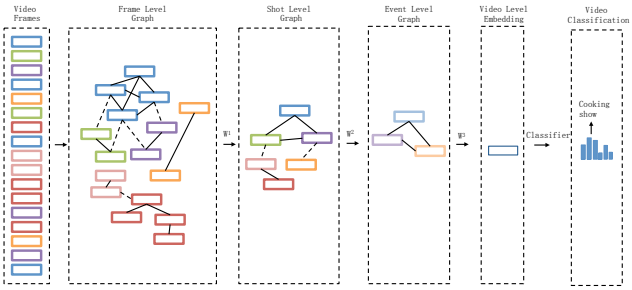

Fig. 2. A high-level illustration of our proposed method DCGN. This is an example with the label of "cooking show", containing shots of chef cooking, host chatting, food, audience, etc. Rectangles with different colour indicate different shots or events. We use graph network to represent the relations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

2. RELATED WORK

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) [2] [4] [5] and GRU (Gated recurrent units) [3] [6] [7], both are the state-of-the-art approaches for many sequence modelling tasks. However, the hidden state of RNN is dependent on previous steps, which prevent parallel computations. Moreover, LSTM or GRU uses gates to solve the RNN gradient vanish problem, but the sigmoid in the gates still causes 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], which has also been widely used in video representation [9].

3. METHOD

In this section, the proposed method is detailed presented. The overall architecture of the proposed method is shown in Fig. 2 with the label of "cooking show" in Fig. 1.

3.1. Graph Network with Deep Convolution

For the ease of reference, we use F=$\{f_1, f_2, ..., f_N\}$ to denote a video consists of N video frames, where $f_t \in \mathbb{R}^D$ indicates the D-dimensional deep feature vector of the t-th frame. We treat each frame, shot, and event as a node of a graph, and the graph is densely connected (each pair of nodes has an edge). The connections between the two nodes are weighted by the similarity in terms of their corresponding feature vectors. Let A be the adjacency matrix of F, where elements are calculated by cosine similarity:

$$A(i,j) = \frac{f_i \cdot f_j}{||f_i|| \cdot ||f_j||} \tag{1}$$

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

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

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 as a graph convolution network(GCN) [10] and can be presented as:

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

where $\overline{D}$ is the diagonal node degree matrix that normalizing the adjacency matrix A such that all rows can sum to be one.

3.1.1. Graph pooling

In Eq. 3 the adjacency matrix A is unchanged during iteration, so the topology of the graph is static. However, the video frame sequence is in a hierarchical structure, hence the graph topology should be abstracted to a higher level gradually. Hence, we use two pooling methods to aggregate graph, *i.e.*, average pooling and self-attention based pooling.

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

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

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. Compared with the 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:

$$\begin{aligned} p^l_{att}[i] &= \alpha^l[i] \otimes h^{l-1}[i \times K : (i+1) \times K], \\ \alpha^l[i] &= softmax(h^{l-1}[i \times K : (i+1) \times K]W^l_{att} + b^l) \end{aligned} \tag{5}$$

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

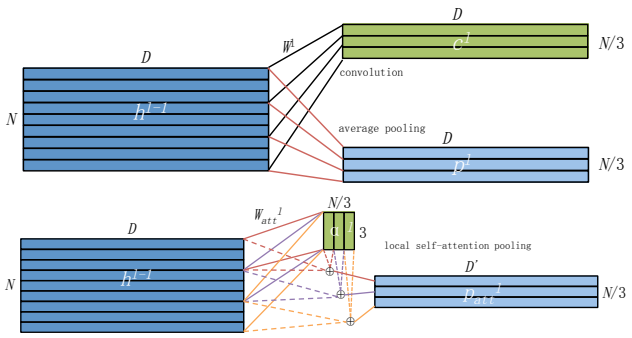

Fig. 3. Convolution and pooling (K=3) of N nodes with D-dimension, outputs new $\frac{N}{3}$ nodes with D-dimension.

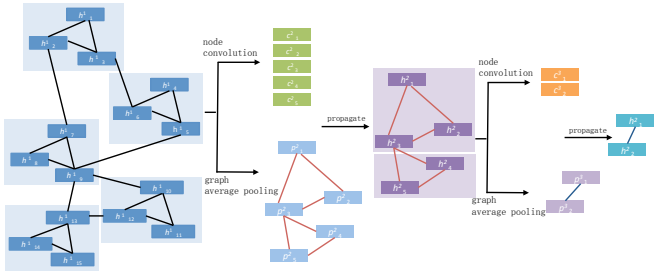

Fig. 4. Illustration of a two-layer convolutional graph network. 15 input feature vectors are aggregated to 2 output feature vectors.

3.1.2. Nodes convolution.

Different from GCN [10] which use fully connected layers to represent the nodes, to represent frame sequence hierarchically and maintain the local sequence order, we represent the nodes by convolution as follows:

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

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

Details of the pooling and convolution process are shown in Fig. 3.

3.1.3. Node feature propagation.

Based on previous steps, we have obtained a new graph topology and new feature vector for each node. 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. Similar form as Eq. 3 is applied:

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

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

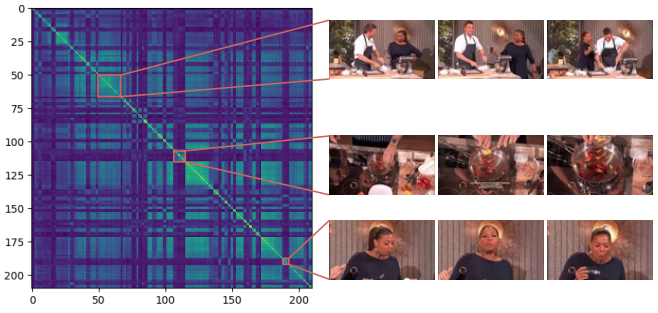

Fig. 5. The left is a matrix of frame-to-frame similarities, which brighter(larger value) points indicate more similar frames. Local bright squares are treated as a shot.

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 the video category and can be obtained by the unsupervised nonparametric way. Inspired by the kernel temporal segmentation (KTS) algorithm [11], 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 a given number of shots, the objective function is:

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

where

$$\begin{aligned} L_{m,n} &= \sum_{i=0}^{m} v_{t_{i-1},t_i}, g(m,n) = m(log(\frac{n}{m}) + 1), \\ v_{t_{i-1},t_i} &= \sum_{t=t_i}^{t_{i=1}-1} \|f_t - \mu_i\|^2, u_i = \frac{\sum_{t=t_i}^{t_{i+1}-1} f_t}{t_{i+1} - t_i} \end{aligned} \tag{9}$$

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

We apply this shot segmentation algorithm to the frame-level graph modelling, and the pooling and convolution process becomes:

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

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

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

Fig. 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

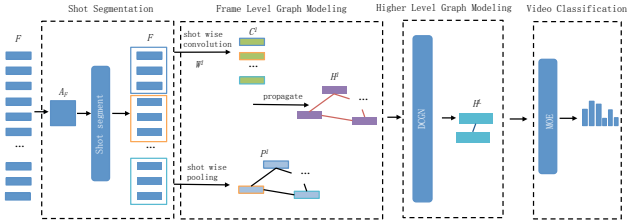

Fig. 6. Overall architecture. Shot-wise convolution and pooling are performed at the frame level, and followed by DCGN to obtain a video level feature H^L. MoE classifier is applied over H^L to provide the final label predictions.

level features H^1. Then deep convolution graph network is applied to model the higher-level features. Finally, the last level feature vectors are concatenated and inputted to a mixture of experts (MoE) [12] model to get the classification score. The model is trained in an end-to-end manner with a cross-entropy:

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

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

4. EXPERIMENTAL RESULTS

In this section, we first introduce the dataset YouTube-8M and then evaluate the proposed method with other alternatives, including average pooling, LSTM, GRU, DBoF based methods.

4.1. Dataset

YouTube-8M is a large multi-label video classification dataset released by Google [1]. It is composed of nearly 8 million videos-500K hours of video-annotated with a vocabulary of 4800 visual entities. The state-of-the-art Inception-v3 network [13] pre-trained on ImageNet is applied to extract frame features at one-frame-per-second frame rate. This feature extraction provides reliable data supporting large-scale video understanding. One of the key challenges is to model the long sequence of frame features efficiently.

On YouTube-8M, there are 3.9 million videos in the training set and 1.1 million videos in the validation set. The dataset has 3862 classes in total. We train these models on the training set and evaluate on the validation set.

Performance metric. We use GAP and Hit@1 [1] as the evaluation metrics:

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

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

Table 1. Performance comparison on YouTube-8M. All scores are reported in percentage(%). Top 2 scores for each metrics are in bold.

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 pooling	**84.1**	87.4	**4.01**
ours + self-attention pooling	**84.5**	**87.7**	**3.98**

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

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.

4.2. Implementation details

In this paper, the proposed method and other baselines are trained for comparison. The proposed method has 5 layers with 1024 filters in each of them. LSTM and GRU based methods have two layers with 1024 cells in each of them. DBoF has cluster size 8192 and hidden size 1024 trained with max pooling. All methods are trained under learning rate 0.001 with learning rate decay 0.8 after each $4M$ examples. The batch size is 1024. Results are obtained after 5 epochs and 2 mixtures in MoE.

4.3. Quantitative Results

Table 1 summaries the performance on the YouTube-8M dataset. Simple averaging on frame features across performs worse. 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. The results verify the effectiveness of our proposed model.

5. CONCLUSION

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 Section 4, the proposed method DCGN outperforms the other alternatives such as LSTM and GRU.

6. ACKNOWLEDGEMENT

This work was supported by Alibaba Group, National Key R&D Program of China (No. 2018YFB1404102), NSFC (No. 61902347) and ZJNSF (No. LQ19F020002).

7. REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[2] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[4] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.

[5] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[6] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[7] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang, "Dual encoding for zero-example video retrieval.," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9346–9355.

[8] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5457–5466.

[9] Zhaoyu Zhang, Xiang Wu, Jianfeng Dong, Yuan He, Hui Xue, and Feng Mao, "Noise learning for weakly supervised segment classification in video," in *Proceedings of the IEEE International Conference on Computer Vision Workshop*, 2019.

[10] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[11] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid, "Category-specific video summarization," in *European conference on computer vision*. Springer, 2014, pp. 540–555.

[12] Michael I Jordan and Robert A Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.