VideoGraph: Recognizing Minutes-Long Human Activities in Videos

1. Idea

- 一个复杂的动作是由多个基本的单元动作组成
- 虽然同一动作有很大的不同，但存在一个潜在的整体的时间结构
- 通过建立图结构来学习这种潜在的关系

2. Method

2.1. LSTM和3D CNN缺点：

- 活动时间太长
- 同一种动作有着很大的不同：例如：煮咖啡，有多个路径可以得到最终的咖啡

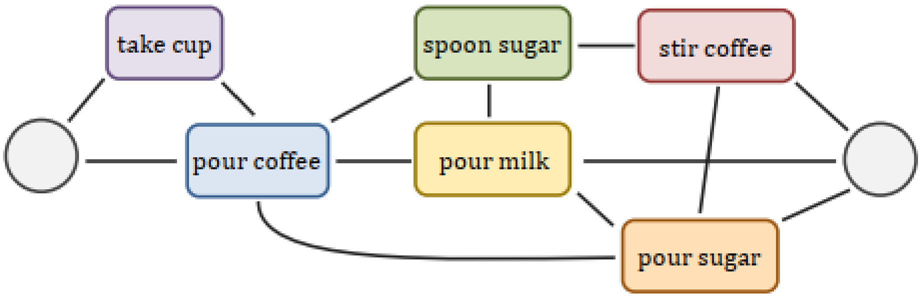


Figure 1: The activity of "preparing coffee" can be represented as undirected graph of unit-actions. We are inspired by graphs to represent this activity. The reason is that a graph can portray the many ways one can carry out such activity. More over, it preserves the temporal structure of the unit-actions. Reproduced from [1].

3. VideoGraph

> - 虽然有很大的不同,但是**整体的时间结构还是有的.**
> - 采用基于图的表示方法:保留时间结构,**可以处理更长时间的动作**.

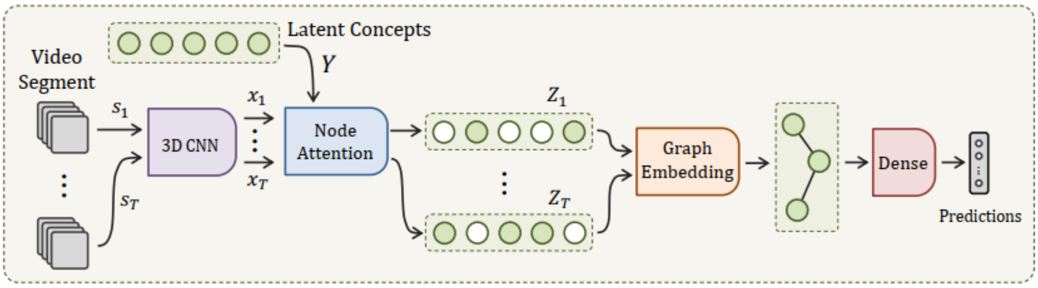


Figure 2: Overview of VideoGraph. It takes as input a video segment s_i of 8 frames from an activity video v. Then, it represents it using standard 3D CNN, .e.g I3D. The corresponding feature representation is x_i. Then, a node attention block attends to a set of N latent concepts based on their similarities with x_i, which results in the node-attenative representation Z_i. A novel graph embedding layer then processes Z_i to learn the relationships between its latent concepts, and arrives at the final video-level representation. Finally, an MLP is used for classification.

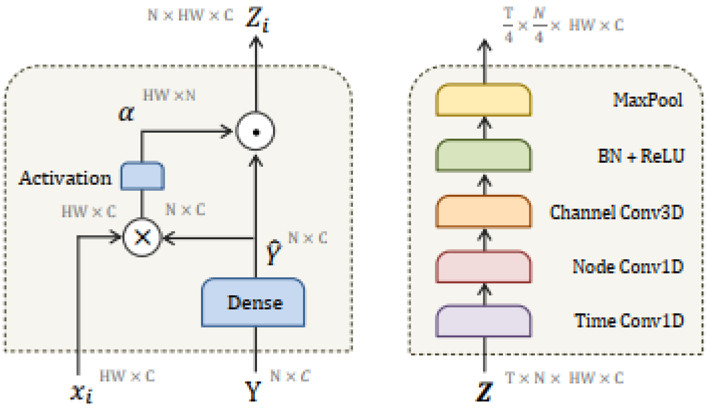
> 采帧方法：随机采T段，每段选择连续的8帧

3.1. 目标：构建人类活动的无向图$G=(N,E)$

- N：表示活动中的关键的单个动作（unit-actions）
- edges：简单动作之间的时序关系

3.2. 学习图结点nodes

3.2.1. Node Attention Block



(a) Node Attention Block

(b) Graph Embedding Layer

Figure 3: (a) Node attention block measures similarities α between segment feature x_i and learned nodes $\hat{Y}$. Then, it attends to each node in $\hat{Y}$ using α. The result is the node-attentive feature Z_i expressing how similar each node to x_i. (b) Graph Embedding layer models a set of T successive node-attentive features $\mathbf{Z}$ using 3 types of convolutions. *i.* Timewise Conv1D learns the temporal transition between node-attentive features $\{Z_i, ..., Z_{i+t}\}$. *ii.* Nodewise Conv1D learns the relationships between nodes $\{z_{i,j}, ..., z_{i,j+n}\}$. *iii.* Channelwise Conv3D updates the representation for each node z_{ij}.

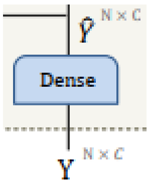
3.2.2. 公式表述

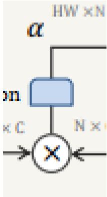
$$\begin{aligned} \hat{Y} &= w * Y + b && (1) \\ \alpha &= \sigma(x_i * \hat{Y}^T) && (2) \\ Z_i &= \alpha \odot \hat{Y} \\ &= \alpha_j \odot y_j,\quad j = 1, 2, ..., N && (3) \end{aligned}$$

3.2.3. 内部组件解释

复杂动作由多个单元简单动作组成，将简单的单元动作作为节点nodes

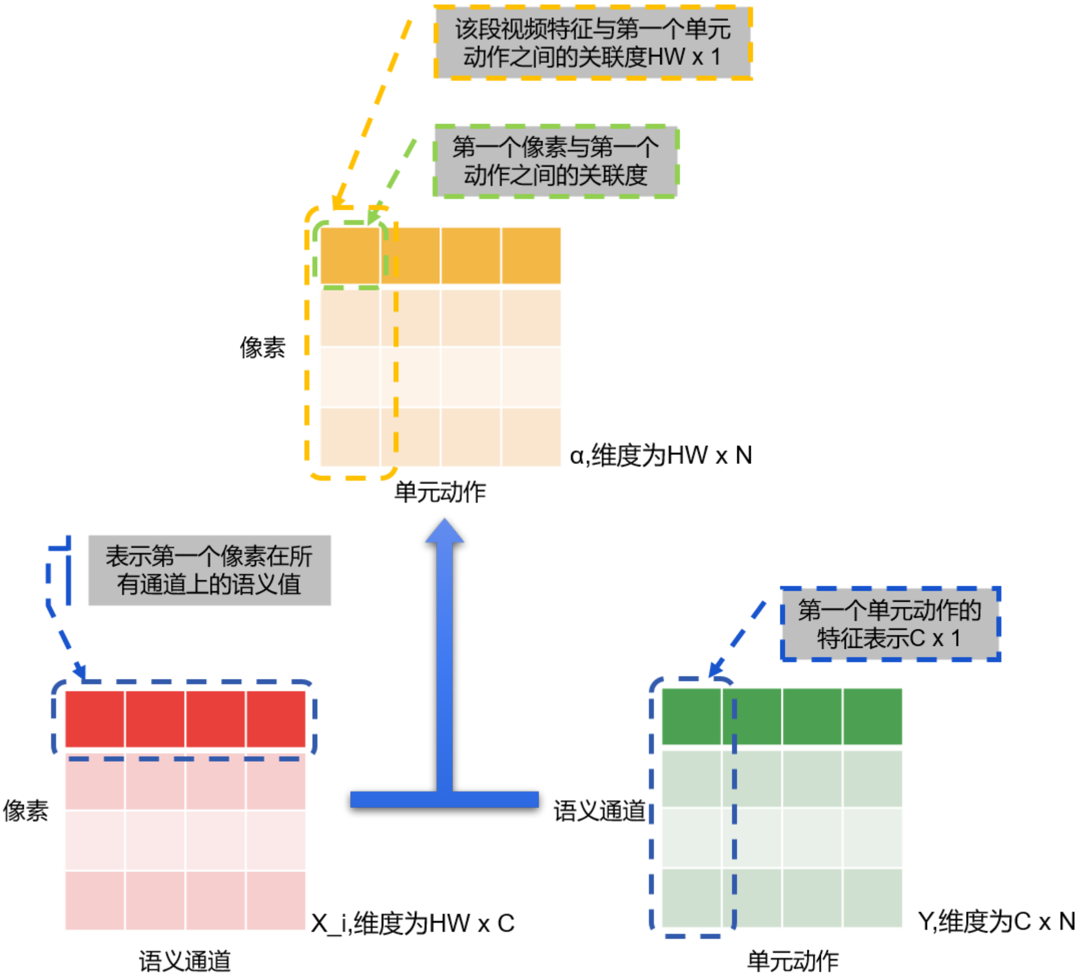
如何将产生的特征x_i与节点 `Y` 相关联？

- `node attention block` 来进行关联：
    - 得到x_1: $s_1->x_1(8\times H\times W\times C->1\times 7\times 7\times 1024)$,$s_i$是第i段视频
    - $Y\in\mathbb{R}^{N\times C}$是一组潜在特征，也是N个节点
- 

MLP操作：增加可学习性
- 

$dotproduct$操作：使用\softmax()增加非线性，用来计算每个视频段特征x_i与单元动作Y之间的关联度。

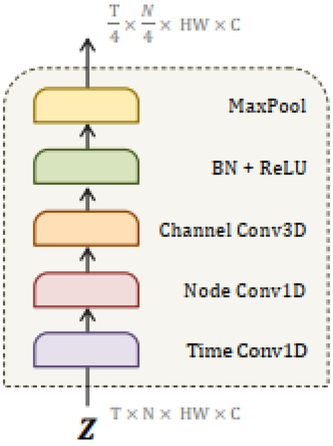
  权重计算意义的解释：


该段视频特征与第一个单元动作之间的关联度HW x 1
第一个像素与第一个动作之间的关联度
像素
α,维度为HW x N
单元动作
表示第一个像素在所有通道上的语义值
第一个单元动作的特征表示C x 1
像素
语义通道
X_i,维度为HW x C
语义通道
Y,维度为C x N
单元动作

> - 一个小问题：Y是如何产生的？
>
>   Y是随机产生。`centroids = np.random.rand(n, dim)`

3.3. Learning The Graph Edges 学习图的边

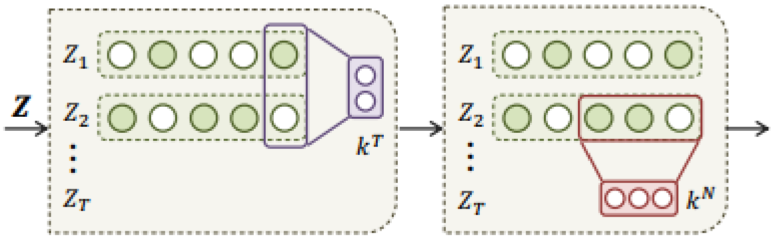
3.3.1. Graph Embedding Layer


$\frac{T}{4} \times \frac{N}{4} \times HW \times C$
MaxPool
BN + ReLU
Channel Conv3D
Node Conv1D
Time Conv1D
Z
$T \times N \times HW \times C$

(b) Graph Embedding Layer

使用图嵌入层来学习两个信息：

- `Timewise Conv1D:` 单元动作之间的时间上的迁移信息
- `Nodewise Conv1D:` 节点之间的关联性

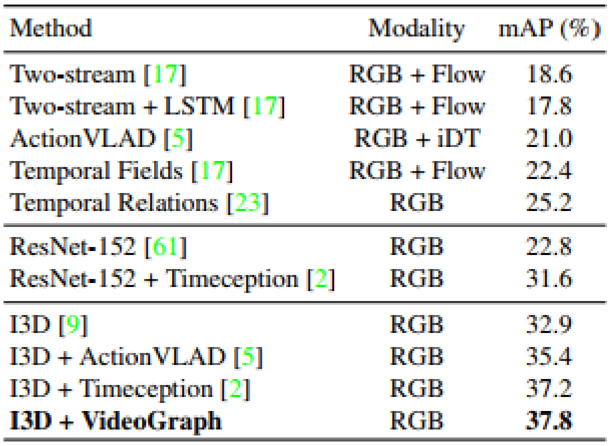


(a) Timewise Conv1D (b) Nodewise Conv1D

Figure 4: (a) Timewise Conv1D learns the temporal transition between successive nodes-embeddings $\{Z_i, ..., Z_{i+t}\}$ using kernel k^T of kernel size t. (b) Nodewise Conv1D learns the relationships between consecutive nodes $\{z_{i,j}, ..., z_{i,j+n}\}$ using kernel k^N of kernel size n.

4. 实验：

4.1. 在charades数据集上的性能



Method	Modality	mAP (%)
Two-stream [17]	RGB + Flow	18.6
Two-stream + LSTM [17]	RGB + Flow	17.8
ActionVLAD [5]	RGB + iDT	21.0
Temporal Fields [17]	RGB + Flow	22.4
Temporal Relations [23]	RGB	25.2
ResNet-152 [61]	RGB	22.8
ResNet-152 + Timeception [2]	RGB	31.6
I3D [9]	RGB	32.9
I3D + ActionVLAD [5]	RGB	35.4
I3D + Timeception [2]	RGB	37.2
I3D + VideoGraph	RGB	**37.8**

Table 1: VideoGraph outperforms related works using the same backbone CNN. Results are for Charades dataset.

4.2. VideoGraph节点的学习过程可视化

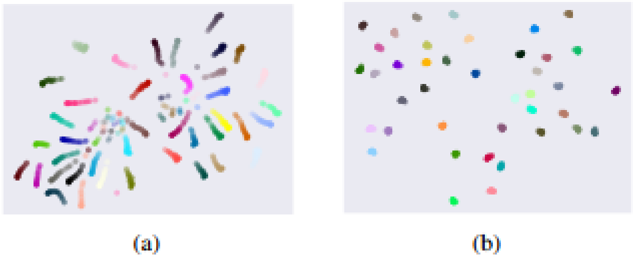
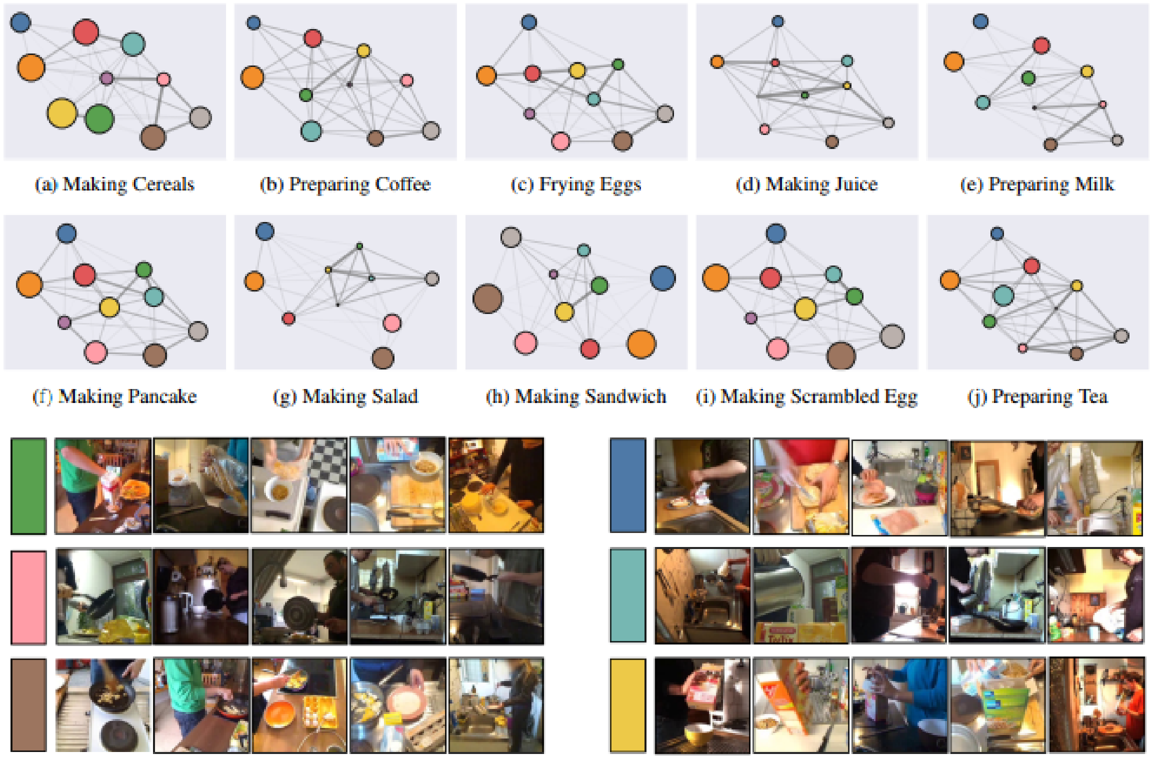


Figure 5: Visualization of the learned graph nodes. In the first 20 epoch during training (left), VideoGraph updates the node features $\hat{Y}$ to increase the pairwise distance between them. That is, VideoGraph learns discriminant representations of the nodes. In the last 20 epoch during training (right), the learning cools down and barely their representation is updated. We visualize using t-SNE [63].

4.3. 分类实例可视化：



(a) Making Cereals (b) Preparing Coffee (c) Frying Eggs (d) Making Juice (e) Preparing Milk

(f) Making Pancake (g) Making Salad (h) Making Sandwich (i) Making Scrambled Egg (j) Preparing Tea

(k) Top related images to the nodes. These nodes are related to: ● cereal, ● pan, ● eggs, ● sandwich, ● kettle, and ● foodbox.

Figure 7: We visualize the relationship discovered by the first layer of graph embedding. Each sub-figure is related to one of the 10 activities in Breafast dataset. In each graph, the nodes represent the latent concepts learned by graph-attention block. Node size reflects how dominant the concept, while graph edges emphasize the relationship between these nodes.