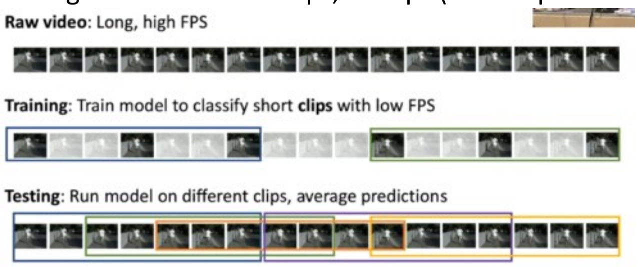
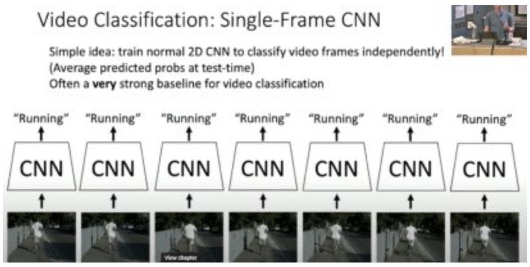
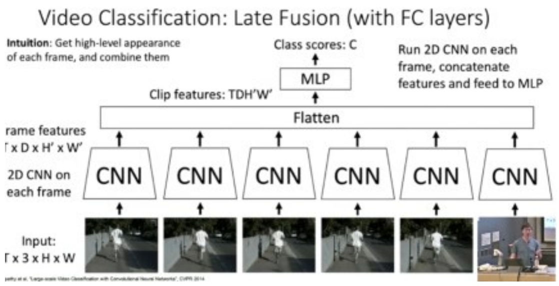
7 Video l18

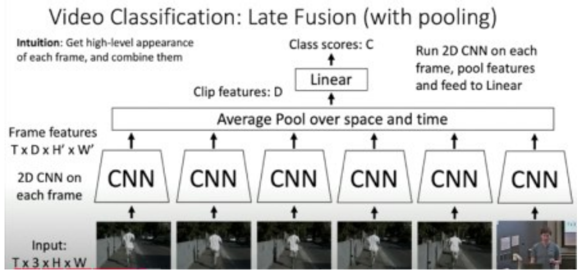
2024年12月18日　　21:53

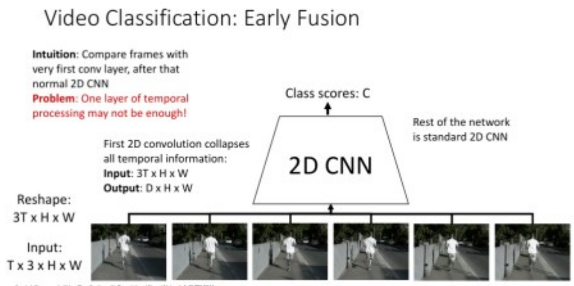
1. video: a sequence of images, 4D tensor
    1. recognize actions
2. problem: big-> train on short clips, low fps (frames per second) and low spatial resolution

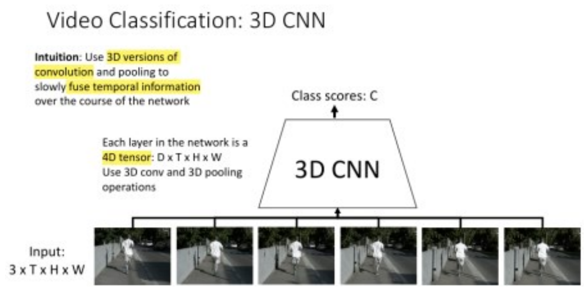


3. basic models
    1. 

       Very important, always try this first. 之后的所有模型都在此基础上增加一点准确率
    2. 



       Hard to compare low level motion between frames.
    3. 

       Notemporal shift-invariance! Needs to learn separate filters for the same motion at different times in the clip
    4. 

       Temporal shift-invariant since each filter slides over time!

5. ## C3D: The VGG of 3D CNNs

3D CNN that uses all 3x3x3 conv and 2x2x2 pooling
(except Pool1 which is 1x2x2)

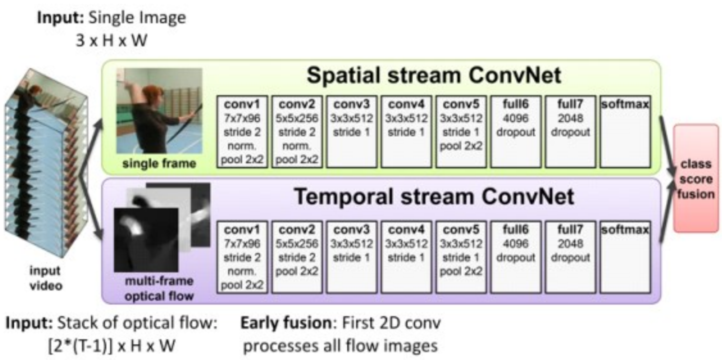
Released model pretrained on Sports-1M: Many people used this as a video feature extractor

Problem: 3x3x3 conv is very expensive!
AlexNet: 0.7 GFLOP
VGG-16: 13.6 GFLOP
C3D: **39.5 GFLOP (2.9x VGG!)** 3D conv to get 1 value.

Layer	Size	MFLOPs
Input	3 x 16 x 112 x 112	
Conv1 (3x3x3)	64 x 16 x 112 x 112	1.04
Pool1 (1x2x2)	64 x 16 x 56 x 56	
Conv2 (3x3x3)	128 x 16 x 56 x 56	11.10
Pool2 (2x2x2)	128 x 8 x 28 x 28	
Conv3a (3x3x3)	256 x 8 x 28 x 28	5.55
Conv3b (3x3x3)	256 x 8 x 28 x 28	11.10
Pool3 (2x2x2)	256 x 4 x 14 x 14	
Conv4a (3x3x3)	512 x 4 x 14 x 14	2.77
Conv4b (3x3x3)	512 x 4 x 14 x 14	5.55
Pool4 (2x2x2)	512 x 2 x 7 x 7	
Conv5a (3x3x3)	512 x 2 x 7 x 7	0.69
Conv5b (3x3x3)	512 x 2 x 7 x 7	0.69
Pool5	512 x 1 x 3 x 3	
FC6	4096	0.51
FC7	4096	0.45
FC8	C	0.05

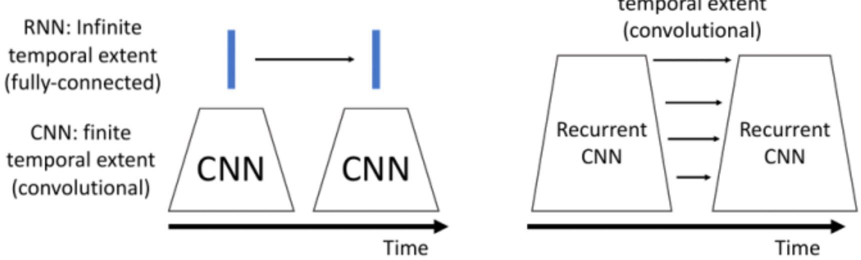
4. Recognize motions 受启发：人类不用看图像就能通过motion cues 认出动作
   1. Optical flow gives a displacement field F between images It and It+1
      Tells where each pixel will move in the next frame:
      Optical Flow highlights local motion

Separating Motion and Appearance: Two-Stream Networks

2. 

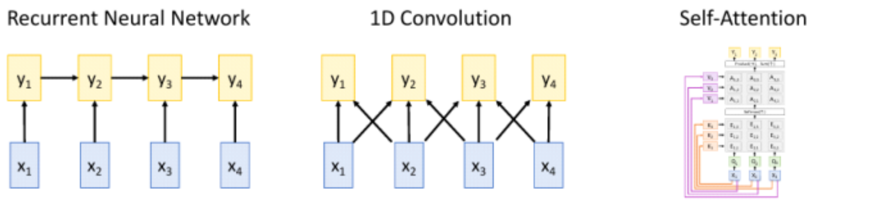
5. Modeling long-term temporal structure
   1. Process local features using recurrent network (e.g. LSTM)
      Manyto many: one output per video frame
      Sometimes don’t backprop to CNN to save memory;
      pretrain and use it as a feature extractor

      Problem: RNNs are slow for long sequences (can’t be parallelized)

      i. 

   2. Spatio-Temporal Self-Attention

Recall: Different ways of processing sequences

i. 

Recurrent Neural Network
Works on **Ordered Sequences**
(+) Good at long sequences: After one RNN layer, h_T "sees" the whole sequence
(-) Not parallelizable: need to compute hidden states sequentially
In video: CNN+RNN, or recurrent CNN

1D Convolution
Works on **Multidimensional Grids**
(-) Bad at long sequences: Need to stack many conv layers for outputs to "see" the whole sequence
(+) Highly parallel: Each output can be computed in parallel
In video: 3D convolution

Self-Attention
Works on **Sets of Vectors**
(-) Good at long sequences: after one self-attention layer, each output "sees" all inputs!
(+) Highly parallel: Each output can be computed in parallel
(-) Very memory intensive
In video: ????

Recall: Self-Attention

ii.

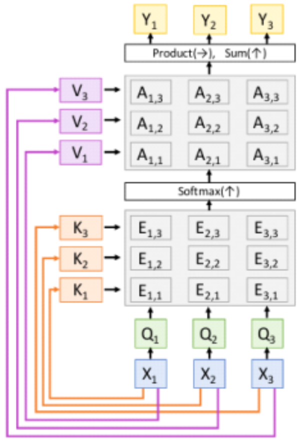
Input: Set of vectors $x_1, \ldots, x_N$

Keys, Queries, Values: Project each x to a key, query, and value using linear layer

Affinity matrix: Compare each pair of x, (using scaled dot-product between keys and values) and normalize using softmax

Output: Weighted sum of values, with weights given by affinity matrix

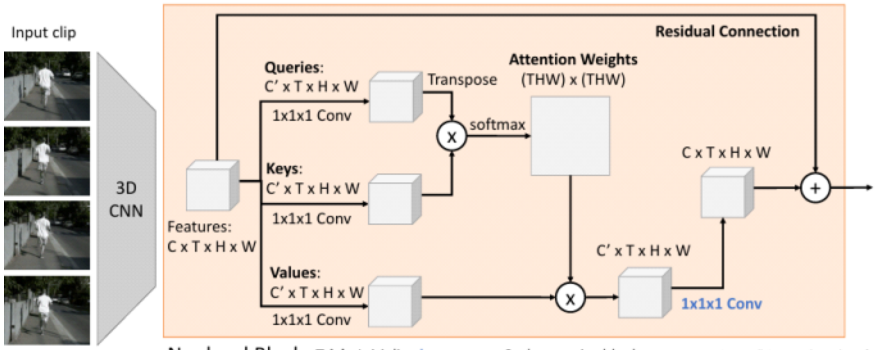
Features in 3D CNN: C x T x H x W
Interpret as a set of THW vectors of dim C



t al, "Attention is all you need", NeurIPS 2017

Spatio-Temporal Self-Attention (Nonlocal Block)

iii.

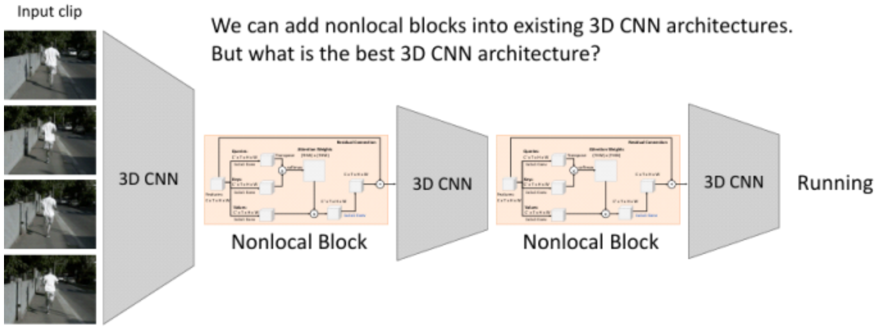


Nonlocal Block **Trick:** Initialize last conv to 0, then entire block computes identity. Can insert into existing 3D CNNs

In practice, actually insert BatchNorm layer after final conv, and initialize scale parameter of BN layer to 0 rather than setting conv weight to 0

iv.

We can add nonlocal blocks into existing 3D CNN architectures.
But what is the best 3D CNN architecture?



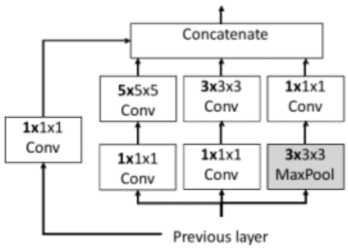
Inflating 2D Networks to 3D (I3D)

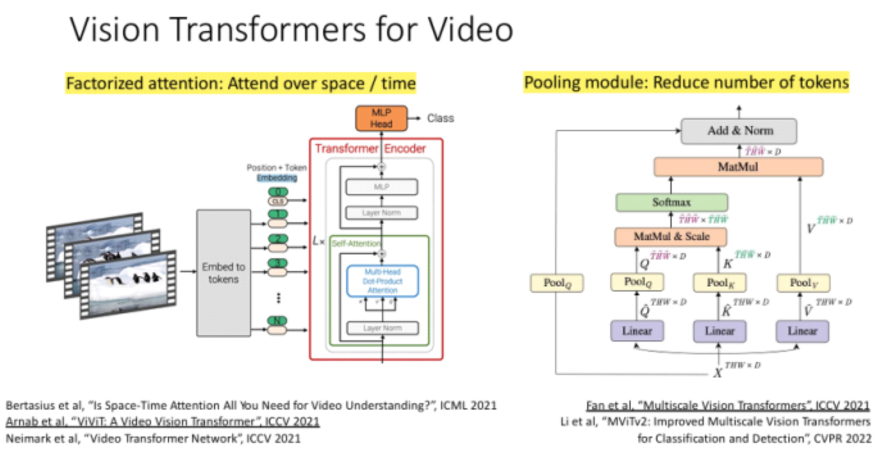
There has been a lot of work on architectures for images.
Can we reuse image architectures for video?

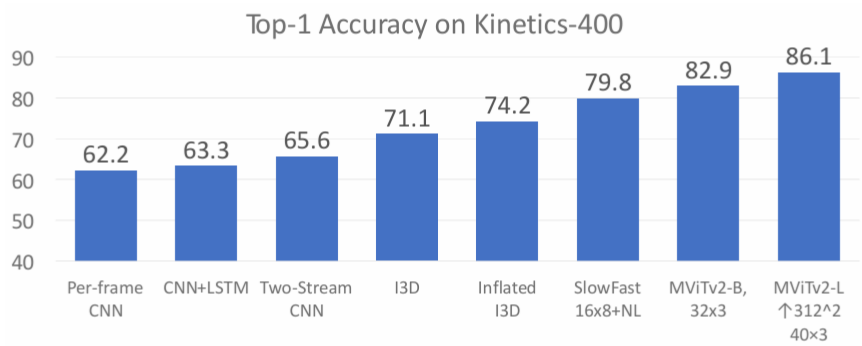
Idea: take a 2D CNN architecture.

3. Replace each 2D K_h x K_w conv/pool layer with a 3D K_t x K_h x K_w version

Inception Block: Inflated



4. 

5. 

6. Other app
   1. visually-guided audio source separation
   2. audio-visual speech separation
   3. co-separating sounds of visual obj
   4. sound source localization