# Master in Computer Vision Barcelona

[http://pagines.uab.cat/mcv/]


Xavier Giro-i-Nieto

@DocXavi

xavier.giro@upc.edu

Associate Professor

Universitat Politècnica de Catalunya


telecos BCN

Institut de Robòtica i Informàtica Industrial

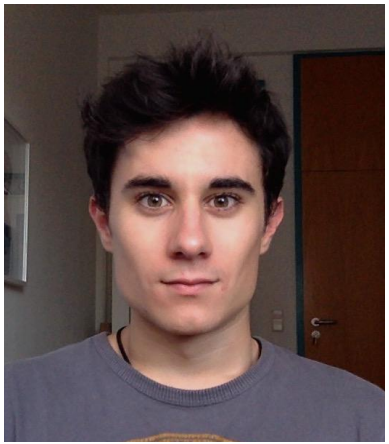
UNIVERSITAT POLITÈCNICA DE CATALUNYA

## Module 6 - Day 9 - Lecture 1

# Neural Architectures for Video Encoding

5th April 2022

# Acknowledgements

Víctor Campos

Amaia Salvador

Alberto Montes

Santiago Pascual

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Video lectures


[Víctor Campos]
[Deep Learning for Computer Vision 2018]
UPC TelecomBCN


[Xavier Giro-i-Nieto]
Master Computer Vision Barcelona 2020
UPC TelecomBCN


**#DeepVideo** Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. . Large-scale video classification with convolutional neural networks. CVPR 2014

# Motivation

track cycling
cycling
track cycling
road bicycle racing
marathon
ultramarathon

ultramarathon
ultramarathon
half marathon
running
marathon
inline speed skating

heptathlon
heptathlon
decathlon
hurdles
pentathlon
sprint (running)

bikejoring
mushing
bikejoring
harness racing
skijoring
carting

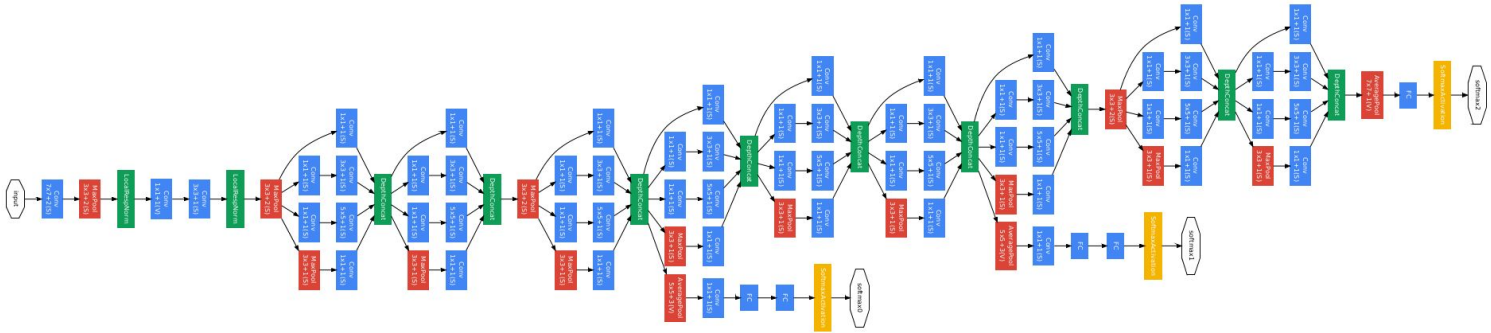
**#DeepVideo** Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. Large-scale video classification with convolutional neural networks. CVPR 2014.

# What is a video?

- Formally, a video is a 3D signal
  - Spatial coordinates: *x*, *y*
  - Temporal coordinate: *t*
- If we fix *t*, we obtain an image. We can understand videos as sequences of images (a.k.a. frames)

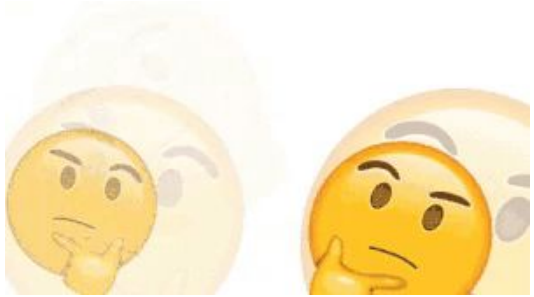
# How do we work with images?

**Convolutional Neural Networks (CNN)** provide state of the art performance on still image analysis tasks



# How do we work with videos ?

How can we extend CNNs to image sequences?



# Deep Video Architectures

Basic deep architectures for video:

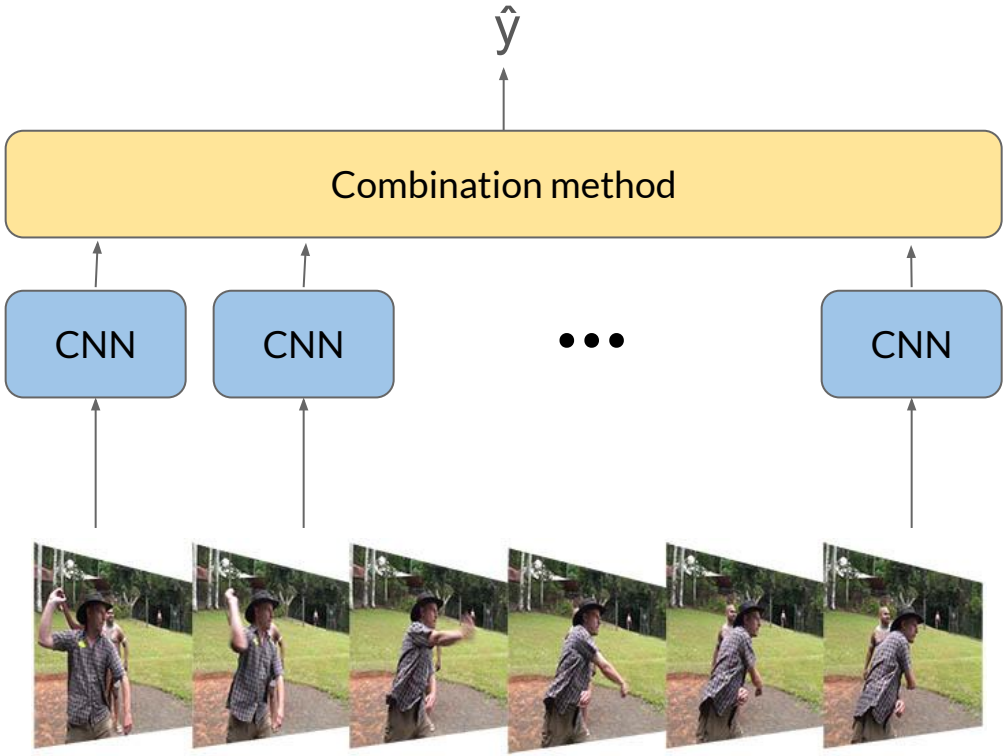
1. Single frame models
2. Spatio-temporal convolutions
3. CNN + RNN
4. RGB + Optical Flow
5. Transformers
6. Miscellaneous

# Deep Video Architectures

Basic deep architectures for video:

1. **Single frame models**
2. Spatio-temporal convolutions
3. CNN + RNN
4. RGB + Optical Flow
5. Transformers
6. Miscellaneous

# Single frame models




Combination is commonly implemented as a small NN on top of a **temporal pooling** operation (e.g. max, average) over the features.

Problem: pooling is not aware of the temporal order!

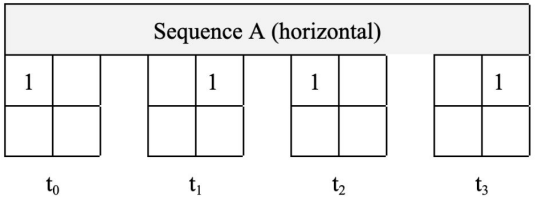
Yue-Hei Ng, Joe, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." CVPR 2015

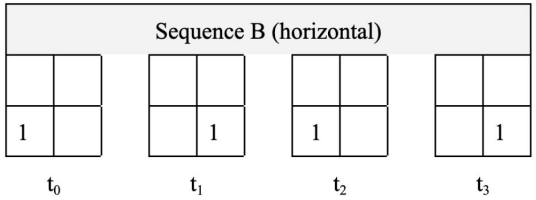
# Exercise

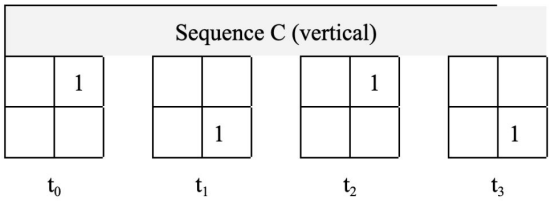
Consider a dataset of very short videos of 4 frames long, and 2x2 pixels of spatial definition. All videos show a 1x1 pixel over a background that may bounce in two different directions, depending on the class of the video: horizontally or vertically.

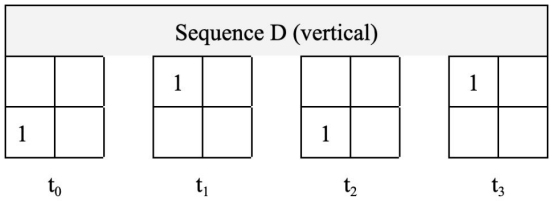
The starting coordinates of the pixel are randomly chosen within the dataset. For the purpose of this exercise, consider the following four sequences (A, B, C, D)

Sequences A & B: Horizontally (left-right).
Sequences C & D: Vertically (top-down).




The ‘1’ cells corresponds to the bouncing pixel, and all blank cells correspond to the background, encoded with a ‘0’ value:

The main purpose of this problem is studying different neural architectures for the task of video classification between the horizontal or vertical classes.

# Exercise: Conv-2D Features

CONV-2D FEATURE EXTRACTION

Study the case of a very simple feature extractor composed of a single 2D convolutional layer defined by the following four 2x2 filters, K={1,2,3,4}:

| K=1 | |
|---|---|
| 1 | |
| | |

| K=2 | |
|---|---|
| | 1 |
| | |

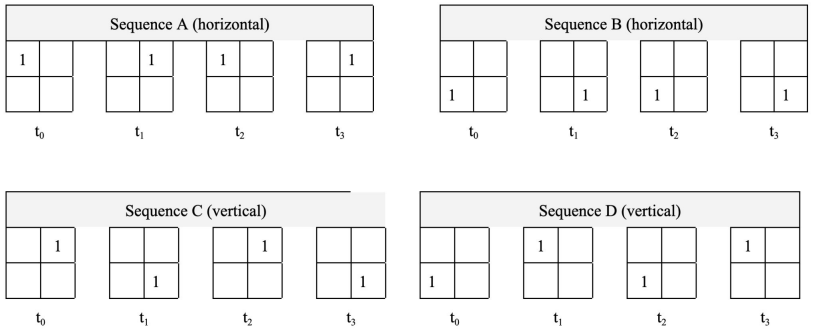
| K=3 | |
|---|---|
| | |
| 1 | |

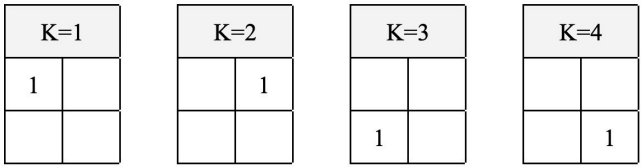
| K=4 | |
|---|---|
| | |
| | 1 |

a) Compute the output of the four filters after the ReLU layer. Provide your answers in the corresponding $t_0$, $t_1$, $t_2$ and $t_3$ columns of the blank grids.

# Exercise: Conv-2D

a) Compute the output of the four filters after a ReLU layer. Provide your answers in the corresponding $t_0$, $t_1$, $t_2$ and $t_3$ columns of the blank grids.




| K |
|---|
| 1 |
| 2 |
| 3 |
| 4 |

Sequence A (horizontal)

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

$t_0$ $t_1$ $t_2$ $t_3$

Sequence B (horizontal)

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

$t_0$ $t_1$ $t_2$ $t_3$

| K |
|---|
| 1 |
| 2 |
| 3 |
| 4 |

Sequence C (vertical)

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

$t_0$ $t_1$ $t_2$ $t_3$

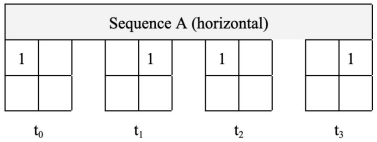
Sequence D (vertical)

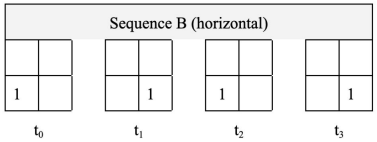
| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

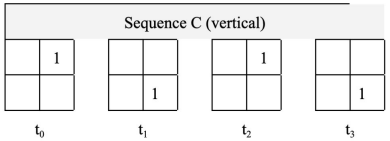
$t_0$ $t_1$ $t_2$ $t_3$

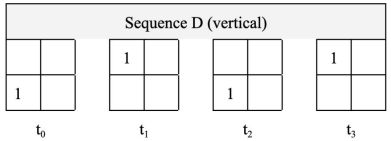
# Solution: Conv-2D

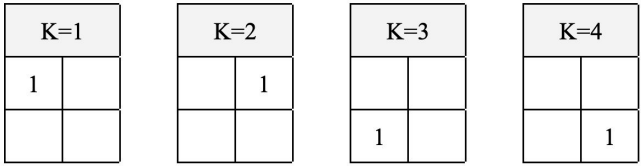
a) Compute the output of the four filters after the ReLU layer. Provide your answers in the corresponding $t_0$, $t_1$, $t_2$ and $t_3$ columns of the blank grids.




| K | Sequence A (horizontal) | | | |
|---|---|---|---|---|
| 1 | 1 | | 1 | |
| 2 | | 1 | | 1 |
| 3 | | | | |
| 4 | | | | |
| | $t_0$ | $t_1$ | $t_2$ | $t_3$ |

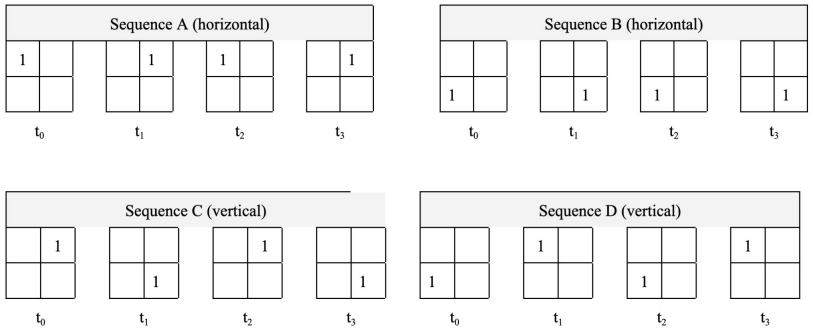
| Sequence B (horizontal) | | | |
|---|---|---|---|
| | | | |
| | | | |
| 1 | | 1 | |
| | 1 | | 1 |
| $t_0$ | $t_1$ | $t_2$ | $t_3$ |

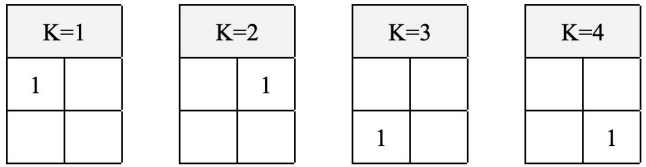
| K | Sequence C (vertical) | | | |
|---|---|---|---|---|
| 1 | | | | |
| 2 | 1 | | 1 | |
| 3 | | | | |
| 4 | | 1 | | 1 |
| | $t_0$ | $t_1$ | $t_2$ | $t_3$ |

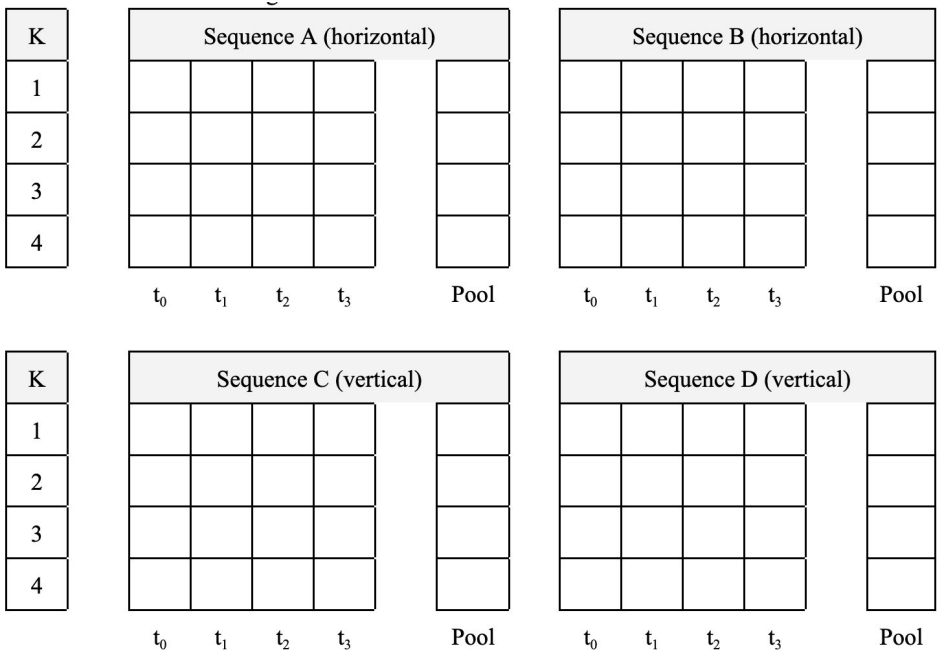
| Sequence D (vertical) | | | |
|---|---|---|---|
| | 1 | | 1 |
| | | | |
| 1 | | 1 | |
| | | | |
| $t_0$ | $t_1$ | $t_2$ | $t_3$ |

# Exercise: Conv-2D + Temporal Pooling

b) Apply a temporal max pooling along the output features of each convolutional filter. Provide your answers in the Pool columns of the blank grids.

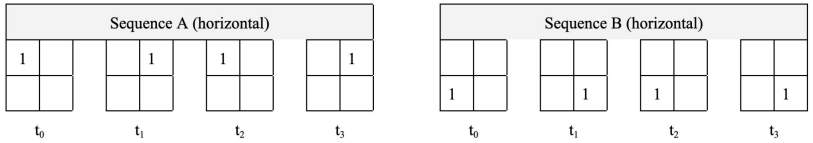


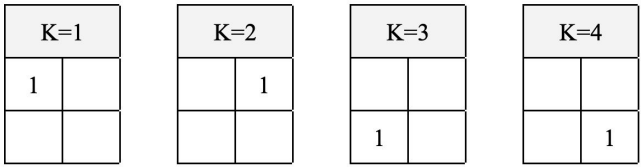



# Solution: : Conv-2D + Temporal Pooling

b) Apply a temporal max pooling along the output features of each convolutional filter. Provide your answers in the Pool columns of the blank grids 2.

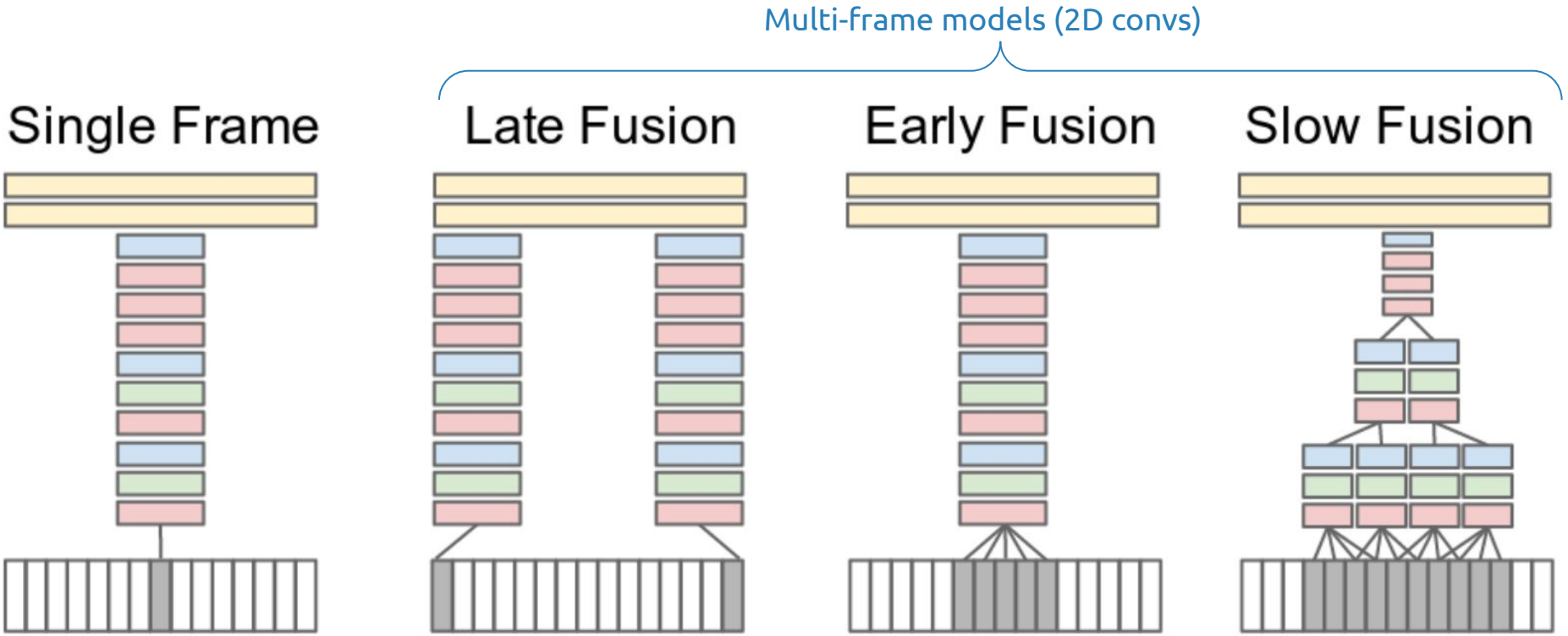



| K | Sequence A (horizontal) | | | | Sequence B (horizontal) | | | |
|---|---|---|---|---|---|---|---|---|
| | $t_0$ | $t_1$ | $t_2$ | $t_3$ | $t_0$ | $t_1$ | $t_2$ | $t_3$ |
| 1 | 1 | | 1 | | | | | |
| 2 | | 1 | | 1 | | | | |
| 3 | | | | | 1 | | 1 | |
| 4 | | | | | | 1 | | 1 |
| Pool | 1 | 1 | | | | | 1 | 1 |

| K | Sequence C (vertical) | | | | Sequence D (vertical) | | | |
|---|---|---|---|---|---|---|---|---|
| | $t_0$ | $t_1$ | $t_2$ | $t_3$ | $t_0$ | $t_1$ | $t_2$ | $t_3$ |
| 1 | | | | | | 1 | | 1 |
| 2 | 1 | | 1 | | | | | |
| 3 | | | | | 1 | | 1 | |
| 4 | | 1 | | 1 | | | | |
| Pool | 1 | | | 1 | 1 | | 1 | |

# Single vs Multi-frame models (2D convs)




Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. . Large-scale video classification with convolutional neural networks. CVPR 2014

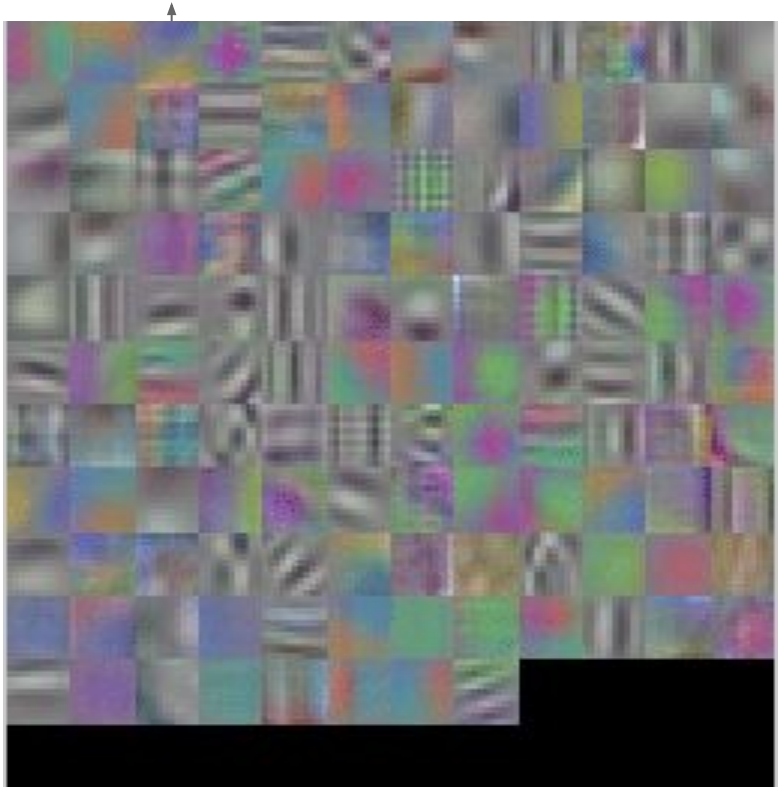
# Single vs Multi-frame models (2D convs)

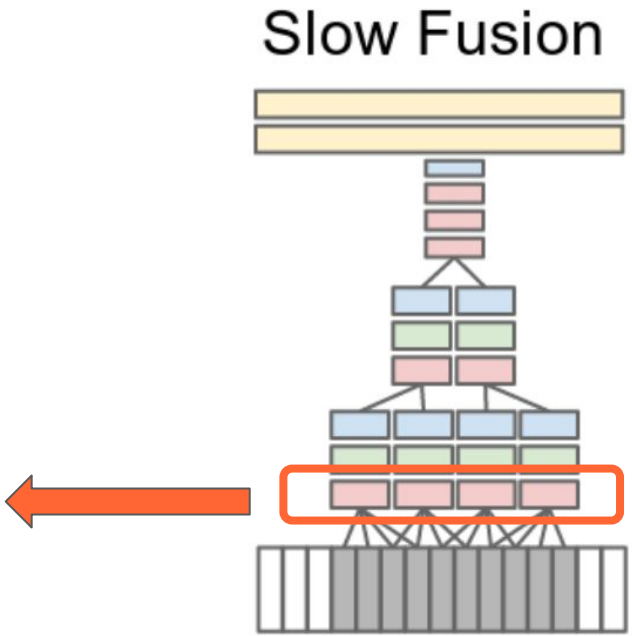


| Model | Clip Hit@1 | Video Hit@1 | Video Hit@5 |
|---|---|---|---|
| Feature Histograms + Neural Net | - | 55.3 | - |
| Single-Frame | 41.1 | 59.3 | 77.7 |
| Early Fusion | 38.9 | 57.7 | 76.8 |
| Late Fusion | 40.7 | 59.3 | 78.7 |
| Slow Fusion | **41.9** | **60.9** | **80.2** |
| CNN Average (Single+Early+Late+Slow) | 41.4 | 63.9 | 82.4 |

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. . Large-scale video classification with convolutional neural networks. CVPR 2014

# Multi-frame models (2D convs)





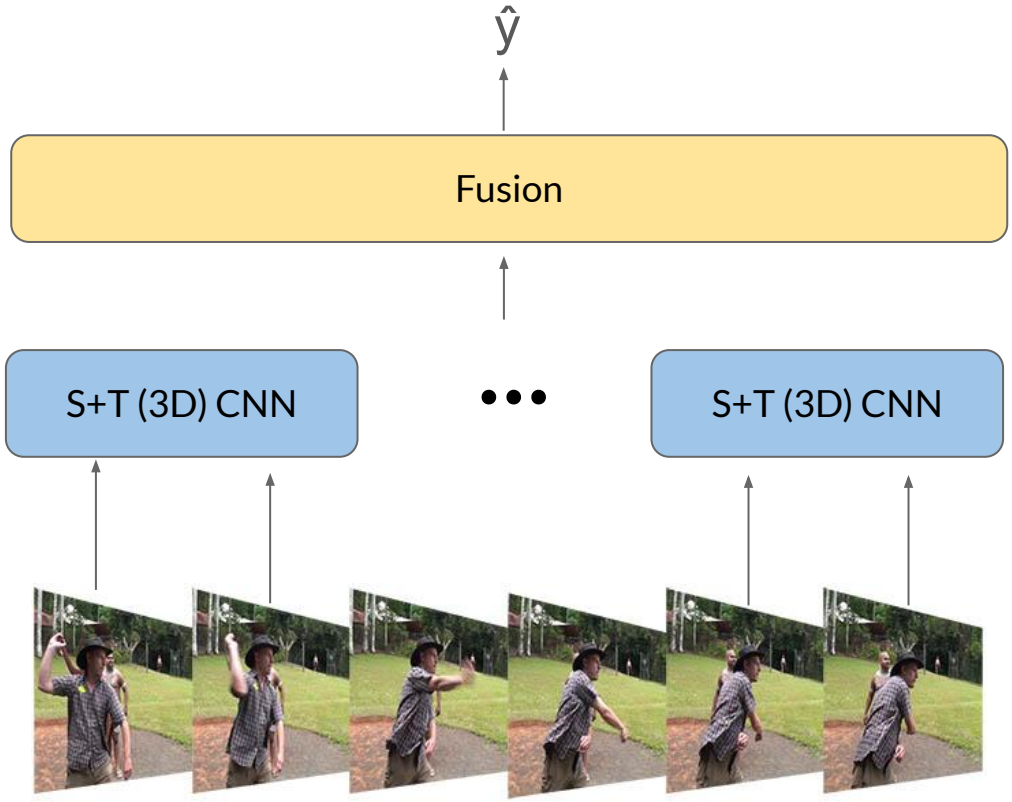

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. . Large-scale video classification with convolutional neural networks. CVPR 2014

# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. **Spatio-temporal convolutions**
3. CNN + RNN
4. RGB + Optical Flow Two-stream CNN
5. Transformers
6. Miscellaneous

# Spatio-Temporal Convolutions



# Reminder: Spatial (2D) Convolutions

A convolutional layer can be understood as a module that transforms some feature maps to other feature maps, one for each convolutional kernel.

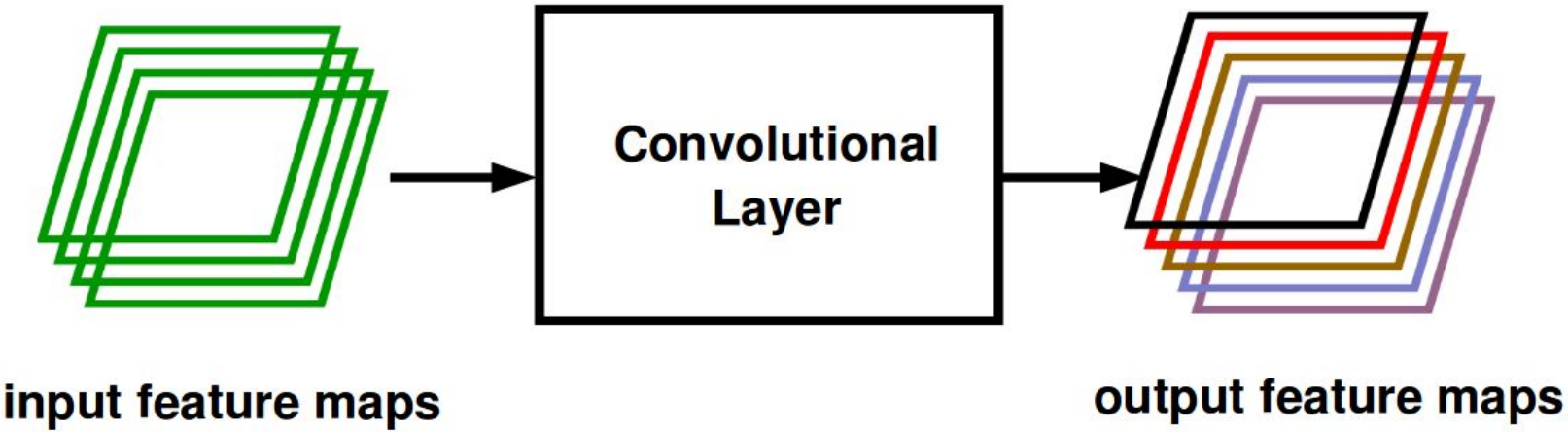
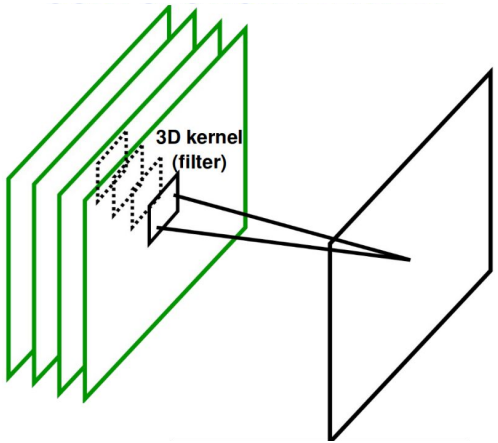



*Figure Credit: Ranzatto*

# Reminder: Spatial (2D) Convolutions

2D Convolutional filters are actually 3D kernels: their depth must match the number of channels of the input tensor.




filter of depth=4

output feature map

Many feature maps

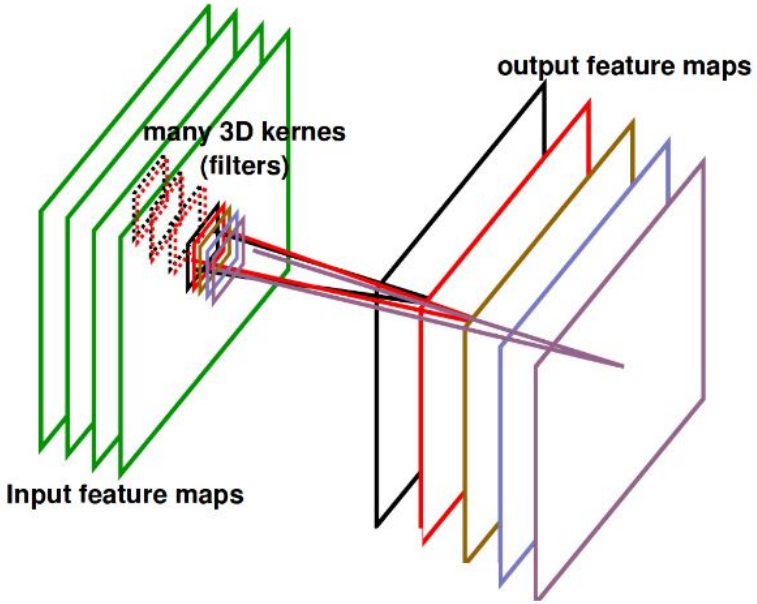
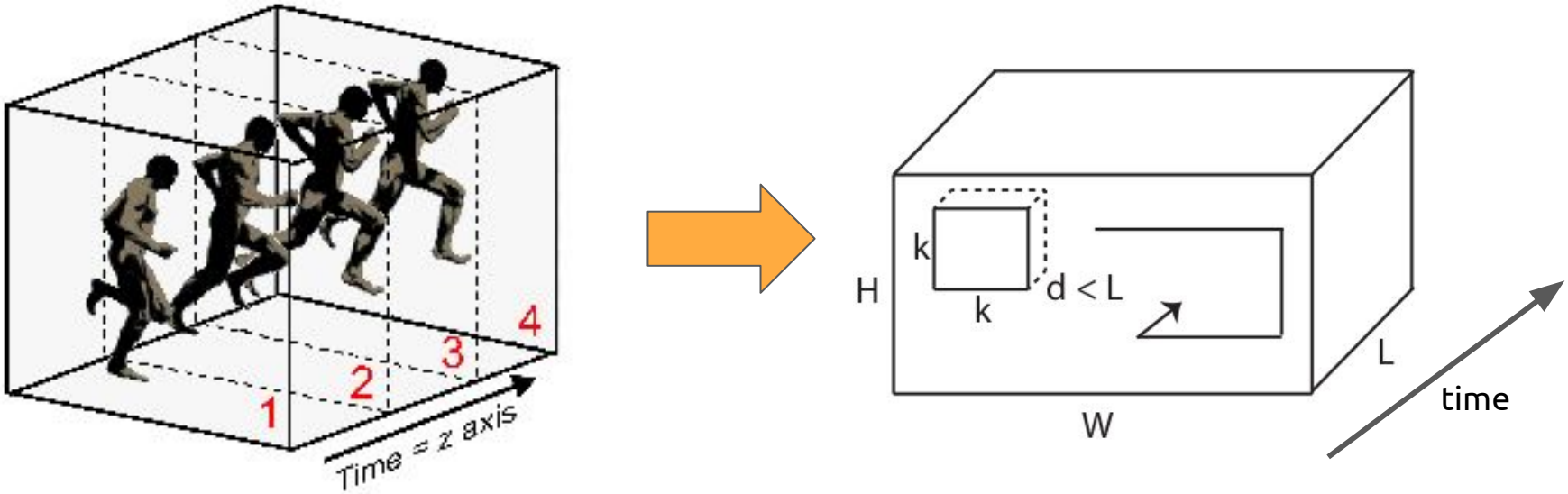



*Figure Credit: Ranzatto*

# 3D CNN (C3D)

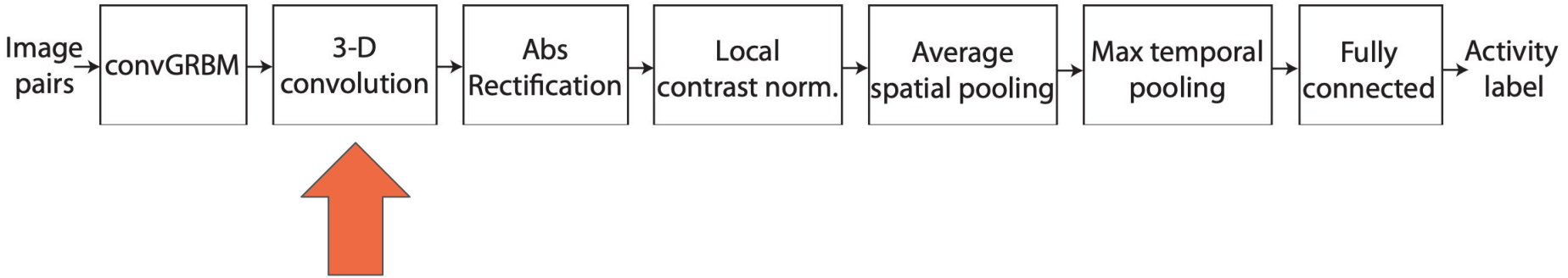
We can add an extra dimension to standard 2D CNN filters:

- A grayscale video clip may be represented by a tensor of size H x W x L.
- A 3D convolutional filter for the 1st conv layer may be of size k x k x d



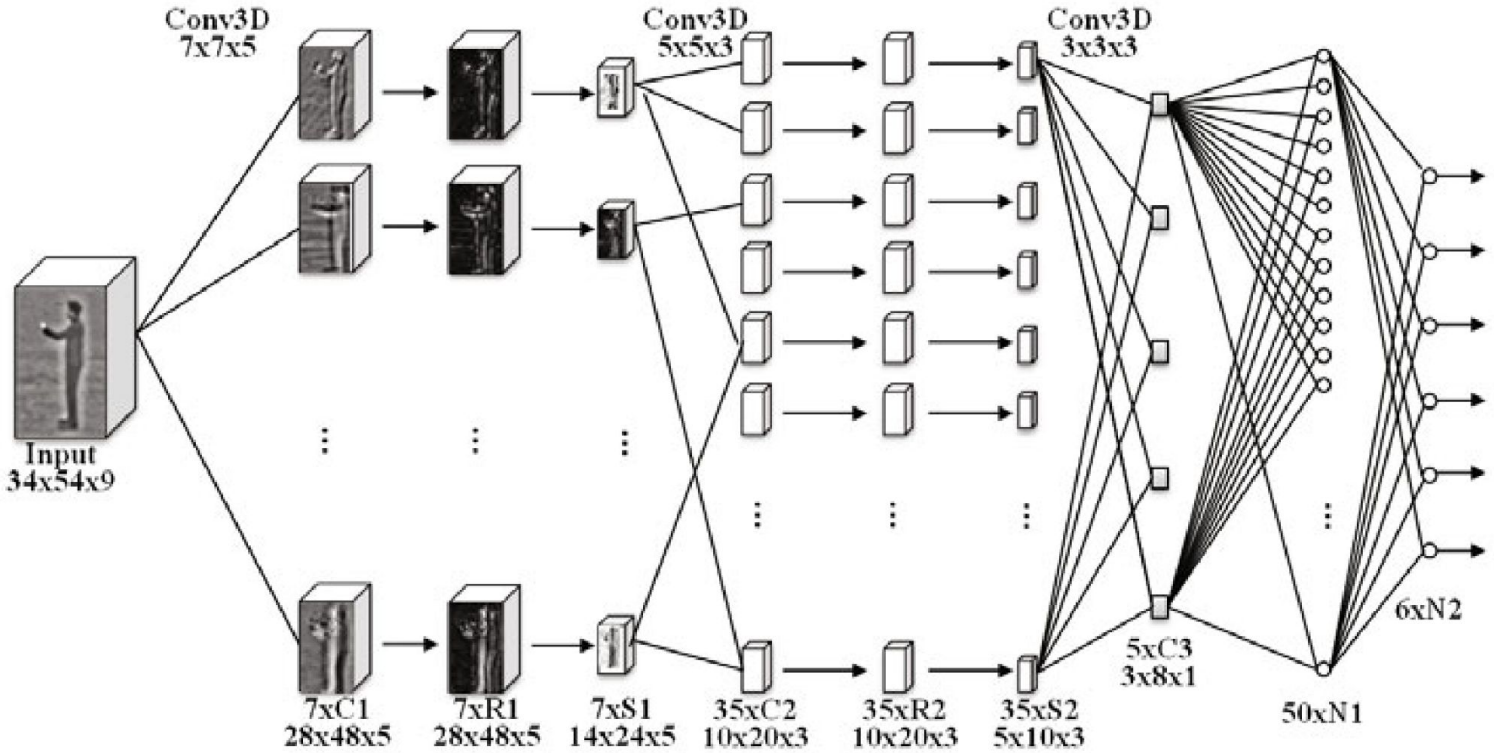

**#C3D** Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3D convolutional networks" ICCV 2015

# Spatio-temporal Convolutions




Taylor, G. W., Fergus, R., LeCun, Y., & Bregler, C. Convolutional learning of spatio-temporal features. ECCV 2010.

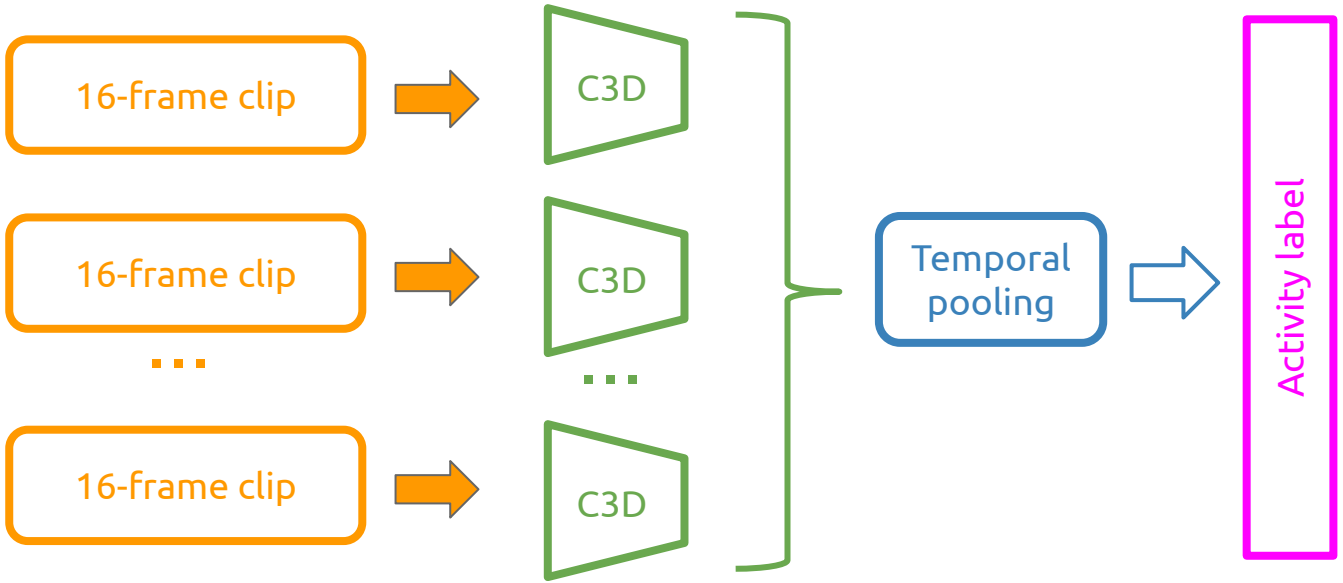
# 3D CNN (C3D)




Baccouche, Moez, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. "Sequential deep learning for human action recognition." HBU 2011.

# 3D CNN (C3D) + temporal pooling

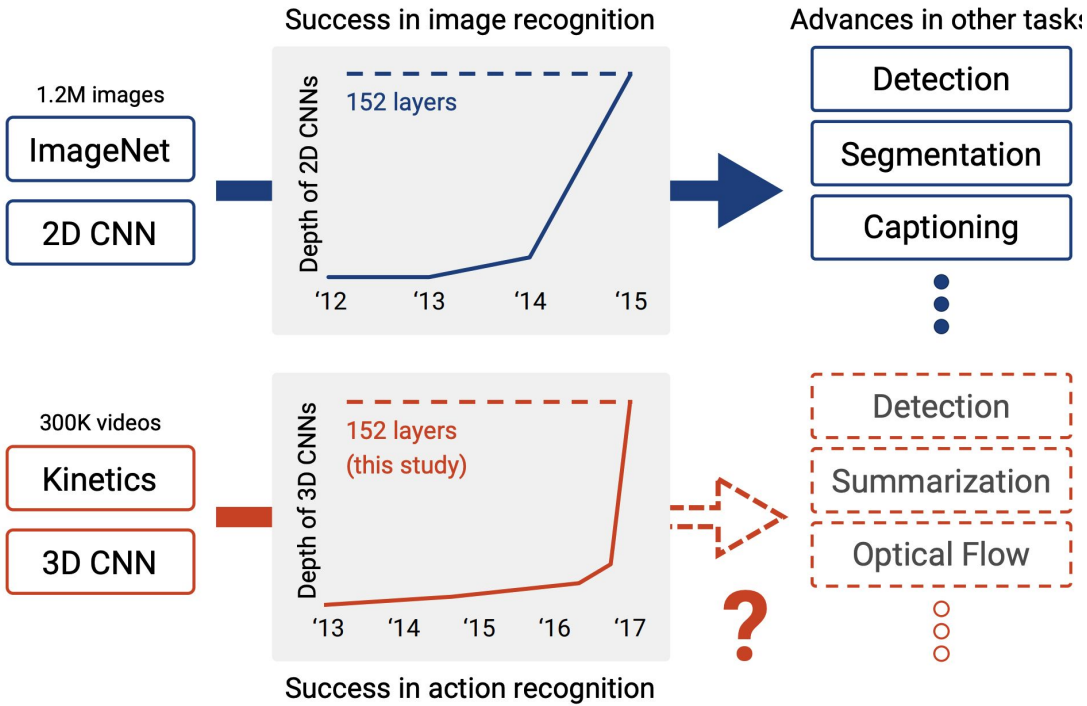
- The full video is split in chunks (also known as *clips*). Eg. 16-frames per clip.
- Each clip is processed separately with 3D Convolutional Filters.
- Predictions for each clip must be later combined. Eg. Average pooling.




**#C3D** Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3D convolutional networks" ICCV 2015

# 3D CNN (C3D)

Large video datasets (eg. Kinetics) allow training deep C3D CNNs.




**#3D Resnet** Hara, Kensho, Hirokatsu Kataoka, and Yutaka Satoh. "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?." CVPR 2018. [code]

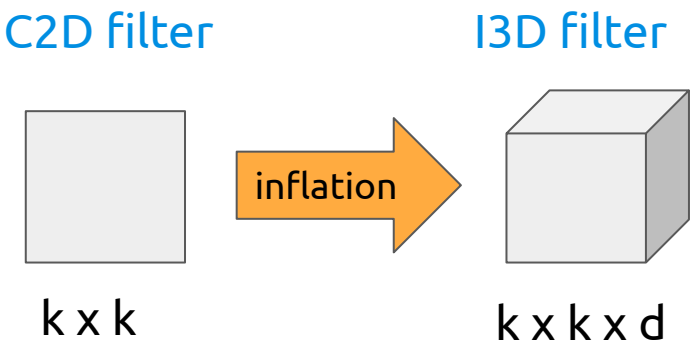
# 3D CNN

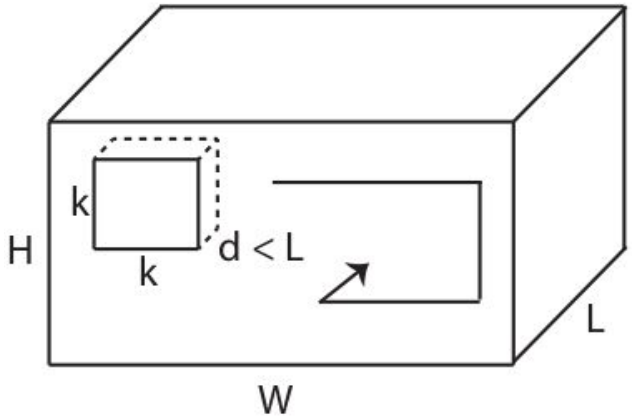
Limitation:

- How can we use pre-trained 2D networks to initialize C3D training ?

# Inflated 3D weights (I3D)

We can adapt C2D filters to C3D by **replicating** them along the temporal axis (and scaling its values by 1/d).



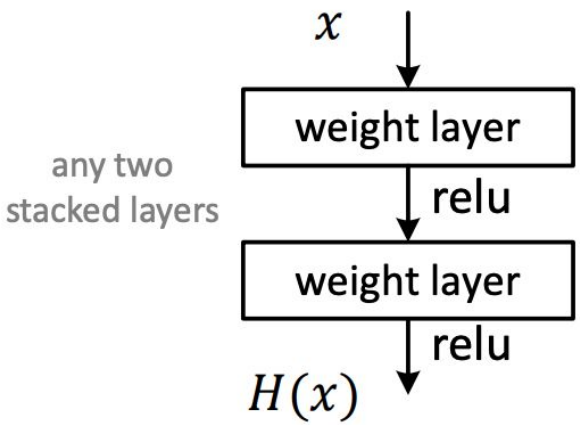




Filter **inflation** allows reusing C2D pre-trained filters (eg. from ImageNet) to initialize C3D filters.

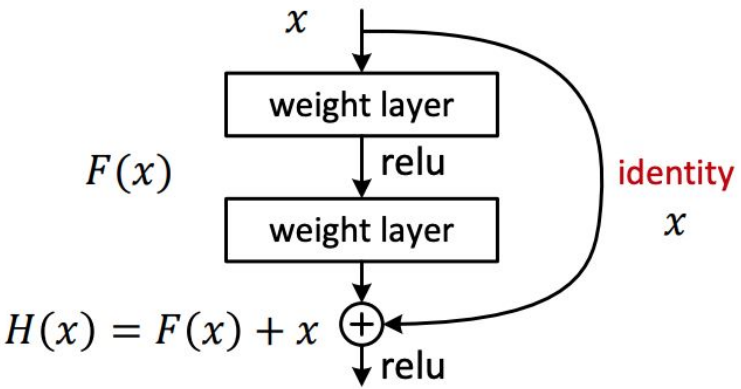
**#I3D** Carreira, J., & Zisserman, A. . Quo vadis, action recognition? A new model and the kinetics dataset. CVPR 2017. [code]

# Residual Learning (reminder)

Plain Net




Residual Net




Residual learning: reformulate the layers as learning residual functions with respect to the identity F(x), instead of learning unreferenced functions H(x)



He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." CVPR 2016. [slides]

# Residual Learning (reminder)

## 2D ResNet

| layer name | output size | 18-layer | 34-layer |
|---|---|---|---|
| conv1 | 112×112 | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ |
| | 1×1 | ave | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ |

How should the a 2D ResNet architecture be modified to consider 3D Convolutions ?

# 3D CNN + Residual (Res3D)

## 2D ResNet

| layer name | output size | 18-layer | 34-layer |
|---|---|---|---|
| conv1 | 112×112 | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ |
| | 1×1 | | ave |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ |

## 3D ResNet

| layer name | output size | 3D-Resnet18 | 3D-Resnet34 |
|---|---|---|---|
| conv1 | 8×56×56 | 3×7×7, 64, stride 1 × 2× 2 | |
| conv2_x | 8×56×56 | $\begin{bmatrix} 3\times3\times3, 64 \\ 3\times3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3\times3, 64 \\ 3\times3\times3, 64 \end{bmatrix}\times3$ |
| conv3_x | 4×28×28 | $\begin{bmatrix} 3\times3\times3, 128 \\ 3\times3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3\times3, 128 \\ 3\times3\times3, 128 \end{bmatrix}\times4$ |
| conv4_x | 2×14×14 | $\begin{bmatrix} 3\times3\times3, 256 \\ 3\times3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3\times3, 256 \\ 3\times3\times3, 256 \end{bmatrix}\times6$ |
| conv5_x | 1×7×7 | $\begin{bmatrix} 3\times3\times3, 512 \\ 3\times3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3\times3, 512 \\ 3\times3\times3, 512 \end{bmatrix}\times3$ |
| | 1×1×1 | average pool, 101-d fc, softmax | |

**#Res3D** Tran, Du, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. "Convnet architecture search for spatiotemporal feature learning." arXiv preprint arXiv:1708.05038 (2017). [code]

# 3D CNN + Residual (Res3D)

| Method | Clip@1 | Video@1 | Video@5 |
|---|---|---|---|
| single model, no long-term modeling | | | |
| DeepVideo [14] | 41.9 | 60.9 | 80.2 |
| C3D [41] | 46.1 | 61.1 | 85.2 |
| AlexNet [24] | N/A | 63.6 | 84.7 |
| GoogleNet [24] | N/A | 64.9 | 86.6 |
| 2D-Resnet* | 45.5 | 59.4 | 83.0 |
| **Res3D (ours)*** | **48.8** | **65.6** | **87.8** |

**#Res3D** Tran, Du, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. "Convnet architecture search for spatiotemporal feature learning." arXiv preprint arXiv:1708.05038 (2017). [code]

# 3D CNN + Residual (Res3D)

Is the comparison in terms of accuracy (Clip@1, Video@1, Video@5) complete ?

## 2D ResNet

| layer name | output size | 18-layer | 34-layer |
|---|---|---|---|
| conv1 | 112×112 | | |
| conv2_x | 56×56 | [3×3, 64; 3×3, 64] ×2 | [3×3, 64; 3×3, 64] ×3 |
| conv3_x | 28×28 | [3×3, 128; 3×3, 128] ×2 | [3×3, 128; 3×3, 128] ×4 |
| conv4_x | 14×14 | [3×3, 256; 3×3, 256] ×2 | [3×3, 256; 3×3, 256] ×6 |
| conv5_x | 7×7 | [3×3, 512; 3×3, 512] ×2 | [3×3, 512; 3×3, 512] ×3 |
| | 1×1 | | ave |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ |

## Res3D

| layer name | output size | 3D-Resnet18 | 3D-Resnet34 |
|---|---|---|---|
| conv1 | 8×56×56 | 3×7×7, 64, stride 1 × 2× 2 | |
| conv2_x | 8×56×56 | [3×3×3, 64; 3×3×3, 64] ×2 | [3×3×3, 64; 3×3×3, 64] ×3 |
| conv3_x | 4×28×28 | [3×3×3, 128; 3×3×3, 128] ×2 | [3×3×3, 128; 3×3×3, 128] ×4 |
| conv4_x | 2×14×14 | [3×3×3, 256; 3×3×3, 256] ×2 | [3×3×3, 256; 3×3×3, 256] ×6 |
| conv5_x | 1×7×7 | [3×3×3, 512; 3×3×3, 512] ×2 | [3×3×3, 512; 3×3×3, 512] ×3 |
| | 1×1×1 | average pool, 101-d fc, softmax | |

| Method | Clip@1 | Video@1 | Video@5 |
|---|---|---|---|
| single model, no long-term modeling | | | |
| DeepVideo [14] | 41.9 | 60.9 | 80.2 |
| C3D [41] | 46.1 | 61.1 | 85.2 |
| AlexNet [24] | N/A | 63.6 | 84.7 |
| GoogleNet [24] | N/A | 64.9 | 86.6 |
| 2D-Resnet* | 45.5 | 59.4 | 83.0 |
| **Res3D (ours)*** | **48.8** | **65.6** | **87.8** |

**#Res3D** Tran, Du, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. "Convnet architecture search for spatiotemporal feature learning." arXiv preprint arXiv:1708.05038 (2017). [code]

# 3D CNN + Residual (Res3D)

Gain in performance comes with a much larger cost in:

- Memory footprint (#params)
- Computation (FLOPS = Floating Point Operations per Second).

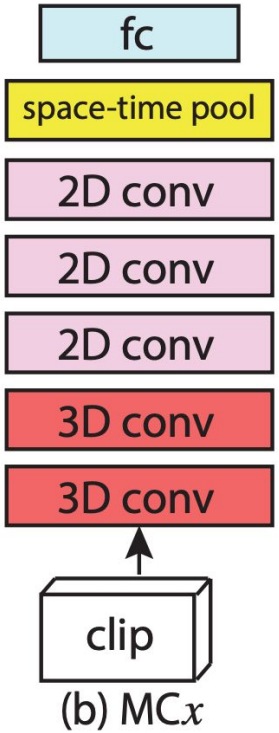
2D ResNet

3D ResNet

| Net | 2D-Res18 | 2D-Res34 | 3D-Res18 | 3D-Res34 |
|---|---|---|---|---|
| # params ($\times 10^6$) | 11.2 | 21.5 | 33.2 | 63.5 |
| FLOPs ($\times 10^9$) | 1.6 | 3.5 | 19.3 | 36.7 |
| Accuracy (%) | 42.2 | 42.2 | 45.6 | 45.9 |

**#Res3D** Tran, Du, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. "Convnet architecture search for spatiotemporal feature learning." arXiv preprint arXiv:1708.05038 (2017). [code]

# Mixed Convolutions (MCx)



(b) MC*x*

Mixed Convolutions (MC*x*)

- *x* First layers: 3D convolutions
- Deeper layers: 2D convolutions

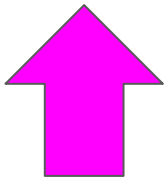
**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Mixed Convolutions (MC & rMC)

MC ResNets:

- 3-4% gain in clip-level accuracy over 2D ResNets of comparable capacity.

| Net | # params | Clip@1 | Video@1 | Clip@1 | Video@1 |
|---|---|---|---|---|---|
| Input | | 8×112×112 | | 16×112×112 | |
| R2D | 11.4M | 46.7 | 59.5 | 47.0 | 58.9 |
| f-R2D | 11.4M | 48.1 | 59.4 | 50.3 | 60.5 |
| R3D | 33.4M | 49.4 | 61.8 | 52.5 | 64.2 |
| MC2 | 11.4M | 50.2 | 62.5 | 53.1 | 64.2 |



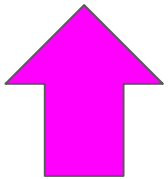
**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Mixed Convolutions (MC & rMC)

MC ResNets:

- match the performance of 3D ResNets, which have 3 times as many parameters.

| Net | # params | Clip@1 | Video@1 | Clip@1 | Video@1 |
|---|---|---|---|---|---|
| Input | | 8×112×112 | | 16×112×112 | |
| R2D | 11.4M | 46.7 | 59.5 | 47.0 | 58.9 |
| f-R2D | 11.4M | 48.1 | 59.4 | 50.3 | 60.5 |
| R3D | 33.4M | 49.4 | 61.8 | 52.5 | 64.2 |
| MC2 | 11.4M | 50.2 | 62.5 | 53.1 | 64.2 |



**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Mixed Convolutions (MC & rMC)

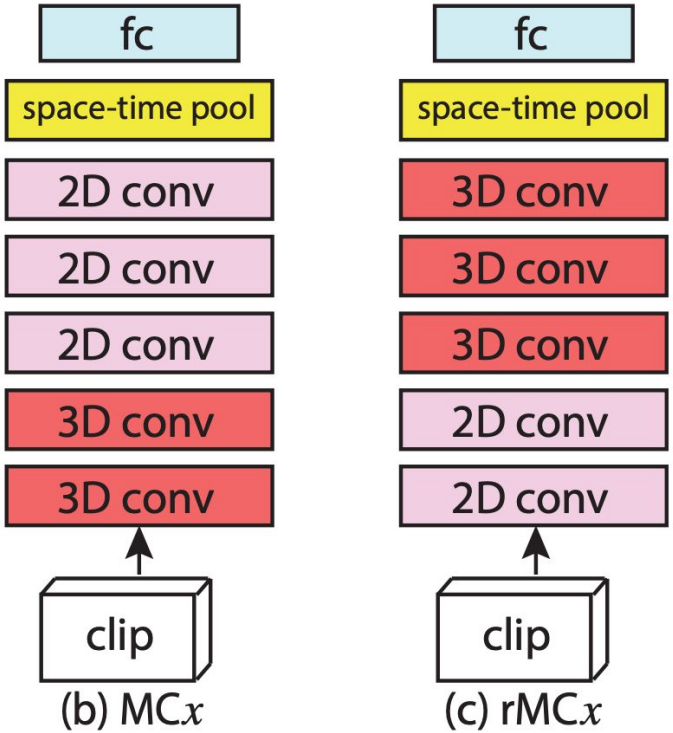
MC ResNets:

- match the performance of 3D ResNets, which have 3 times as many parameters.

| Net | # params | Clip@1 | Video@1 | Clip@1 | Video@1 |
|---|---|---|---|---|---|
| Input | | 8×112×112 | | 16×112×112 | |
| R2D | 11.4M | 46.7 | 59.5 | 47.0 | 58.9 |
| f-R2D | 11.4M | 48.1 | 59.4 | 50.3 | 60.5 |
| R3D | 33.4M | 49.4 | 61.8 | 52.5 | 64.2 |
| MC2 | 11.4M | 50.2 | 62.5 | 53.1 | 64.2 |
| MC3 | 11.7M | 50.7 | 62.9 | 53.7 | 64.7 |
| MC4 | 12.7M | 50.5 | 62.5 | 53.7 | 65.1 |
| MC5 | 16.9M | 50.3 | 62.5 | 53.7 | 65.1 |

**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Reversed Mixed Convolutions(rMC*x*)




Mixed Convolutions (MC*x*)

- *x* First layers: 3D convolutions
- Deeper layers: 2D convolutions

Reversed Mixed Convolutions (rMC*x*)

- *x* First layers: 2D convolutions
- Deeper layers: 3D convolutions

**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Mixed Convolutions (MC vs rMC)

MC ResNets:

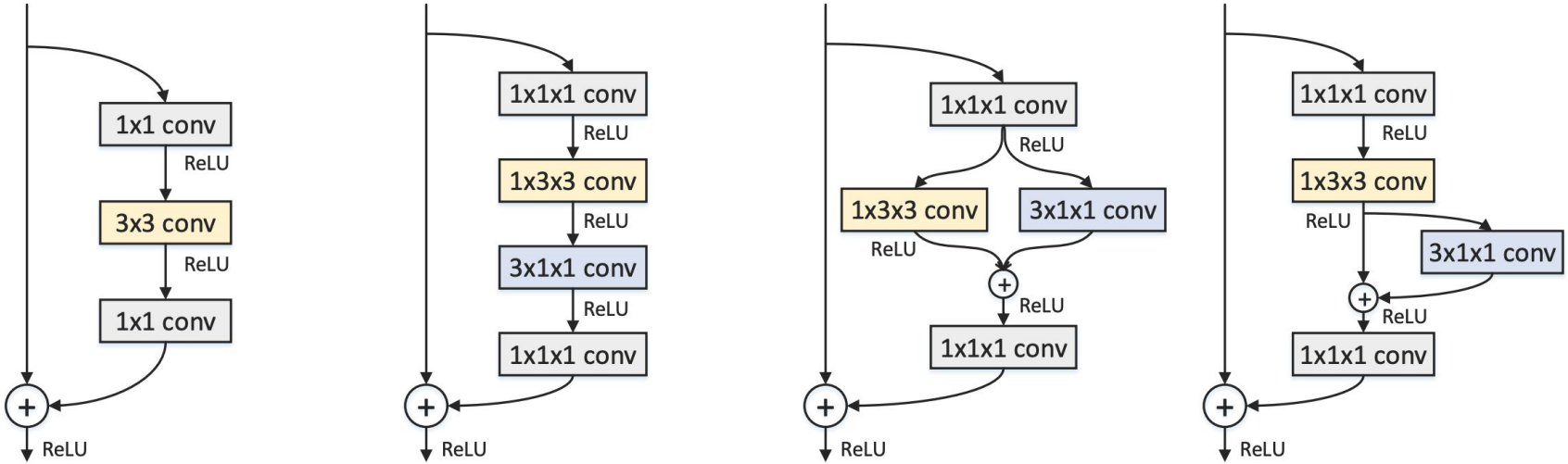
- match the performance of rMC ResNets, which have almost 3 times as many parameters.

| Net | # params | Clip@1 | Video@1 | Clip@1 | Video@1 |
|---|---|---|---|---|---|
| Input | | 8×112×112 | | 16×112×112 | |
| R2D | 11.4M | 46.7 | 59.5 | 47.0 | 58.9 |
| f-R2D | 11.4M | 48.1 | 59.4 | 50.3 | 60.5 |
| R3D | 33.4M | 49.4 | 61.8 | 52.5 | 64.2 |
| MC2 | 11.4M | 50.2 | 62.5 | 53.1 | 64.2 |
| MC3 | 11.7M | 50.7 | 62.9 | 53.7 | 64.7 |
| MC4 | 12.7M | 50.5 | 62.5 | 53.7 | 65.1 |
| MC5 | 16.9M | 50.3 | 62.5 | 53.7 | 65.1 |
| rMC2 | 33.3M | 49.8 | 62.1 | 53.1 | 64.9 |
| rMC3 | 33.0M | 49.8 | 62.3 | 53.2 | 65.0 |
| rMC4 | 32.0M | 49.9 | 62.3 | 53.4 | 65.1 |
| rMC5 | 27.9M | 49.4 | 61.2 | 52.1 | 63.1 |

**#MC** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# Pseudo-3D + Residual (P3D)

Residual connections with 3 variations to combine Spatial (S - 1x3x3) and Temporal (T - 3x1x1) convolutions.

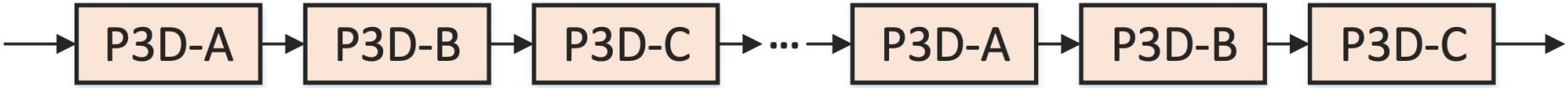


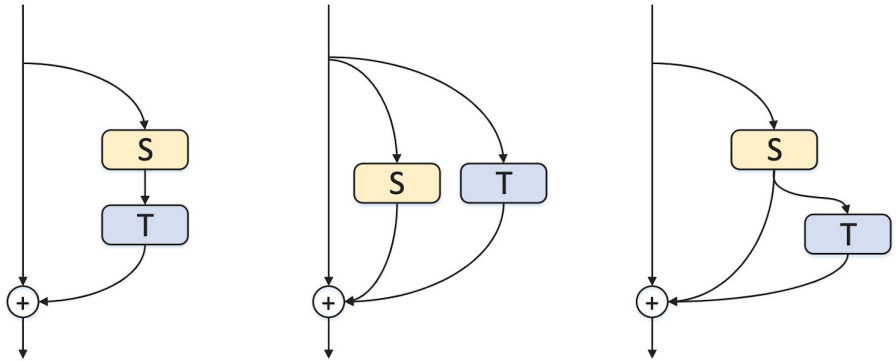

(a) Residual Unit [7] (b) P3D-A (c) P3D-B (d) P3D-C

**#P3D** Qiu, Z., Yao, T., & Mei, T.. Learning spatio-temporal representation with pseudo-3d residual networks. ICCV 2017.

# Pseudo-3D + Residual (P3D)

P3DNet is built by interleaving P3D-A, P3D-B and P3D-C blocks.



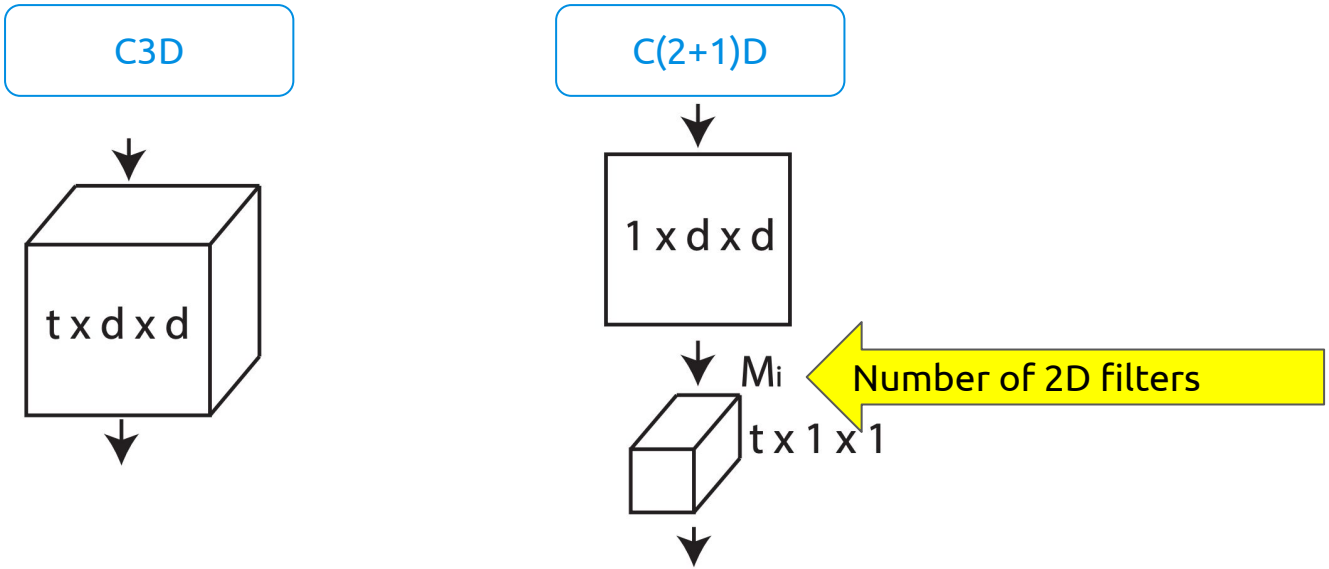

(a) P3D-A (b) P3D-B (c) P3D-C

**#P3D** Qiu, Z., Yao, T., & Mei, T.. Learning spatio-temporal representation with pseudo-3d residual networks. ICCV 2017.

# C(2+1)D

Splits the computation into a spatial 2D convolution followed by a temporal 1D convolution.

Example: In the case of a single input feature channel,

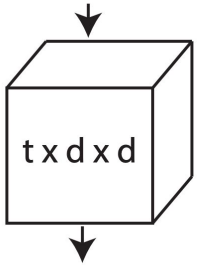



**#R(2+1)D** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# C(2+1)D

Compare the amount of parameters required in a layer of 32 convolutional filters with spatial size d=3 and temporal size t=3 for the C3D or C(2+1)D cases. Ignore biases and consider $M_i$=1.

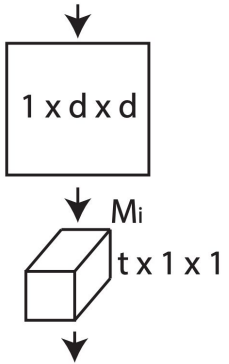
C3D




For one single C3D filter:
3 x 3 x 3 = 27 parameters

Considering M=32 filters:
32 x 27 = 864 parameters

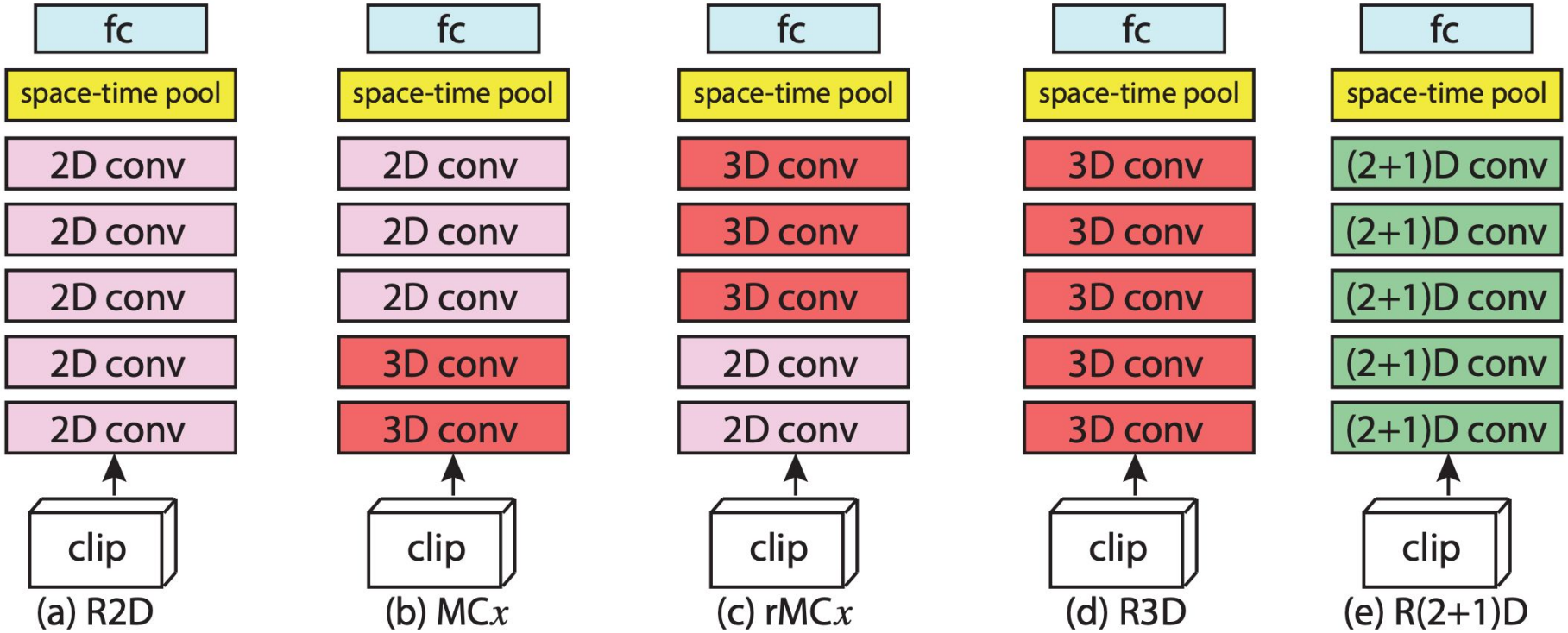
C(2+1)D




For one single C(2+1)D filter:
3 x 3 + 3 = 12 parameters

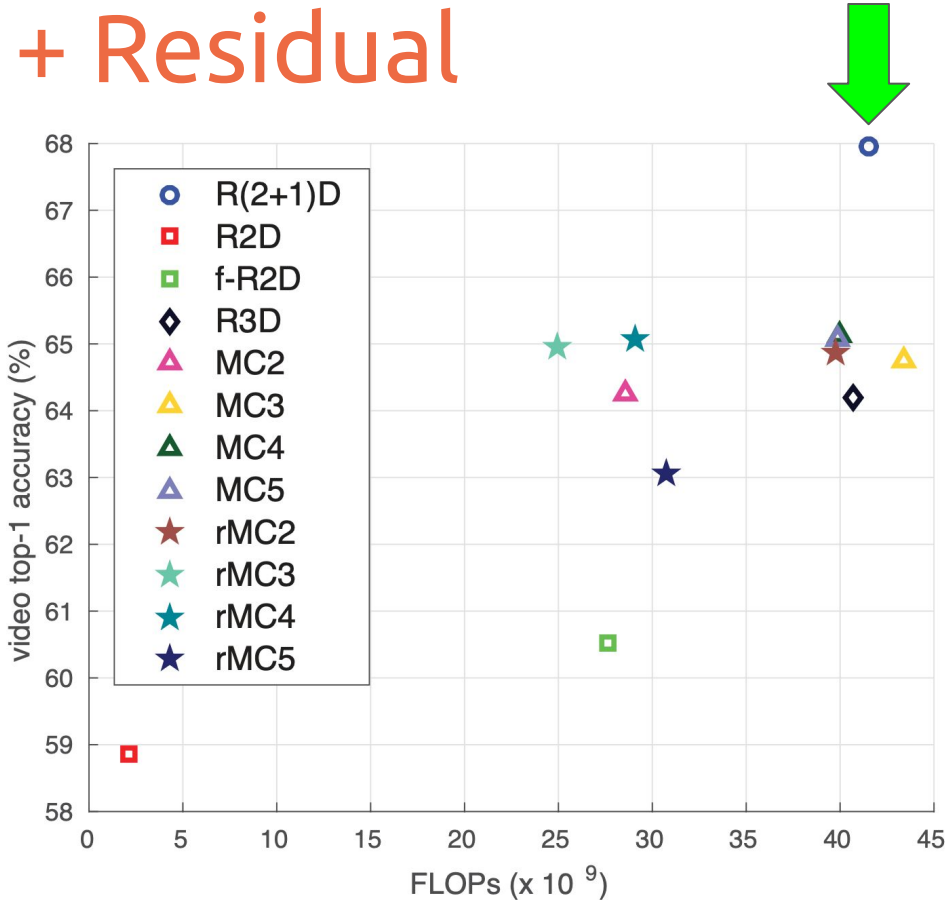
Considering M=32 filters:
32 x 12 = 384 parameters

# C(2+1D) + Residual




**#R(2+1)D** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

# 2+1D CNN + Residual



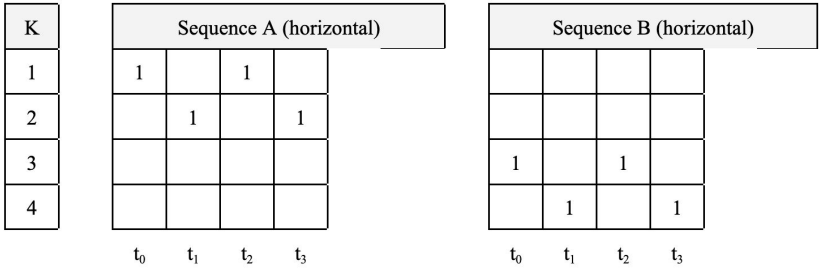

**#R(2+1)D** Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." CVPR 2018. [code]

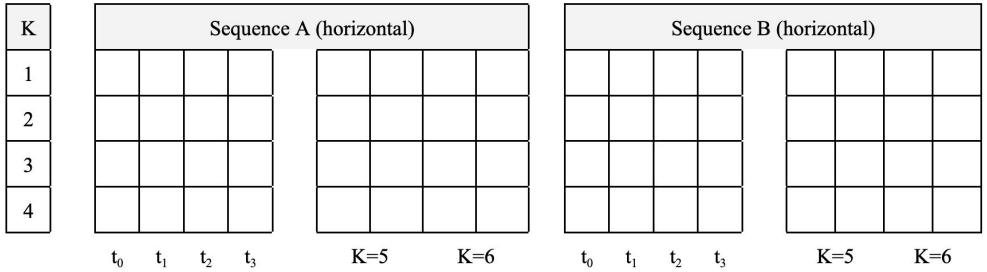
# Exercise: C(2+1)D

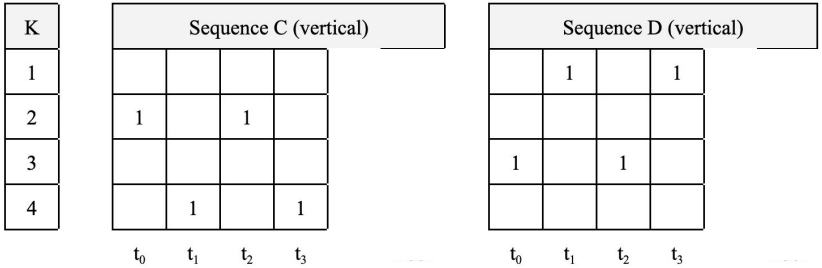
c) Instead of the temporal pooling, consider now the two 1D-temporal filters K={5,6} with stride 2, and apply them over the outputs, followed by another ReLU activation.

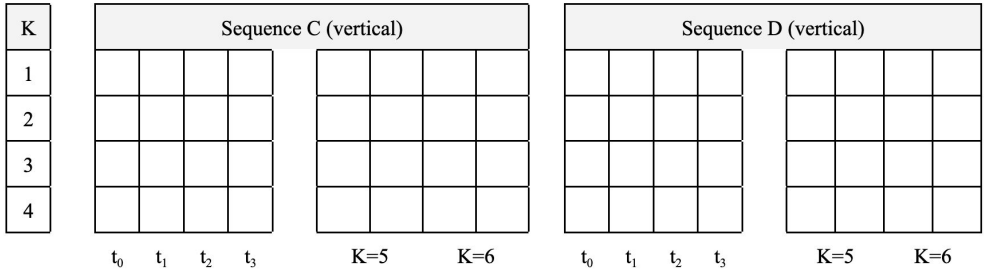
| K=5 | |
|---|---|
| 1 | |

| K=6 | |
|---|---|
| | 1 |



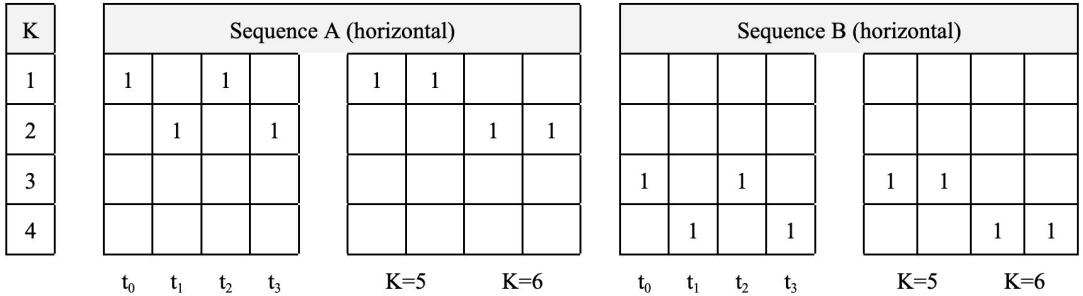



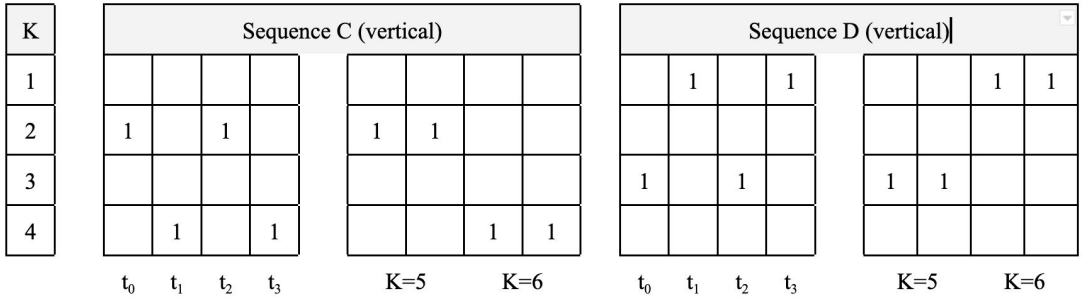
# Solution: C(2+1)D

c) Instead of the temporal pooling, consider now the two 1D-temporal filters K={5,6} with stride 2, and apply them over the outputs, followed by another ReLU activation.

| K=5 | |
|---|---|
| 1 | |

| K=6 | |
|---|---|
| | 1 |



# Separable (group) 2D convolutions (reminder)

Depth-wise (in channels) separable convolutions.

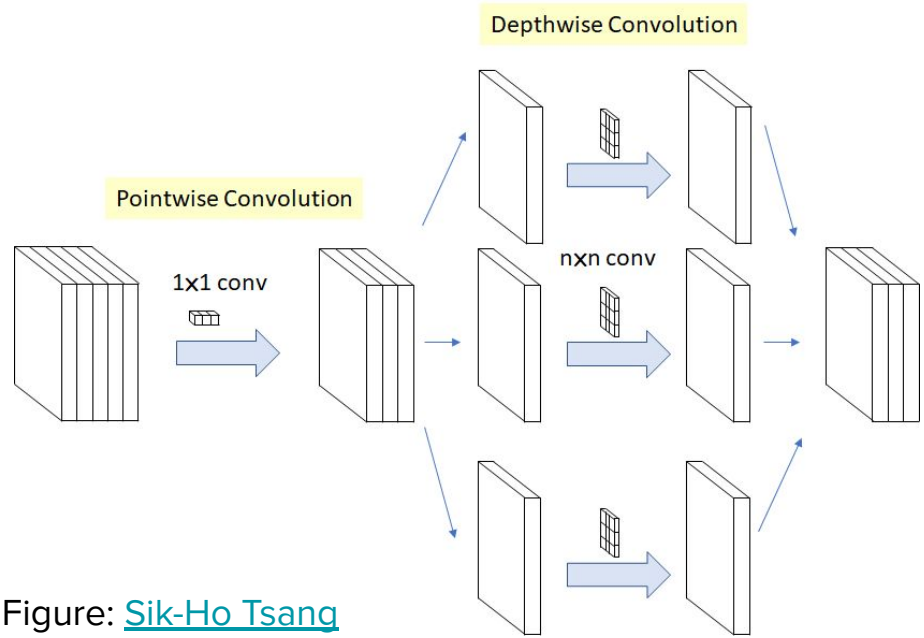
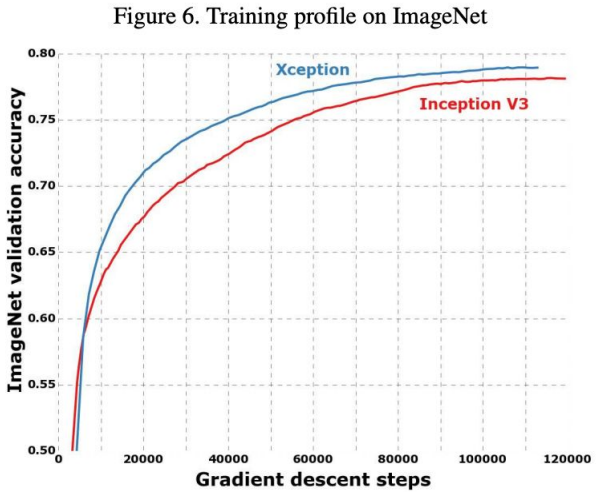



Figure: Sik-Ho Tsang






**#XCeption** Chollet, François. "Xception: Deep learning with depthwise separable convolutions." CVPR 2017.

# Inception + 3D Separable Convolutions

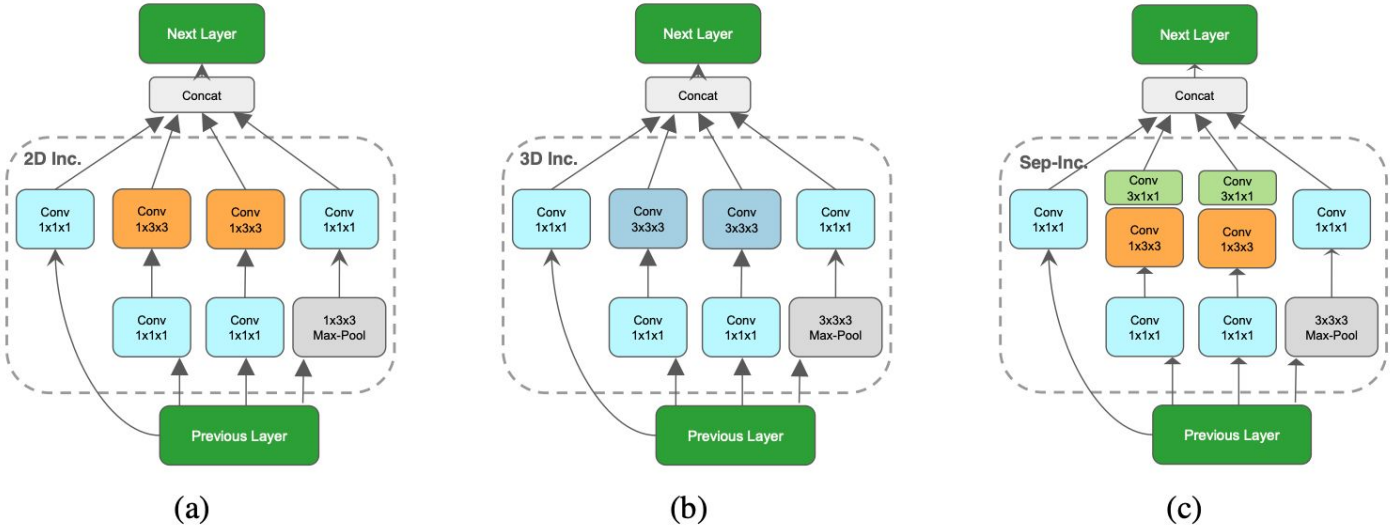
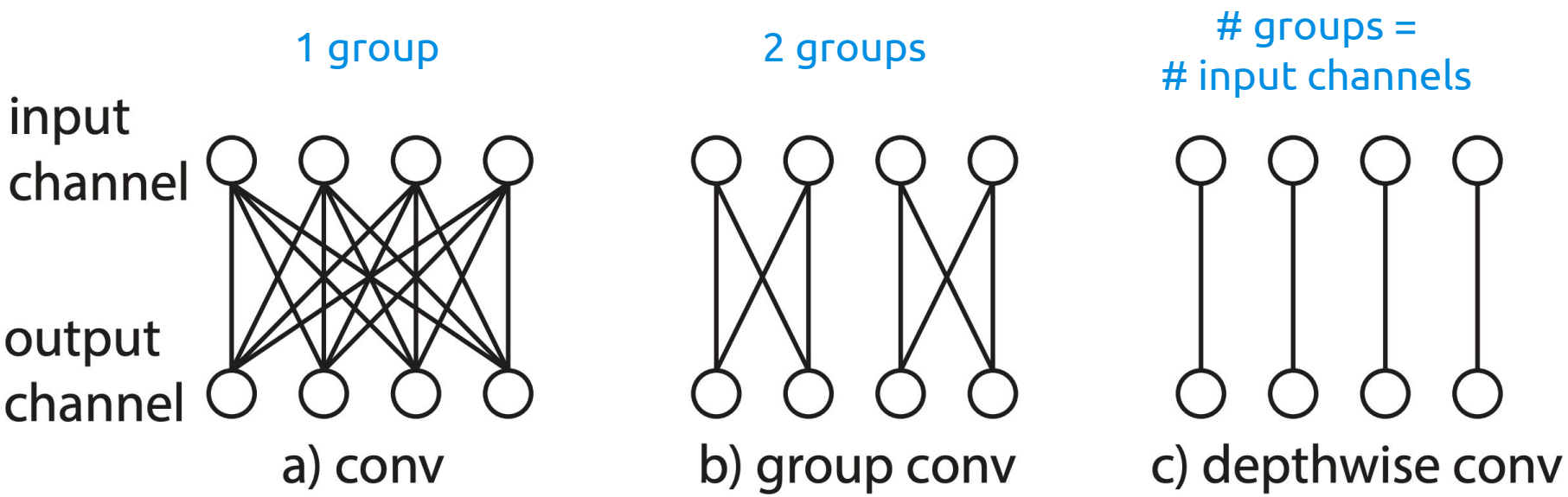



**Fig. 3.** (a) 2D Inception block; (b) 3D Inception block; (c) 3D temporal separable Inception block used in S3D networks.

**#S3D** Xie, S., Sun, C., Huang, J., Tu, Z., & Murphy, K.. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. ECCV 2018

# Res3D + Separable (group) convolutions

Channel interactions and spatiotemporal interactions are separated.




**#CSN** D. Tran, H. Wang, L. Torresani and M. Feiszli. Video Classification with Channel-Separated Convolutional Networks. ICCV 2019. [Caffe2]

# Res3D + Separable (group) convolutions

| Method | input | video@1 | video@5 | GFLOPs×crops |
|---|---|---|---|---|
| C3D [31] | RGB | 61.1 | 85.2 | N/A |
| P3D [25] | RGB | 66.4 | 87.4 | N/A |
| Conv Pool [41] | RGB+OF | 71.7 | 90.4 | N/A |
| R(2+1)D [32] | RGB | 73.0 | 91.5 | 152×N/A |
| R(2+1)D [32] | RGB+OF | 73.3 | 91.9 | 305×N/A |
| ir-CSN-101 | RGB | 74.8 | 92.6 | 56.5×10 |
| ip-CSN-101 | RGB | 74.9 | 92.6 | 63.6×10 |
| ir-CSN-152 | RGB | **75.5** | **92.7** | 74.0×10 |
| ip-CSN-152 | RGB | **75.5** | **92.8** | 83.3×10 |

**#CSN** D. Tran, H. Wang, L. Torresani and M. Feiszli. Video Classification with Channel-Separated Convolutional Networks. ICCV 2019. [Caffe2]

# Efficient spatio-temporal convolutions

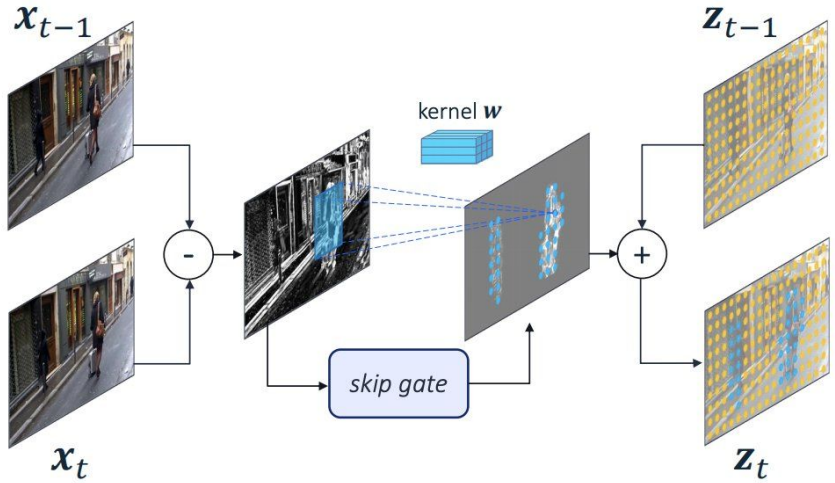



Figure 1: Skip-Convolution illustration for the input layer. Convolutions are computed only on a few locations in the residual features determined by a gate function (blue dots). In other locations, output features are copied from the previous time step (yellow dots). Frames taken from [66].

| | GMAC | Head | Sho. | Elb. | Wri. | Hip | Knee | Ank. | Avg |
|---|---|---|---|---|---|---|---|---|---|
| Park *et al.* [37] | - | 79.0 | 60.3 | 28.7 | 16.0 | 74.8 | 59.2 | 49.3 | 52.5 |
| Nie *et al.* [57] | - | 80.3 | 63.5 | 32.5 | 21.6 | 76.3 | 62.7 | 53.1 | 55.7 |
| Iqbal *et al.* [17] | - | 90.3 | 76.9 | 59.3 | 55.0 | 85.9 | 76.4 | 73.0 | 73.8 |
| Song *et al.* [45] | - | 97.1 | 95.7 | 87.5 | 81.6 | 98.0 | 92.7 | 89.8 | 92.1 |
| Luo *et al.* [31] | 70.98 | 98.2 | 96.5 | 89.6 | 86.0 | 98.7 | 95.6 | 90.9 | 93.6 |
| DKD *et al.* [34] | 8.65 | 98.3 | 96.6 | 90.4 | 87.1 | 99.1 | 96.0 | **92.9** | 94.0 |
| HRNet-w32 [52] | 10.19 | 98.5 | 97.3 | 91.8 | 87.6 | 98.4 | 95.4 | 90.7 | 94.5 |
| +S-SVD [18] | 5.04 | 97.9 | 96.9 | 90.6 | 87.3 | 98.7 | 95.3 | 91.1 | 94.3 |
| +W-SVD [68] | 5.08 | 97.9 | 96.3 | 87.2 | 82.8 | 98.1 | 93.2 | 88.8 | 92.4 |
| +L0 [30] | 4.57 | 97.1 | 95.5 | 86.5 | 81.7 | 98.5 | 92.9 | 88.6 | 92.1 |
| **+Skip-Conv** | 5.30 | **98.7** | **97.7** | **92.0** | **88.1** | **99.3** | **96.6** | 91.0 | **95.1** |

Table 1: Comparison with the state-of-the-art on JHMDB. Skip-Conv outperforms in PCK the best image and video models, whilst requiring fewer MAC per frame.

**#Skip-Conv** Amirhossein Habibian, Davide Abati, Taco S. Cohen, Babak Ehteshami Bejnordi, "Skip-Convolutions for Efficient Video Processing". CVPR 2021. [tweet]

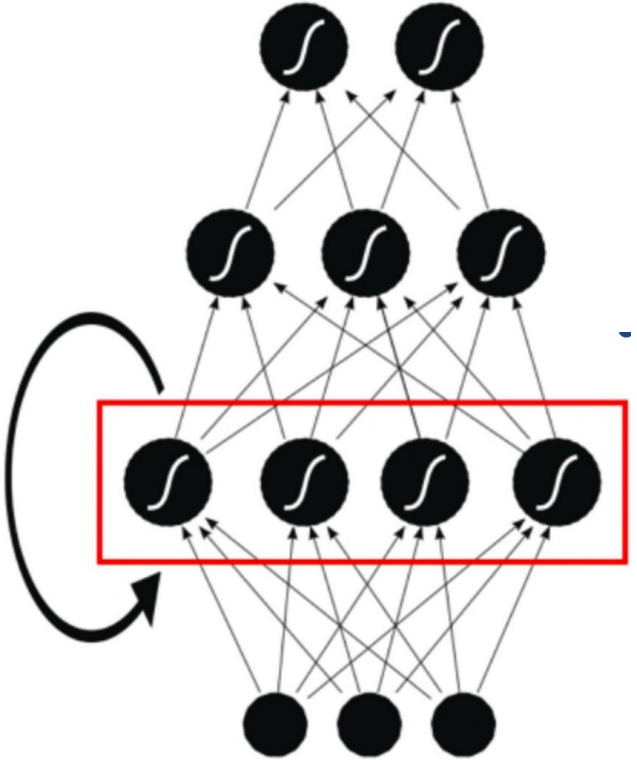
# Deep Video Architectures

Basic deep architectures for video:

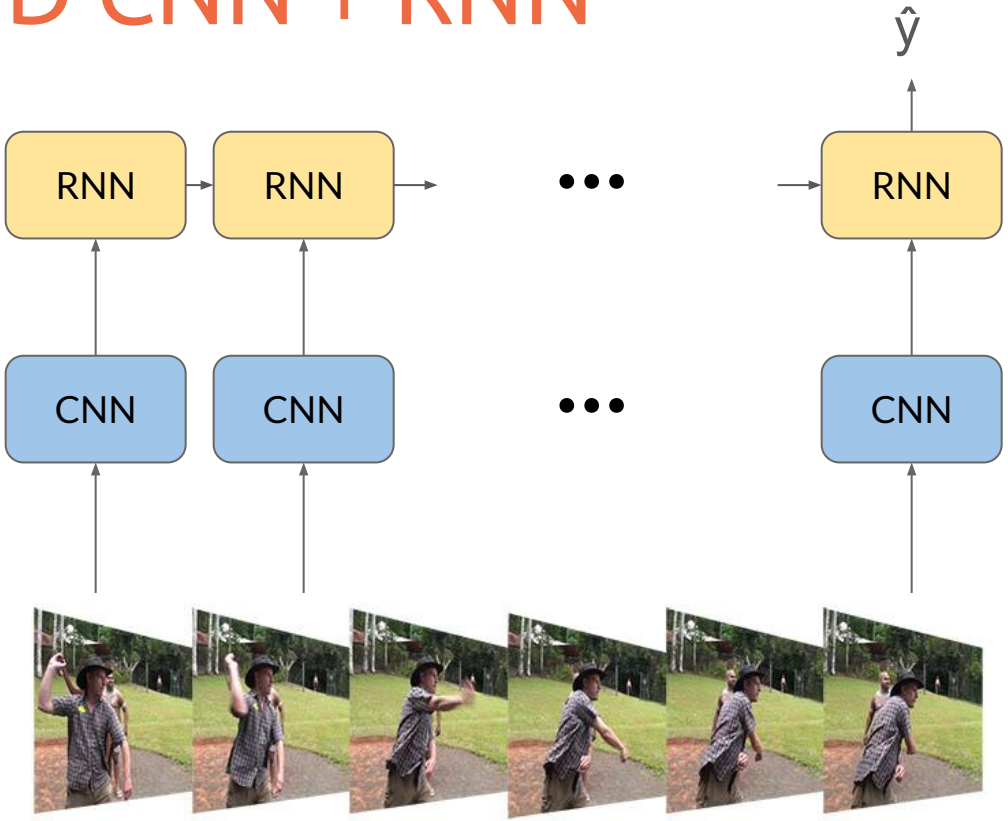
1. Single frame models
2. Spatio-temporal convolutions
3. **CNN + RNN**
4. RGB + Optical Flow
5. Transformers
6. Advanced solutions

# Recurrent Neural Network (RNN)


The hidden layers and the output depend from previous states of the hidden layers



# 2D CNN + RNN

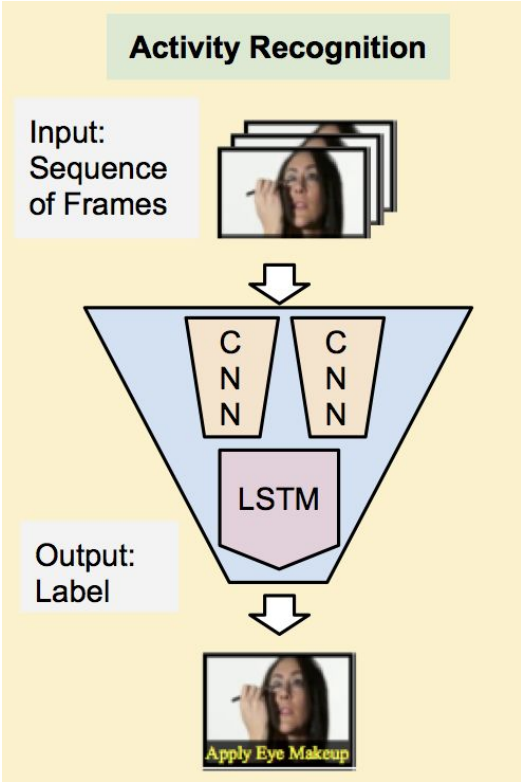



Recurrent Neural Networks are well suited for processing sequences.

Limitation:
RNNs are sequential and cannot be parallelized at training (and inference) time, while convolutions can.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrel. Long-term Recurrent Convolutional Networks for Visual Recognition and Description, CVPR 2015. code

# 2D CNN + RNN




Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrel. Long-term Recurrent Convolutional Networks for Visual Recognition and Description, CVPR 2015. [code]

# Exercise 1: 2D CNN + RNN

From now on, consider the output of the four C2D filters at timesteps $t_0$, $t_1$, $t_2$ and $t_3$ as a sequence of four tokens $e_0$, $e_1$, $e_2$ and $e_3$.

d) Instead of the 1D temporal convolutions in the previous section, consider now that the sequence of tokens is now fed into a single recurrent layer. If we want to keep the same amount of parameters (ignoring biases), how many neurons are there in this recurrent layer ? Develop your reasoning and calculations.

# Solution 1: 2D CNN + RNN

From now on, consider the output of the four C2D filters at timesteps $t_0$, $t_1$, $t_2$ and $t_3$ as a sequence of four tokens $e_0$, $e_1$, $e_2$ and $e_3$.

d) Instead of the 1D temporal convolutions used in the C(2+1)d case, consider now that the sequence of tokens is now fed into a single recurrent layer. If we want to keep the same amount of parameters (ignoring biases), how many neurons are there in this recurrent layer ? Develop your reasoning and calculations.

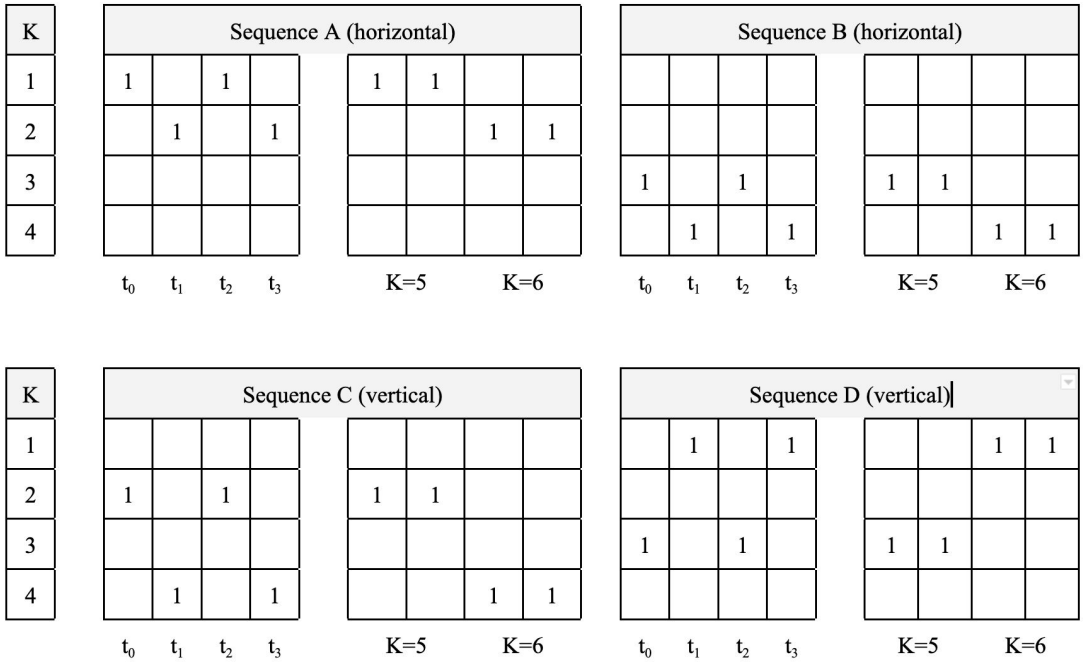
The 1D temporal convolutional layer is defined by two filters of size 2. So it contains 4 weights.

The video frame feature after the 2D convolutions is of size 4.

As a result, the recurrent layer can only be composed of a single FC recurrent neuron with 4 weights. Actually, the temporal recurrency over this single neuron will require an additional parameter .

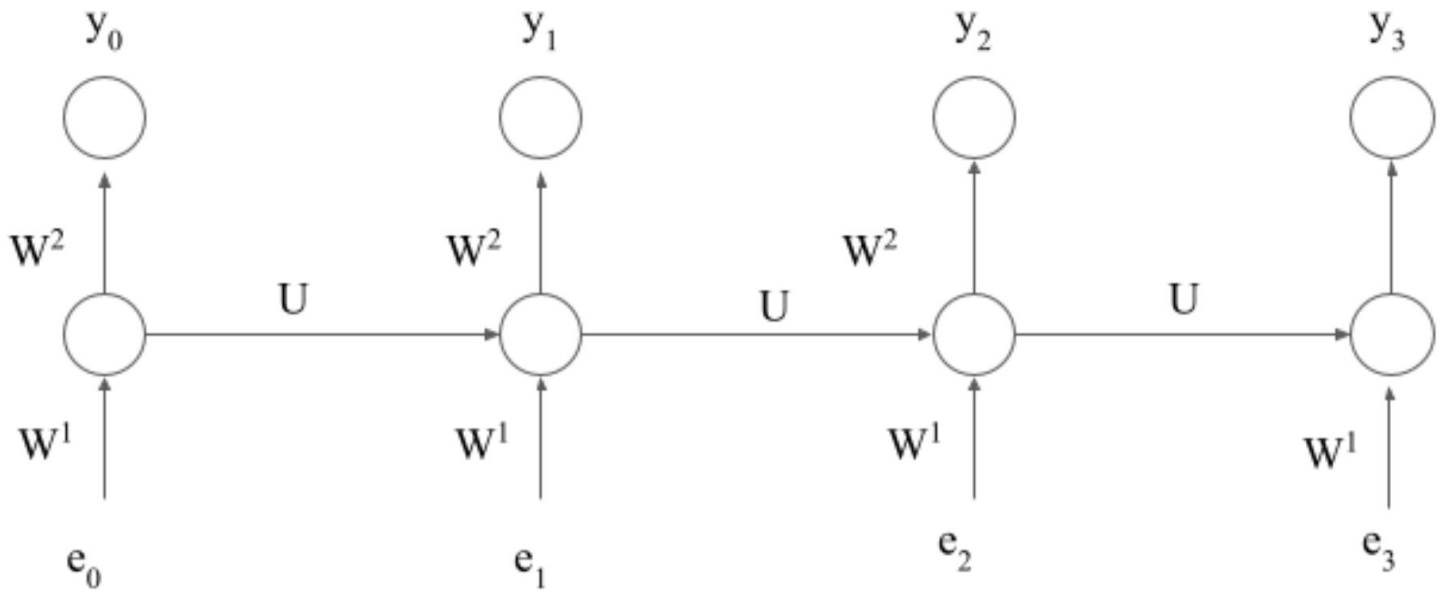
# Exercise 2: 2D CNN + RNN

Draw a temporally unfolded representation of the RNN architecture that depicts 4 timesteps over the sequence e0, e1, e2 and e3. Refer to the feedforward layers as W1 (recurrent layer) and W2 (output layer), and the weights of the recurrent layer as U. Ignore the biases in your figure.



# Solution 2: 2D CNN + RNN

Draw a temporally unfolded representation of the RNN architecture that depicts 4 timesteps over the sequence e0, e1, e2 and e3. Refer to the feedforward layers as W1 (recurrent layer) and W2 (output layer), and the weights of the recurrent layer as U. Ignore the biases in your figure.



# 2D CNN + RNN vs Spatio-temporal Convs

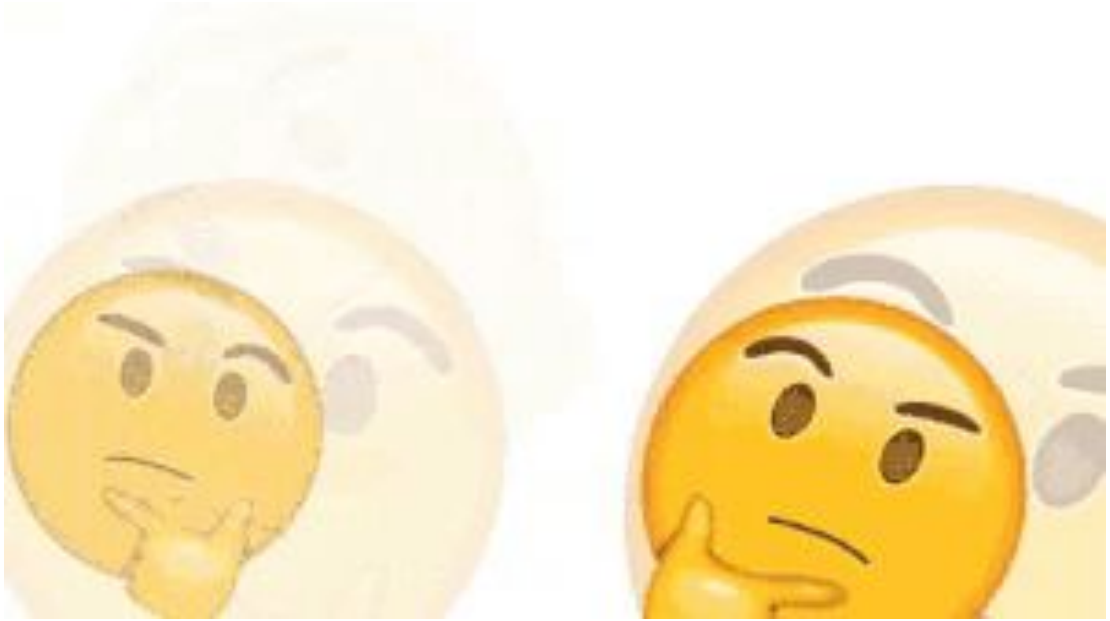
| Method | Clip@1 | Video@1 | Video@5 |
|---|---|---|---|
| single model, no long-term modeling | | | |
| DeepVideo [14] | 41.9 | 60.9 | 80.2 |
| C3D [41] | 46.1 | 61.1 | 85.2 |
| AlexNet [24] | N/A | 63.6 | 84.7 |
| GoogleNet [24] | N/A | 64.9 | 86.6 |
| 2D-Resnet* | 45.5 | 59.4 | 83.0 |
| **Res3D (ours)*** | **48.8** | **65.6** | **87.8** |
| with long-term modeling | | | |
| LSTM+AlexNet [24] | N/A | 62.7 | 83.6 |
| LSTM+GoogleNet [24] | N/A | 67.5 | 87.1 |

**#Res3D** Tran, Du, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. "Convnet architecture search for spatiotemporal feature learning." arXiv preprint arXiv:1708.05038 (2017). [code]

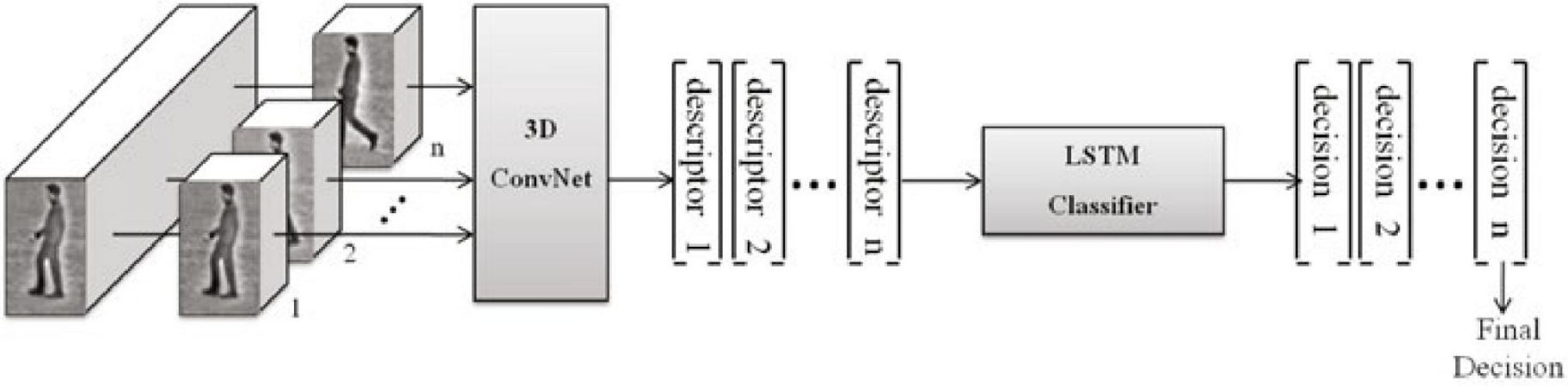
# 3D CNN (C3D)

Limitations of 16 clips inputs to C3D:

- How can we handle longer videos?
- How can we capture longer temporal dependencies?

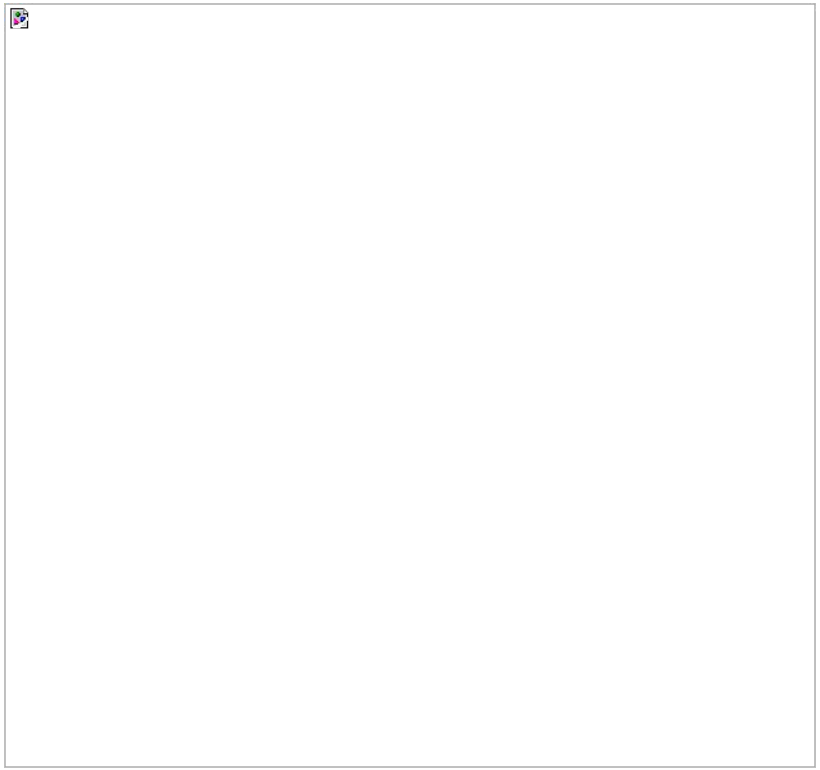


# 3D CNN + RNN




Baccouche, Moez, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. "Sequential deep learning for human action recognition." HBU 2011.

# 3D CNN + RNN

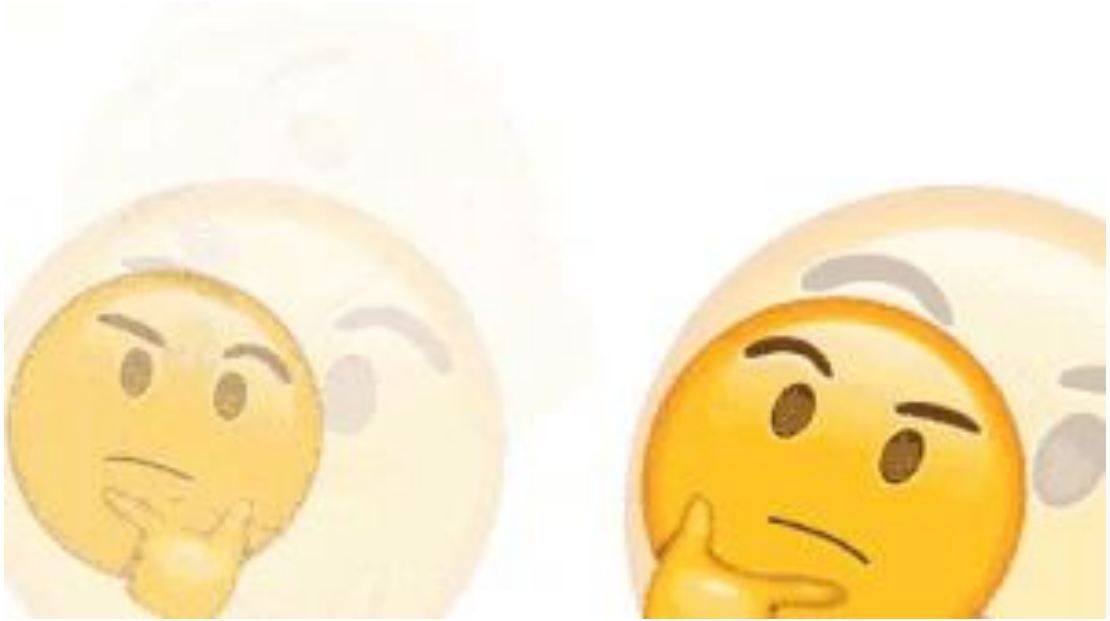



A. Montes, Salvador, A., Pascual-deLaPuente, S., and Giró-i-Nieto, X., "**Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks**", *NIPS Workshop 2016* **(best poster award)**

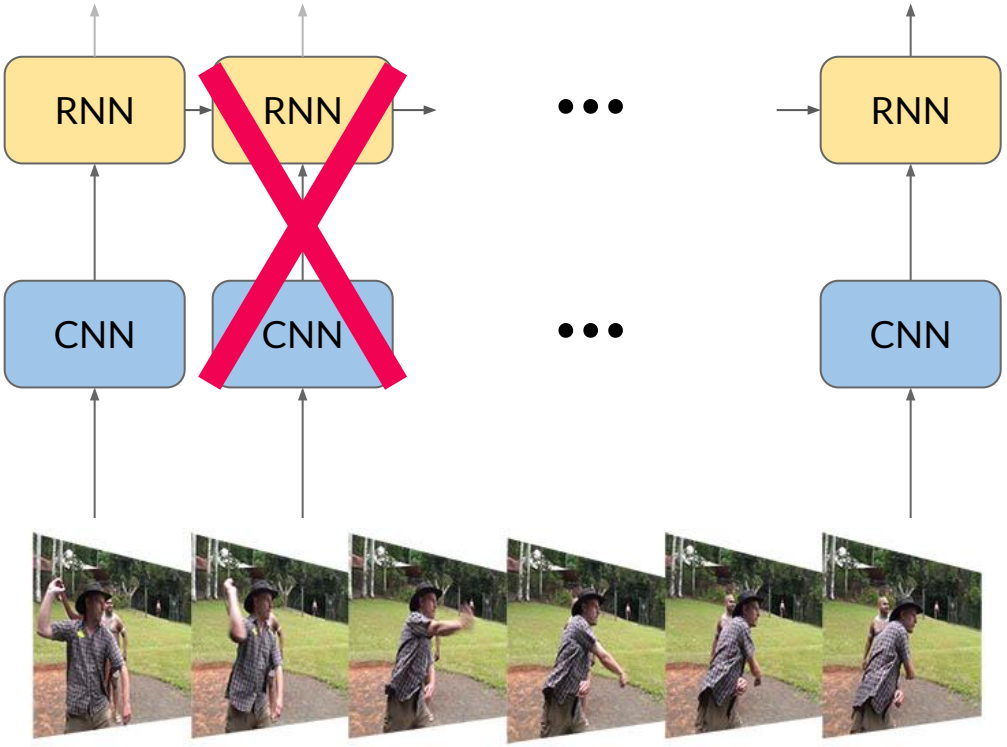
# 2D CNN + RNN

Limitations of RNNs for video:

- Do we really need to analyze 25/30 frames per second for the action recognition task ?



# 2D CNN + RNN: skip redundancy



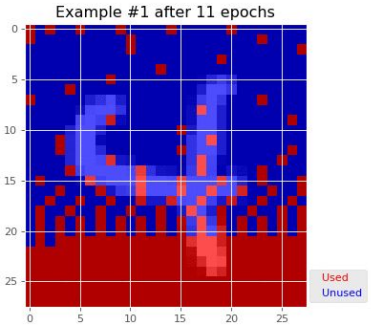

After processing a frame, let the RNN decide how many future frames can be skipped

In skipped frames, simply copy the output and state from the previous time step

There is no ground truth for which frames can be skipped. The RNN **learns** it by itself during training!

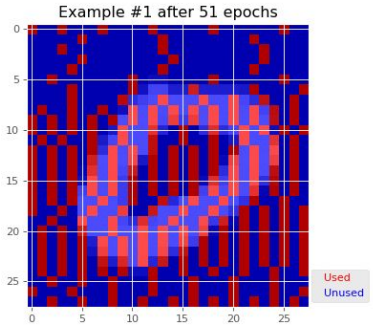
# 2D CNN + RNN

11 epochs



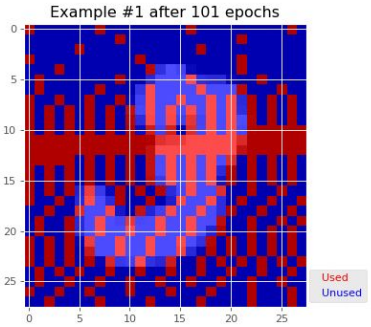

~30% acc

51 epochs



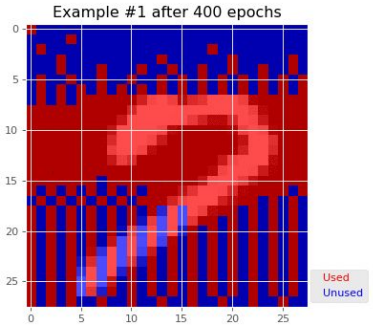

~50% acc

101 epochs




~70% acc

400 epochs



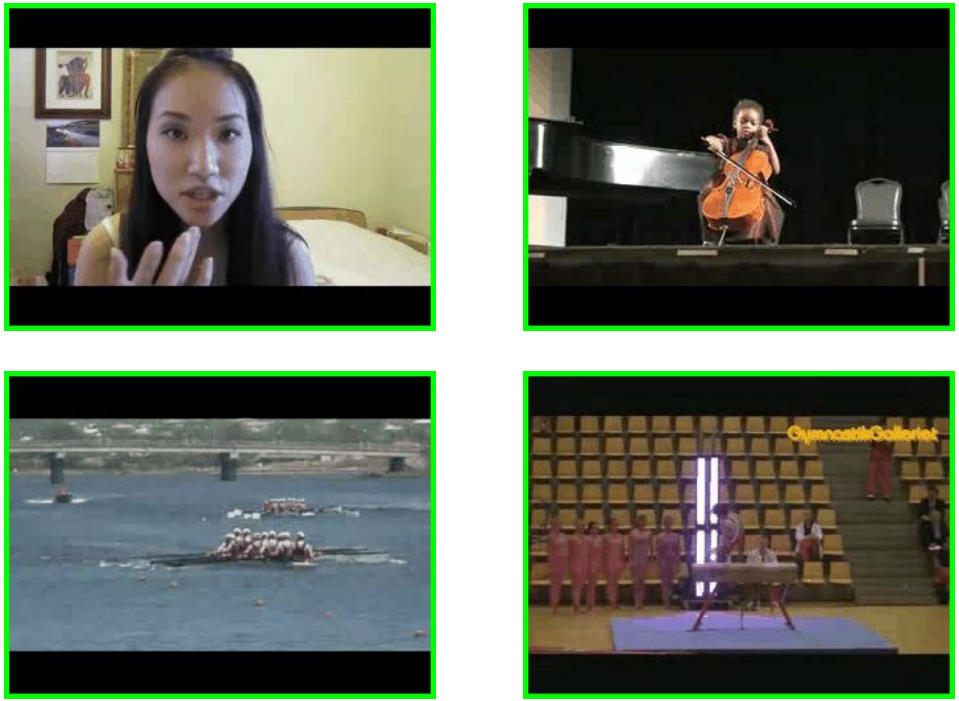

~95% acc

Epochs for Skip LSTM ($\lambda = 10^{-4}$)

**Used**
**Unused**

**#SkipRNN** Victor Campos, Brendan Jou, Xavier Giro-i-Nieto, Jordi Torres, and Shih-Fu Chang. "Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks", ICLR 2018.

# 2D CNN + RNN: skip redundancy



Used
Unused

# Deep Video Architectures

Basic deep architectures for video:

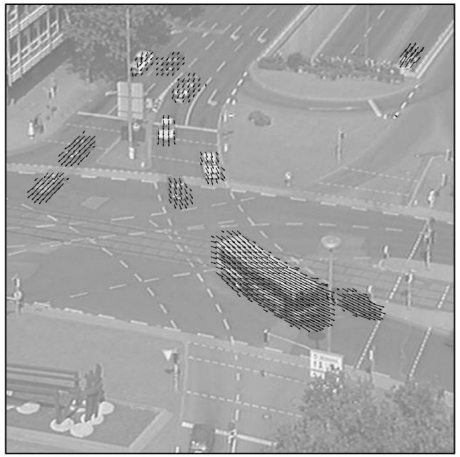
1. Single frame models
2. Spatio-temporal convolutions
3. CNN + RNN
4. **RGB + Optical Flow**
5. Transformers
6. Miscellaneous

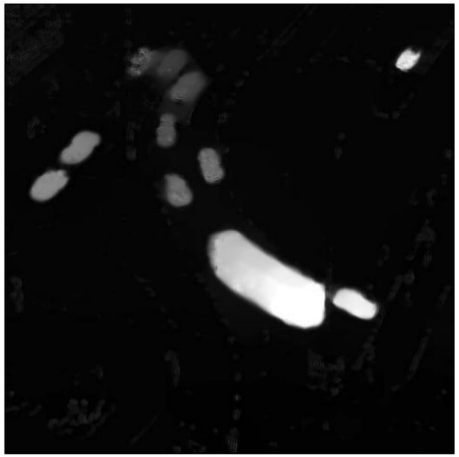

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. "Flownet 2.0: Evolution of optical flow estimation with deep networks." CVPR 2017. [code]

# Optical flow

Popular implementations for optical flow:

- PyFlow
- FlowNet2
- Improved Dense Trajectories - IDT
- Lucas-Kanade (OpenCV)
- Farneback 2003 (OpenCV)
- Middlebury
- Horn and schunk
- Tikhonov regularized and vectorized
- DeepFlow (2013)
- (...)

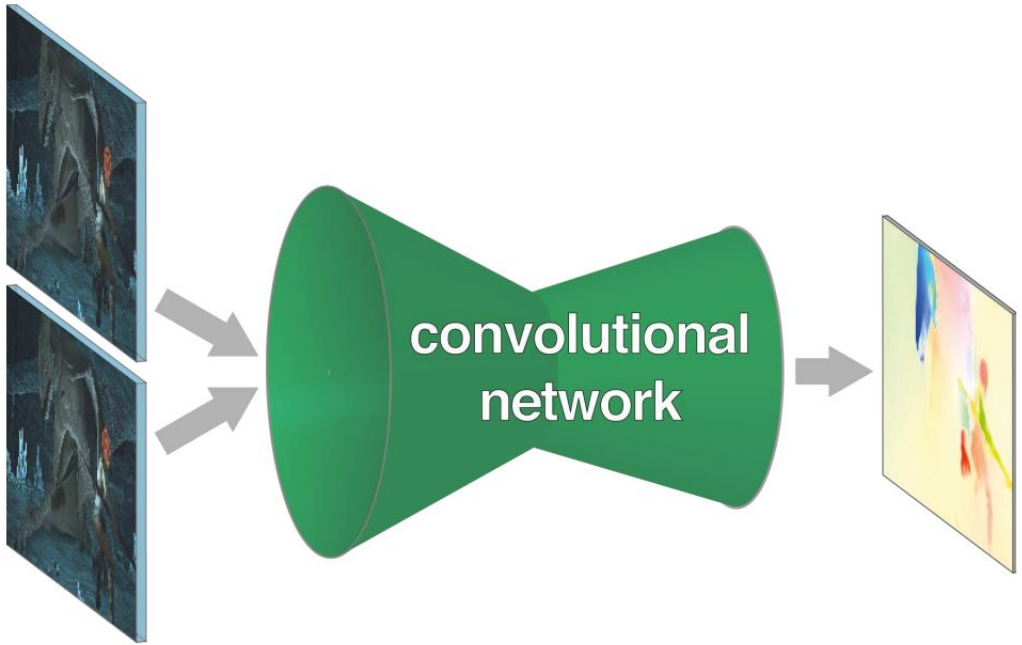




**#Coarse2Fine** Brox, Thomas, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. "High accuracy optical flow estimation based on a theory for warping." ECCV 2004.

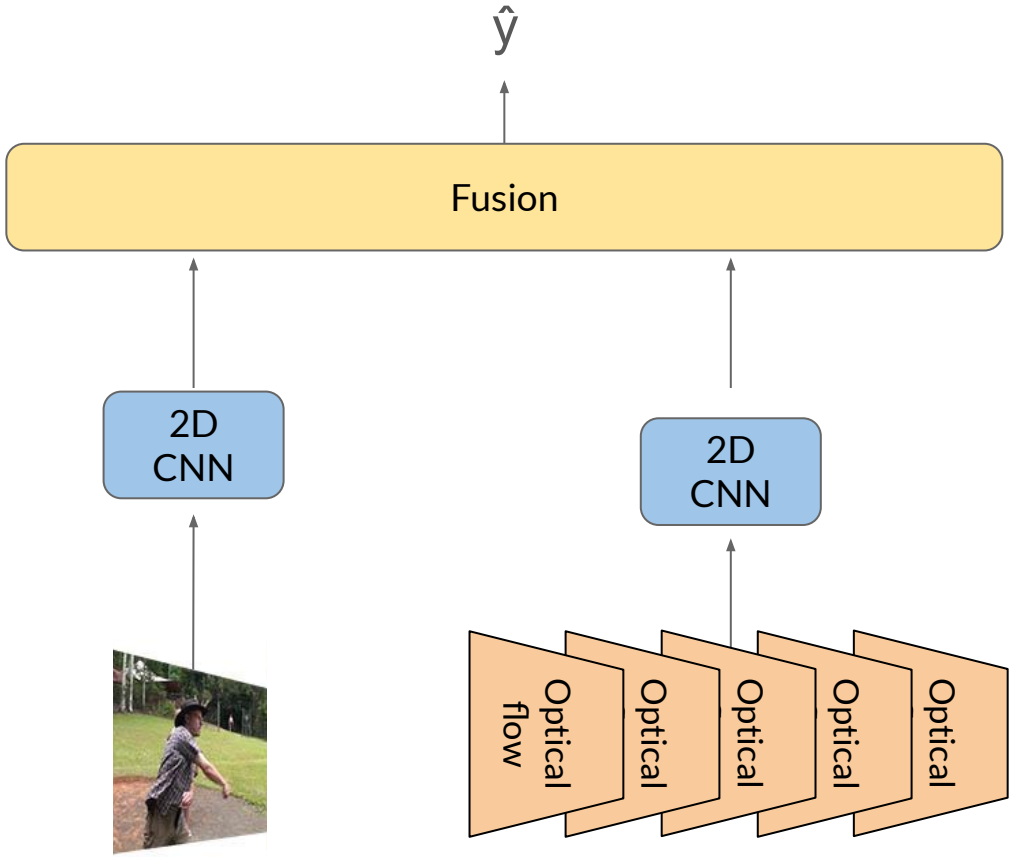
# Optical flow

Deep Neural Networks have actually also been trained to predict optical flow:



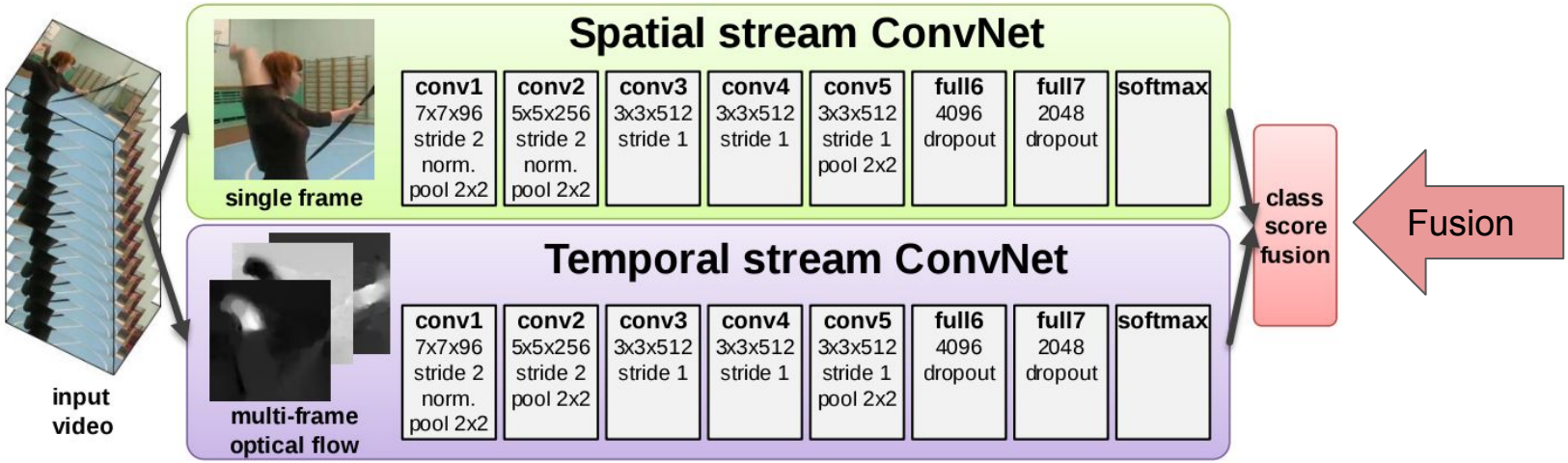

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

# Two-streams 2D CNNs



# Two-streams 2D CNNs

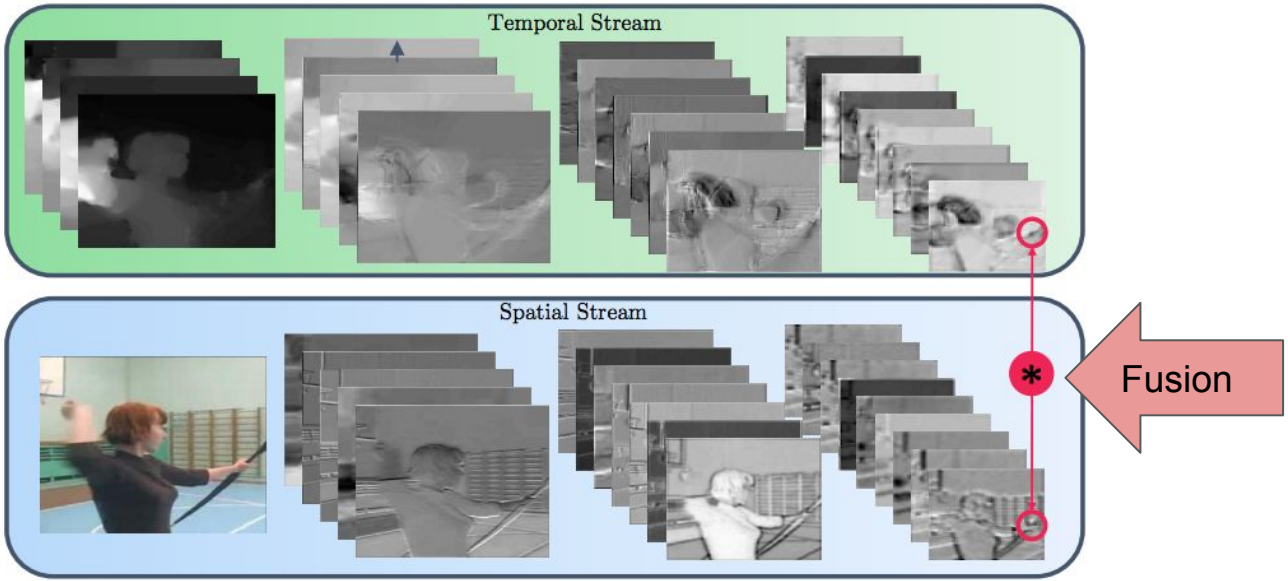
Problem: Single frame models do not take into account motion in videos.
Solution: extract **optical flow** for a stack of frames and use it as an input to a CNN.




Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." NIPS 2014.

# Two-streams 2D CNNs

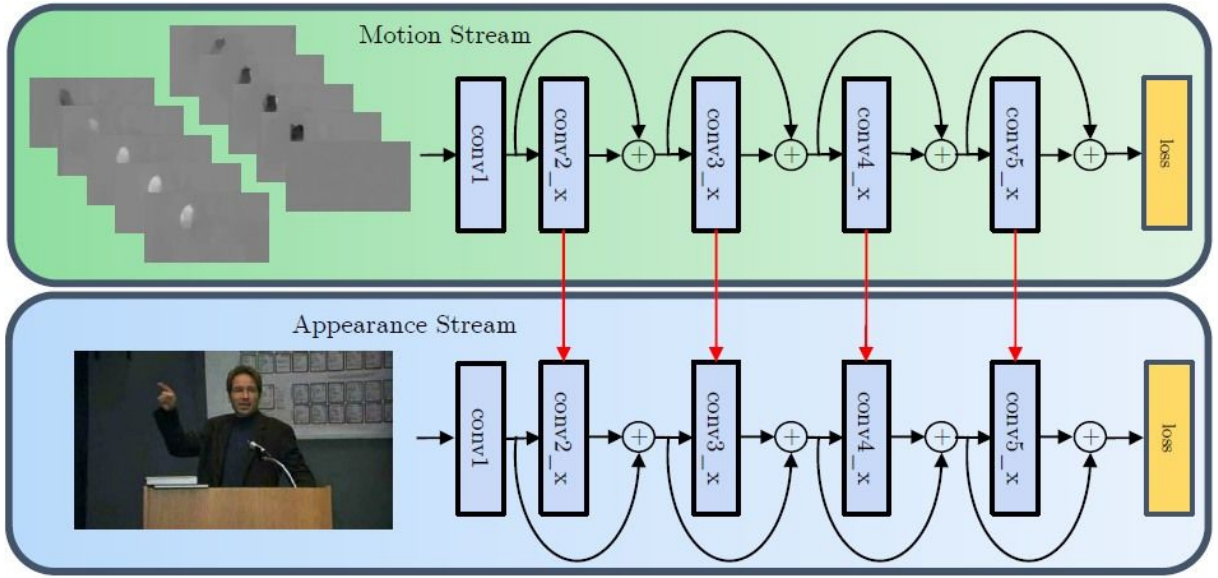
Problem: Single frame models do not take into account motion in videos.
Solution: Extract **optical flow** for a stack of frames and use it as an input to a CNN.




Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." CVPR 2016. [code]

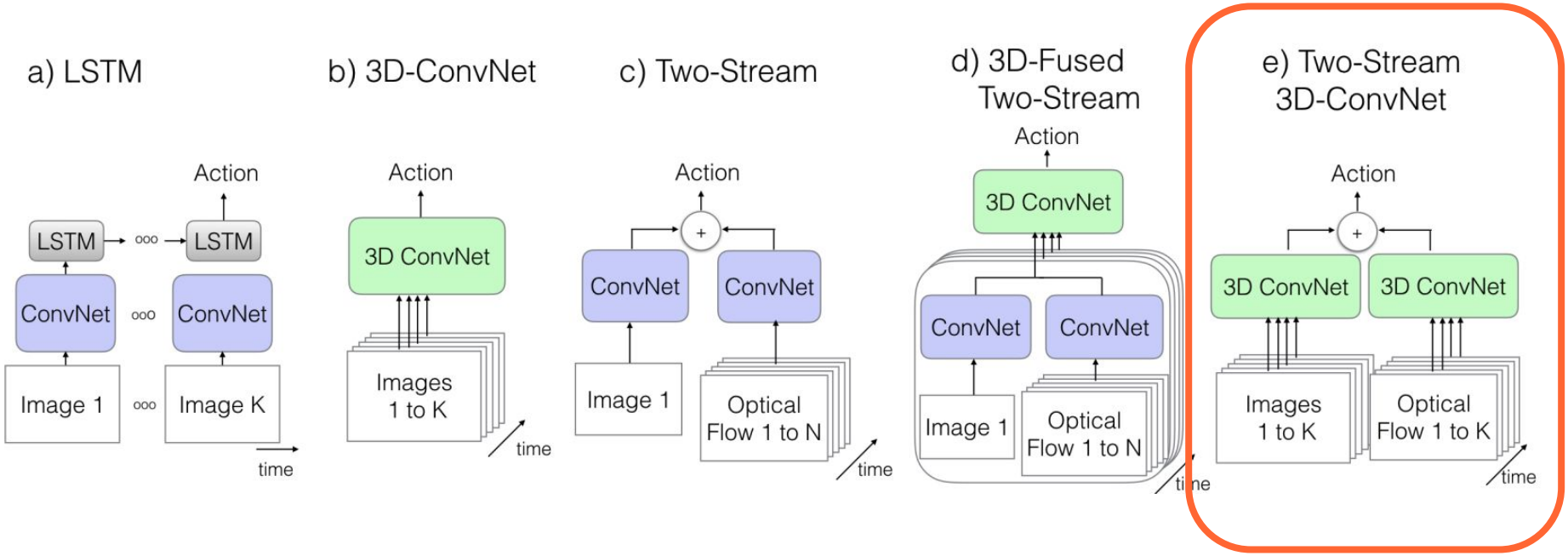
# Two-streams 2D CNNs + Residual

Problem: Single frame models do not take into account motion in videos.
Solution: Extract optical flow for a stack of frames and feed features in the appearance stream.



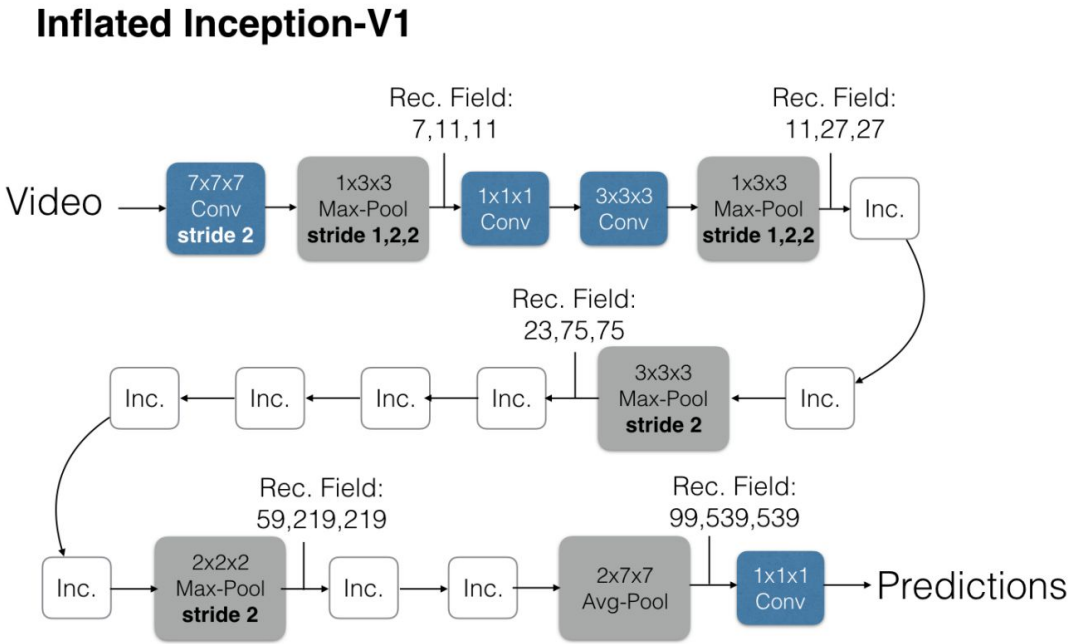

Feichtenhofer, Christoph, Axel Pinz, and Richard Wildes. "Spatiotemporal residual networks for video action recognition." NIPS 2016. [code]

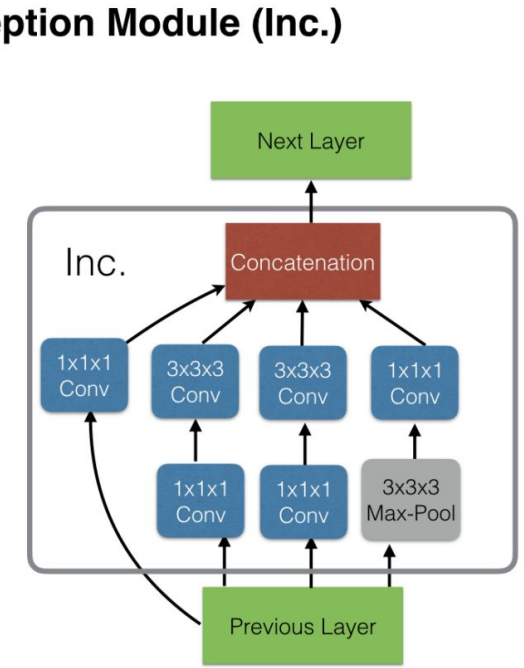
# Two-streams + Inflated 3D




**#I3D** Carreira, J., & Zisserman, A. . Quo vadis, action recognition? A new model and the kinetics dataset. CVPR 2017. [code]

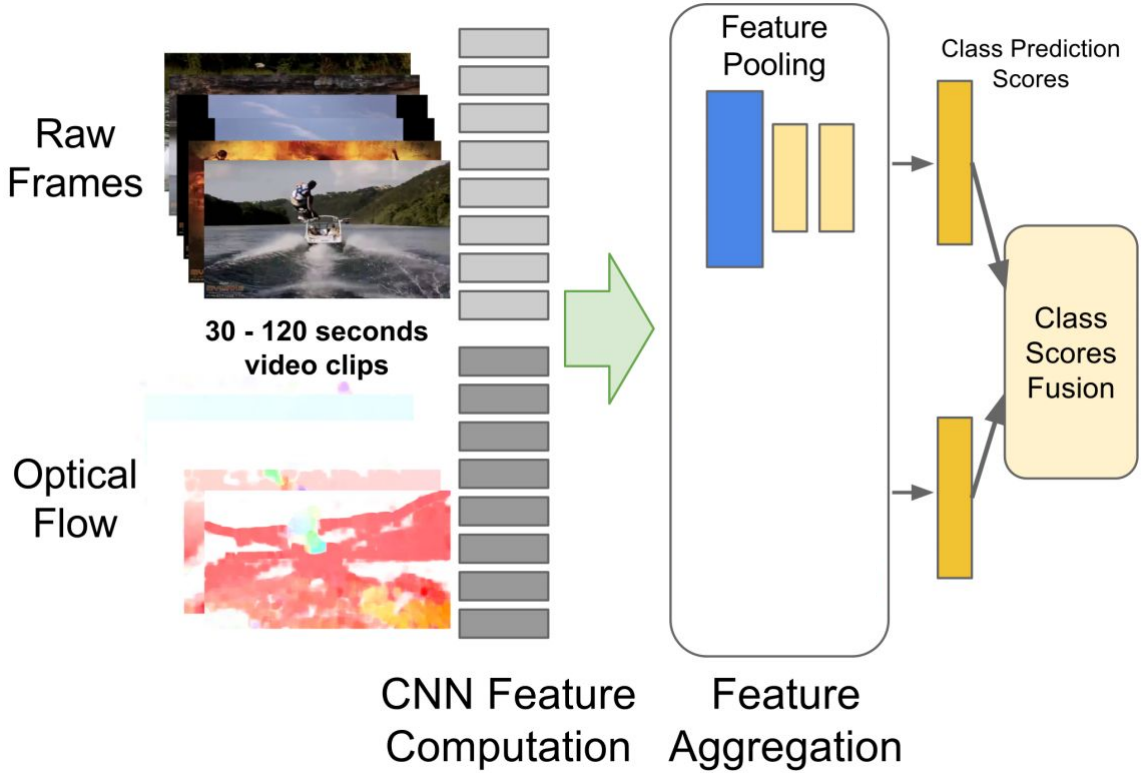
# Two-streams + Inflated 3D + Inception



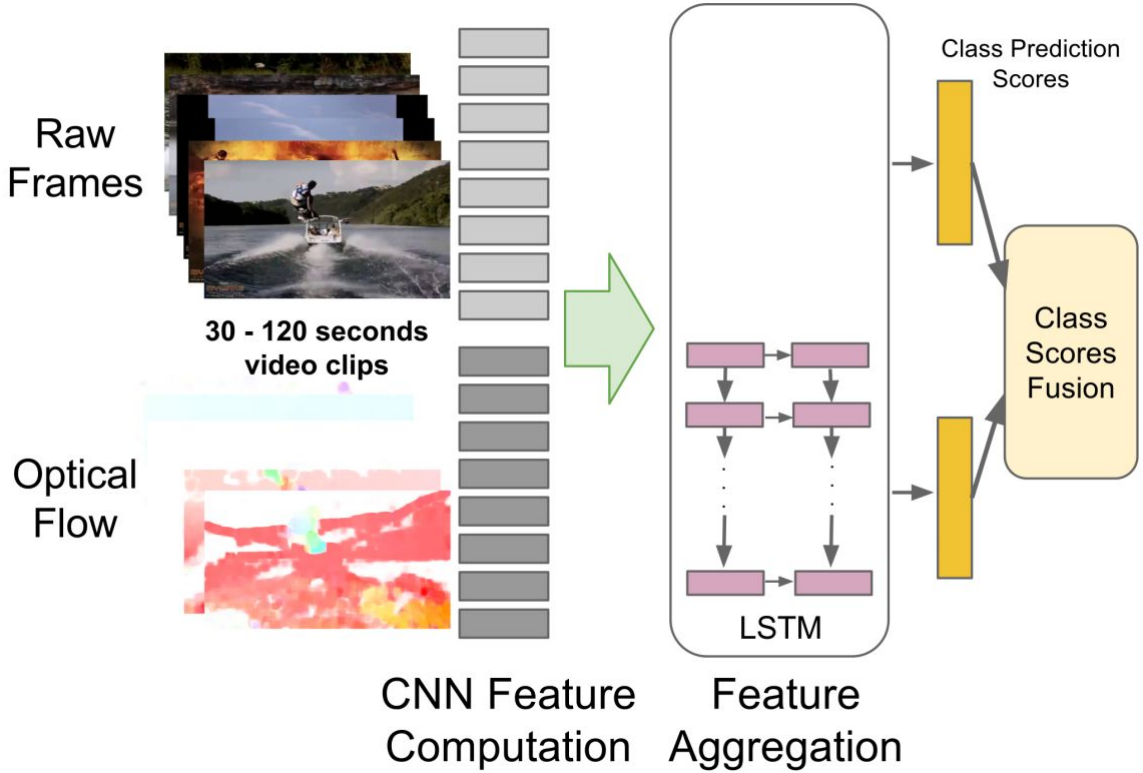

**#I3D** Carreira, J., & Zisserman, A. . Quo vadis, action recognition? A new model and the kinetics dataset. CVPR 2017. [code]

# Two-streams 2D CNNs + Time Pooling




**#ConvPool** Yue-Hei Ng, Joe, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." CVPR 2015

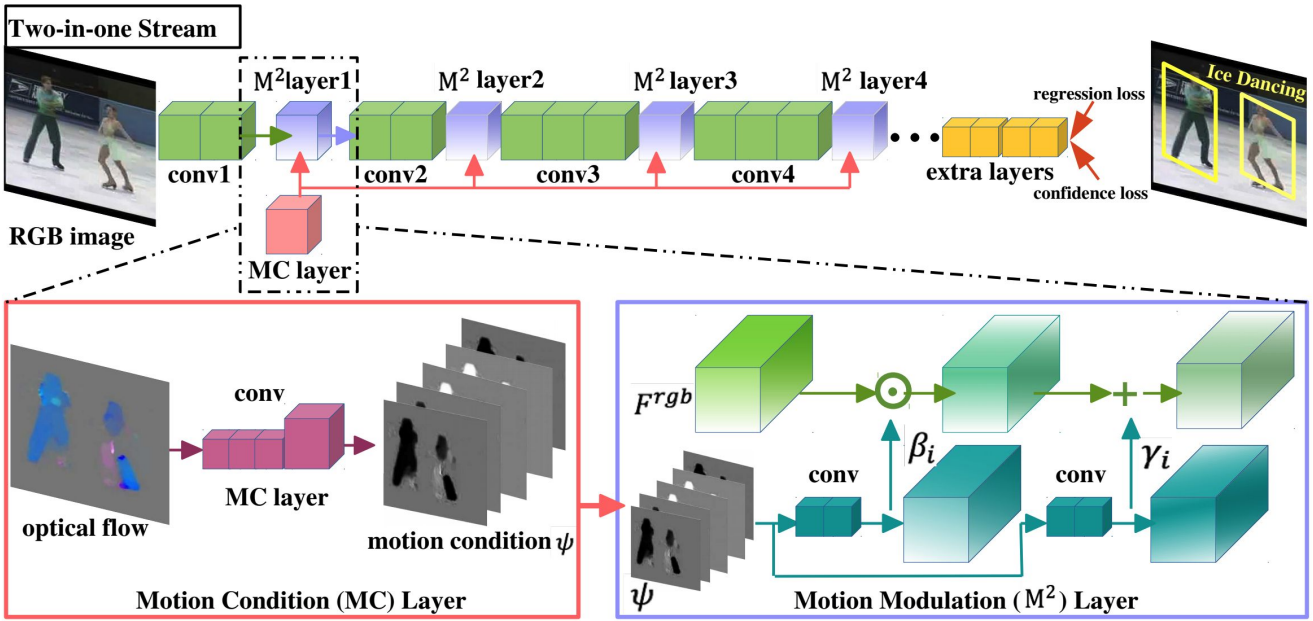
# Two-streams 2D CNNs + RNN




Yue-Hei Ng, Joe, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." CVPR 2015

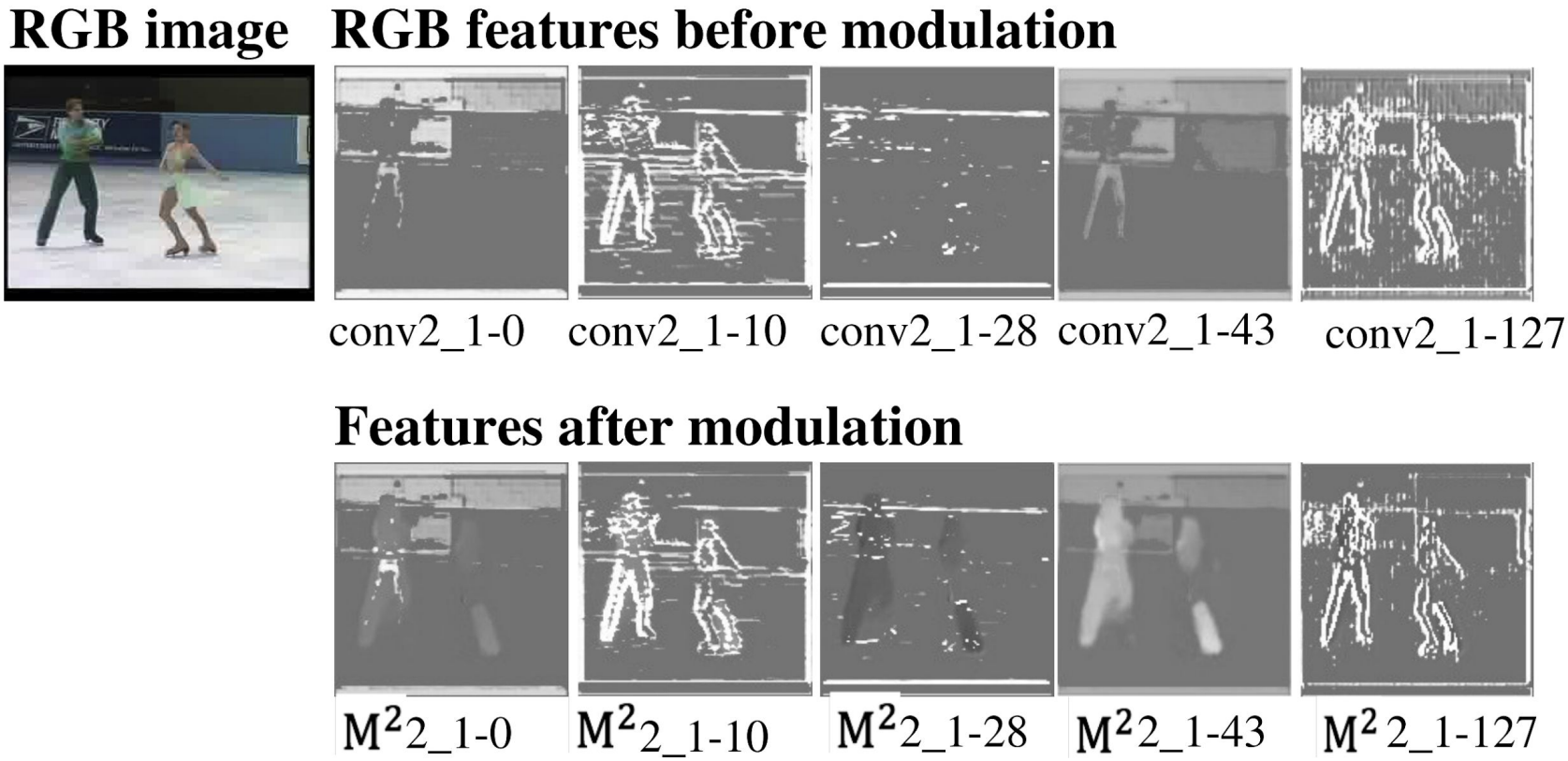
# One stream RGB + Optical Flow

Leveraging the motion condition to modulate RGB features improves detection accuracy.




**#2in1** Zhao, Jiaojiao, and Cees GM Snoek. "Dance with Flow: Two-in-One Stream Action Detection." CVPR 2019.

# One stream RGB + Optical Flow






**#2in1** Zhao, Jiaojiao, and Cees GM Snoek. "Dance with Flow: Two-in-One Stream Action Detection." CVPR 2019.

# One stream RGB + Optical Flow

2in1 has:

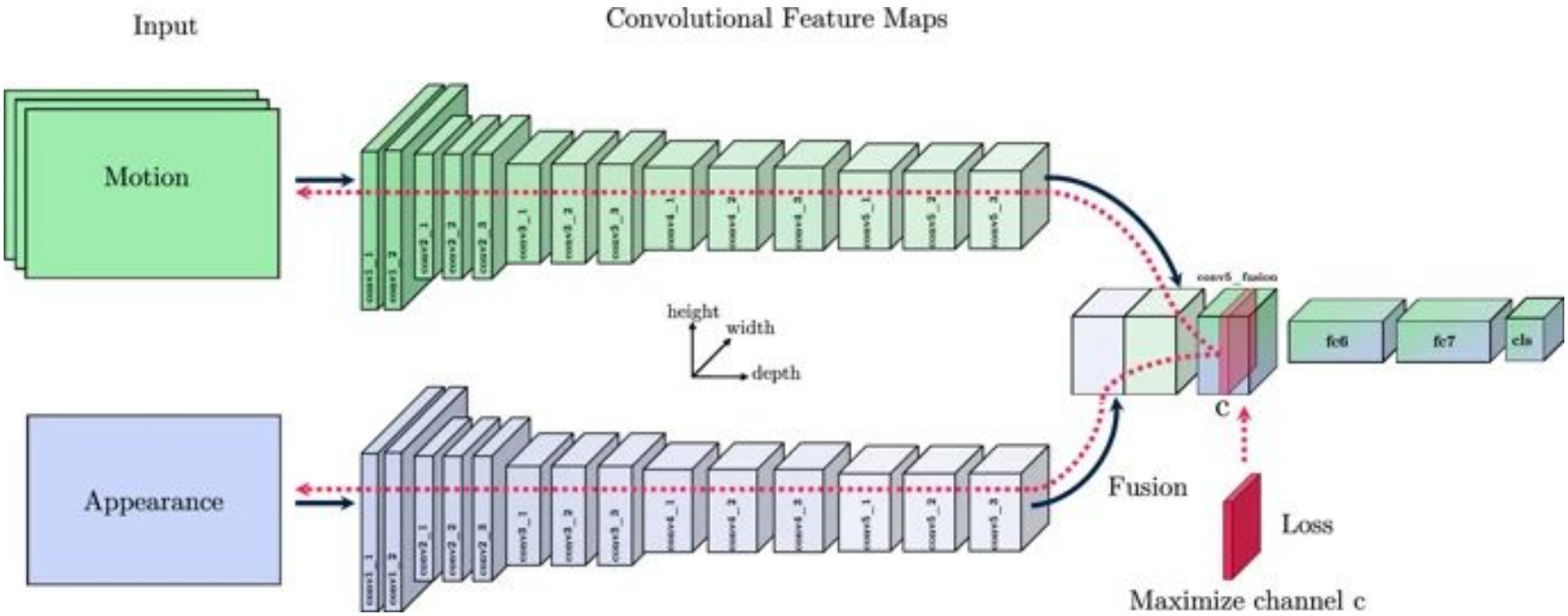
- half the computation and parameters of a two-stream equivalent
- better action detection accuracy (but worse for action classification).

| | Action Detection | | | Action Classification | | |
|---|---|---|---|---|---|---|
| Method | *mAP* | *Efficiency* | | *Top1 Accuracy* | *Efficiency* | |
| | % | sec/frame | # param. (M) | % | sec/frame | # param. (M) |
| flow-stream | 11.60 | 0.04 | 26.82 | 81.65 | 1.10 | 58.35 |
| RGB-stream | 18.49 | 0.04 | 26.82 | 84.99 | 1.10 | 58.35 |
| two-stream | 19.79 | 0.09 | 53.64 | 91.14 | 2.10 | 116.70 |
| two-in-one stream | 20.15 | 0.04 | 26.93 | 86.94 | 1.15 | 58.48 |
| two-in-one two stream | **22.02** | 0.09 | 53.75 | **92.00** | 2.13 | 116.83 |



**#2in1** Zhao, Jiaojiao, and Cees GM Snoek. "Dance with Flow: Two-in-One Stream Action Detection." CVPR 2019.

# Two-streams (feature visualization)



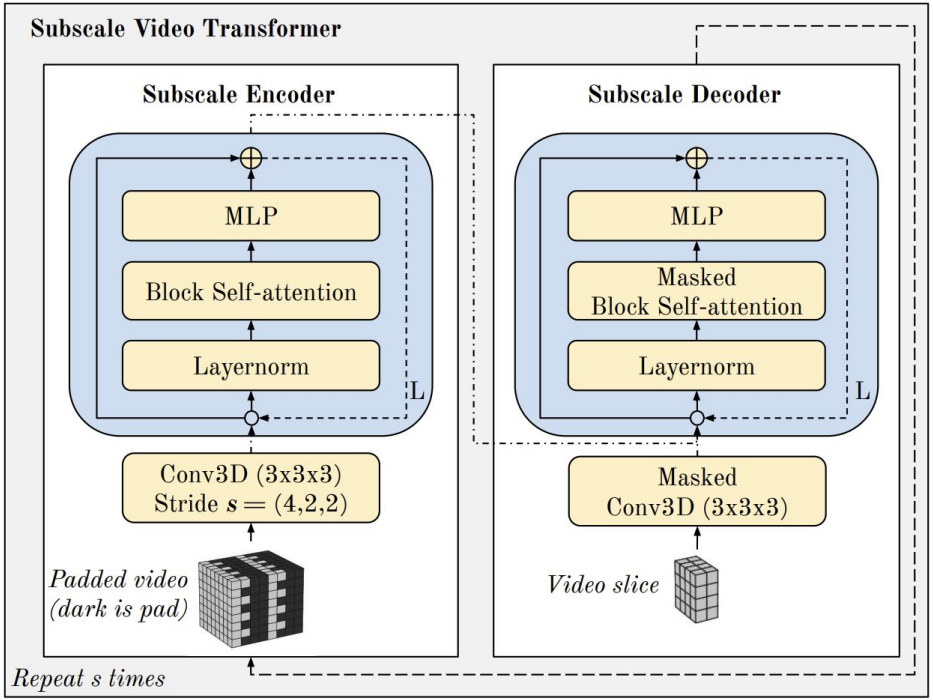

Feichtenhofer, Christoph, Axel Pinz, Richard P. Wildes, and Andrew Zisserman. "What have we learned from deep representations for action recognition?." CVPR 2018.
Feichtenhofer, Christoph, Axel Pinz, Richard P. Wildes, and Andrew Zisserman. "Deep Insights into Convolutional Networks for Video Recognition." IJCV 2019.

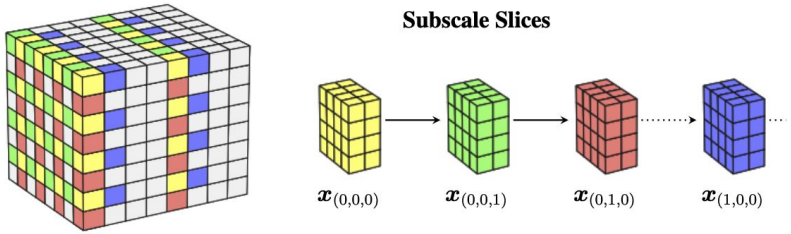
# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. RGB + Optical Flow
5. **Transformers**
6. Miscellaneous

# The Transformer for Vision

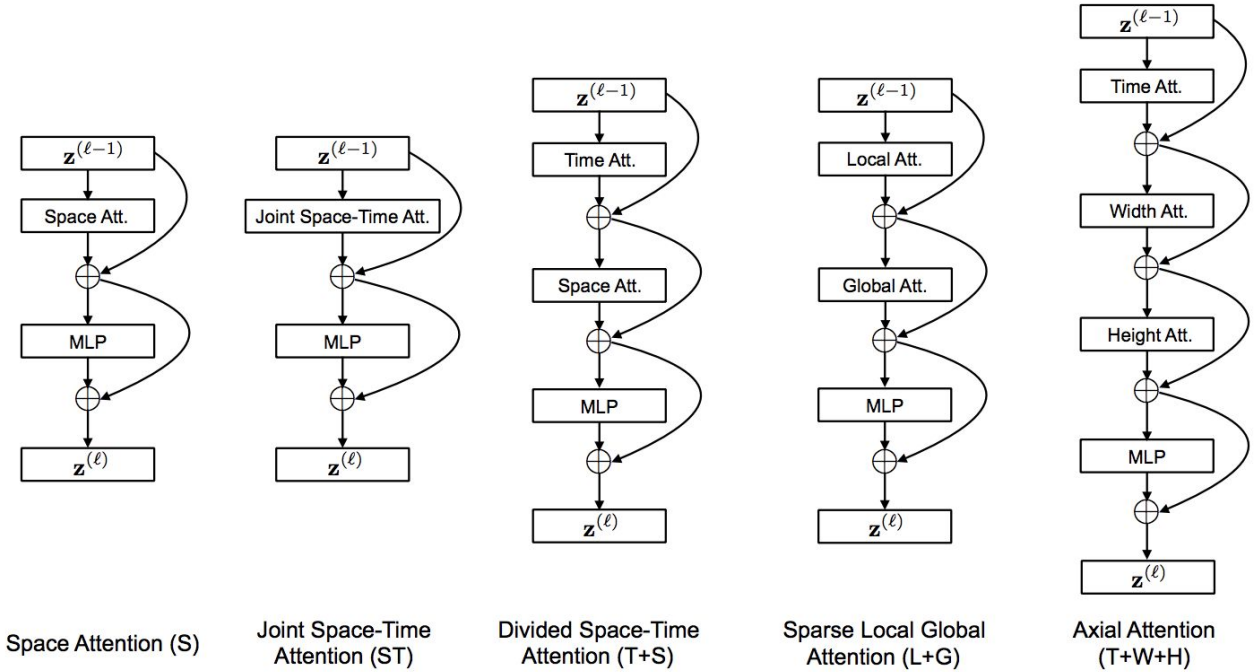



**#VideoTransformer** Weissenborn, Dirk, Oscar Täckström, and Jakob Uszkoreit. "Scaling autoregressive video models." ICLR 2020

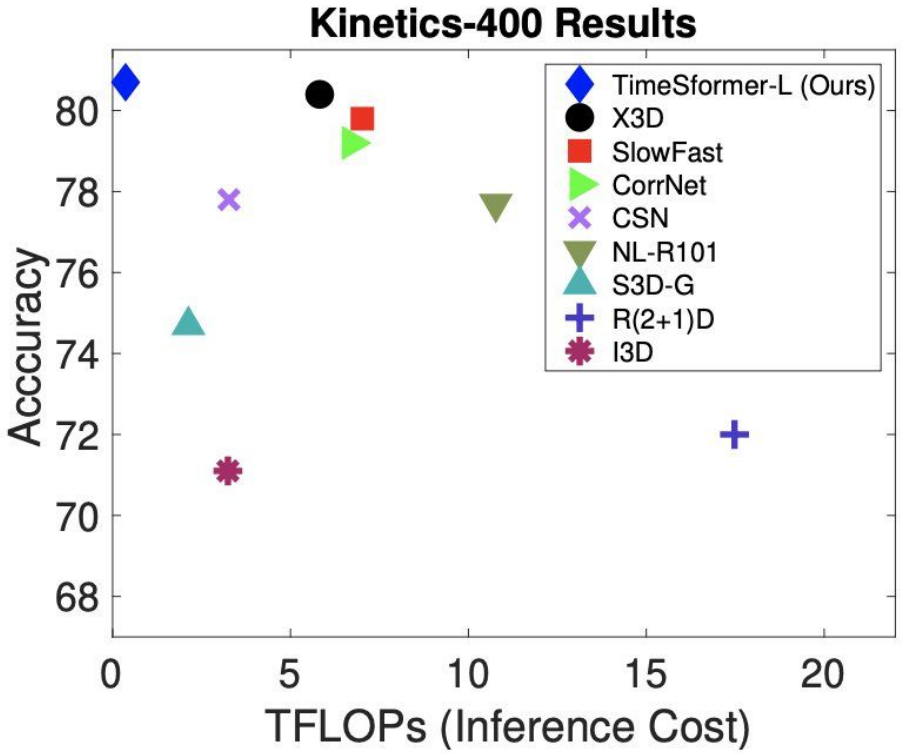
# The Transformer for Vision: Video

Attention along the time axis, before the spatial axis.



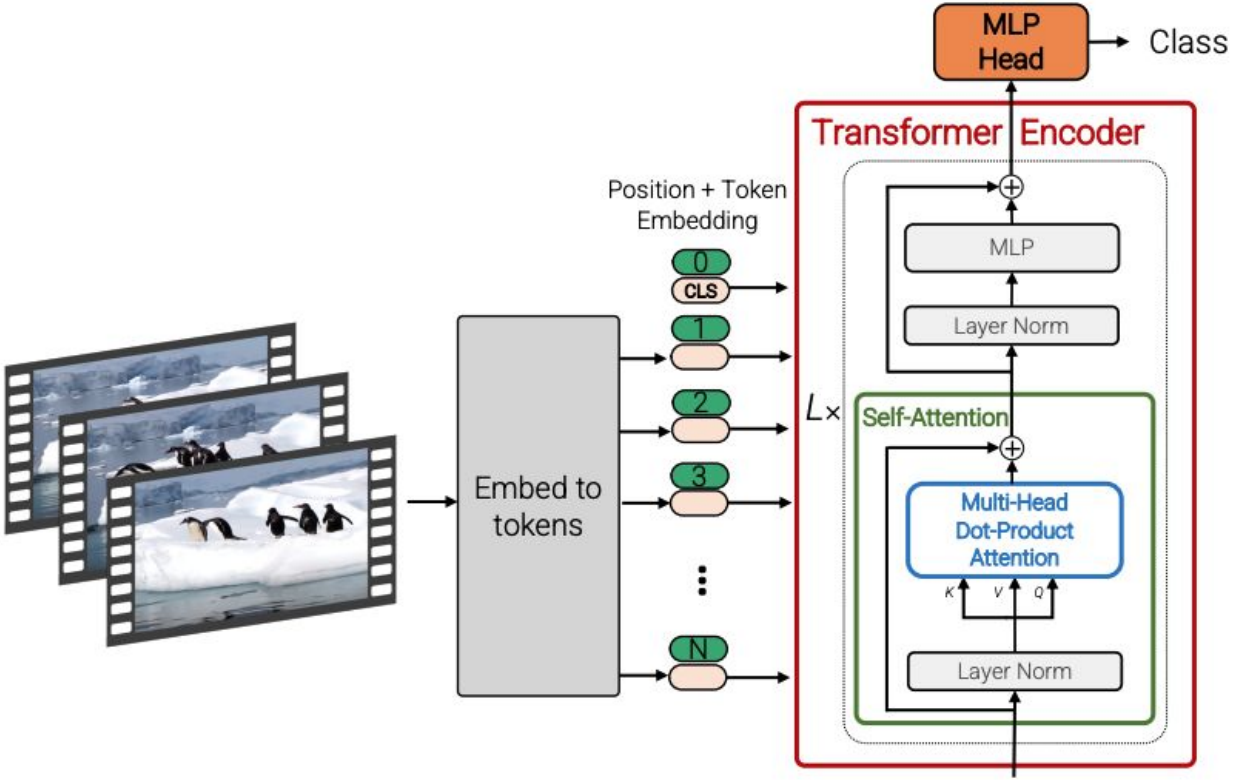

**#TimeSformer** Bertasius, G., Wang, H., & Torresani, L. "Is Space-Time Attention All You Need for Video Understanding?". ICML 2021. [tweet] [official code] [unofficial code]

# The Transformer for Vision: Video



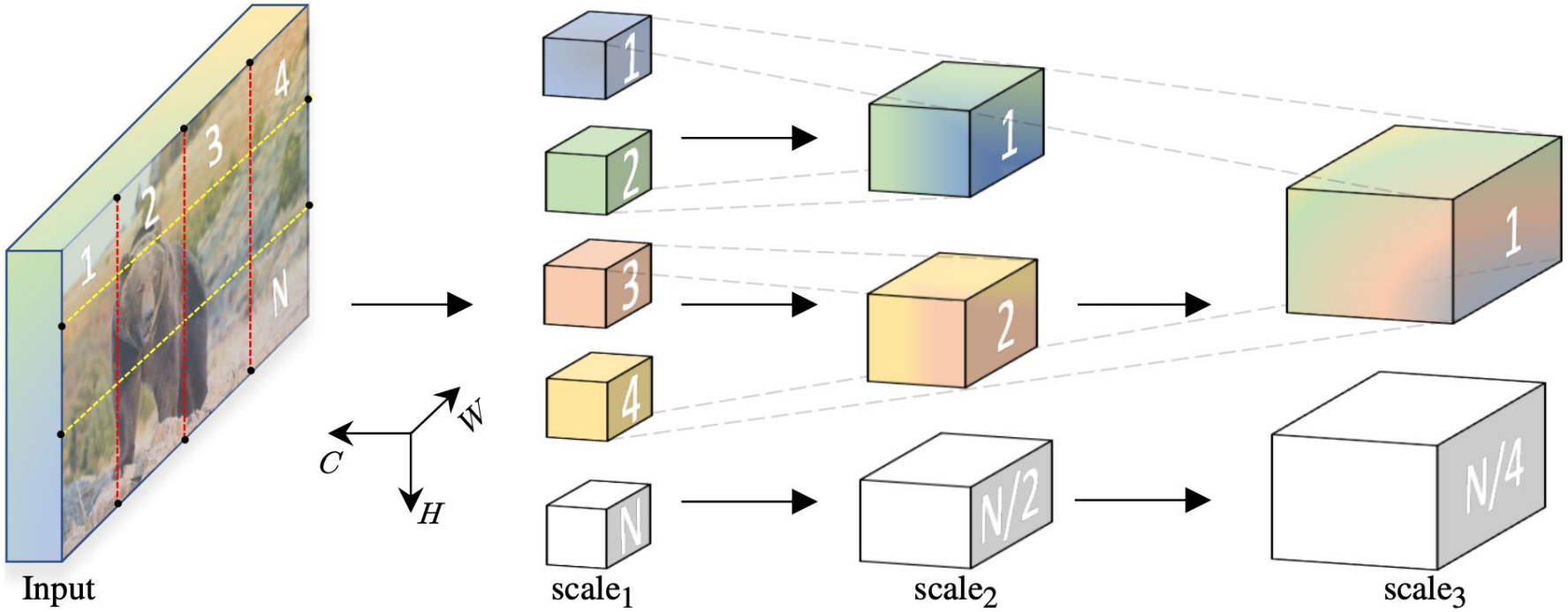

**#TimeSformer** Bertasius, G., Wang, H., & Torresani, L. "Is Space-Time Attention All You Need for Video Understanding?". ICML 2021. [tweet] [official code] [unofficial code]

# The Transformer for Vision: Video



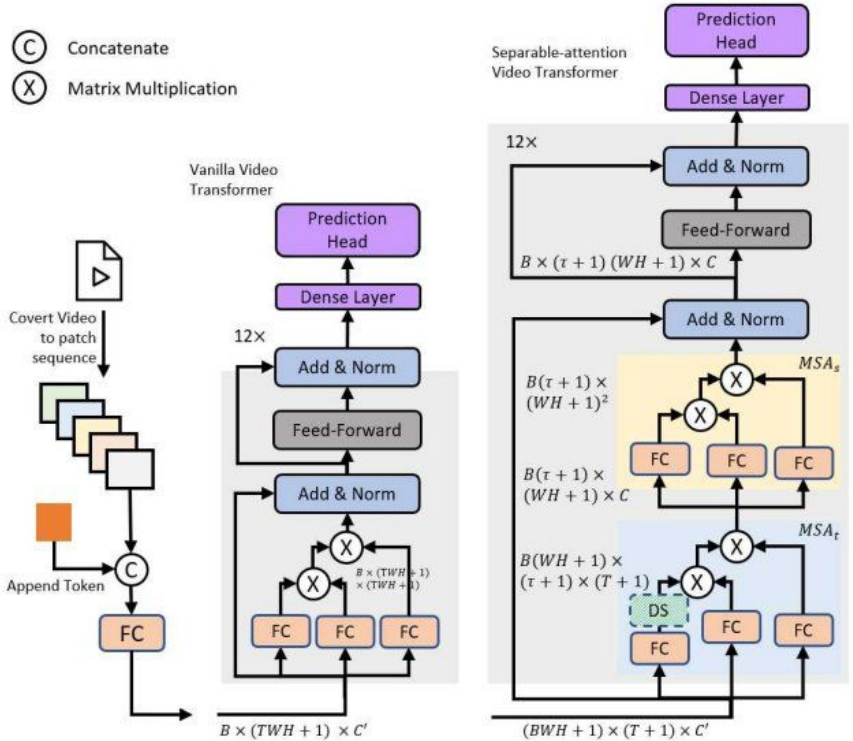

**#ViViT** Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. Vivit: A video vision transformer. ICCV 2021.

# The Transformer for Vision: Video



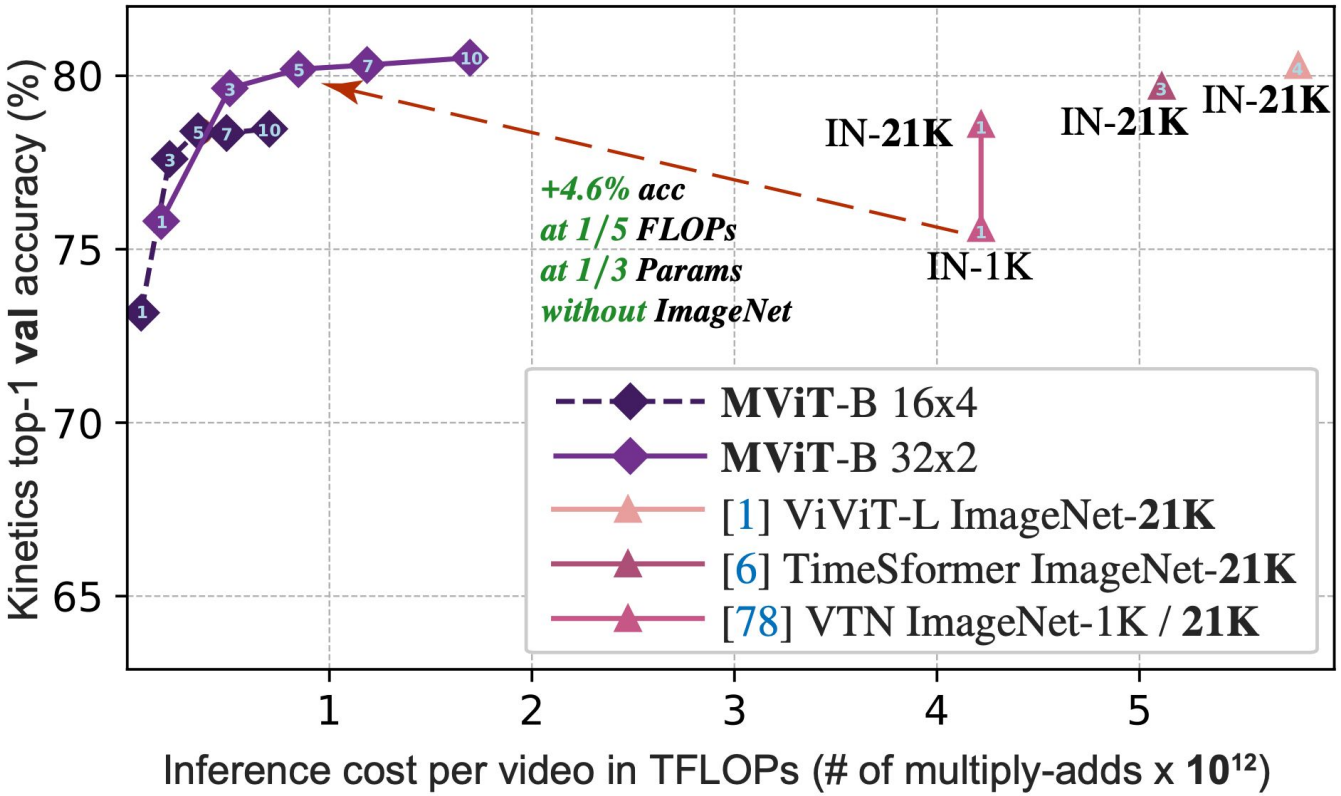

**#MViT** Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., & Feichtenhofer, C. Multiscale vision transformers. ICCV 2021.

# Transformer: Video



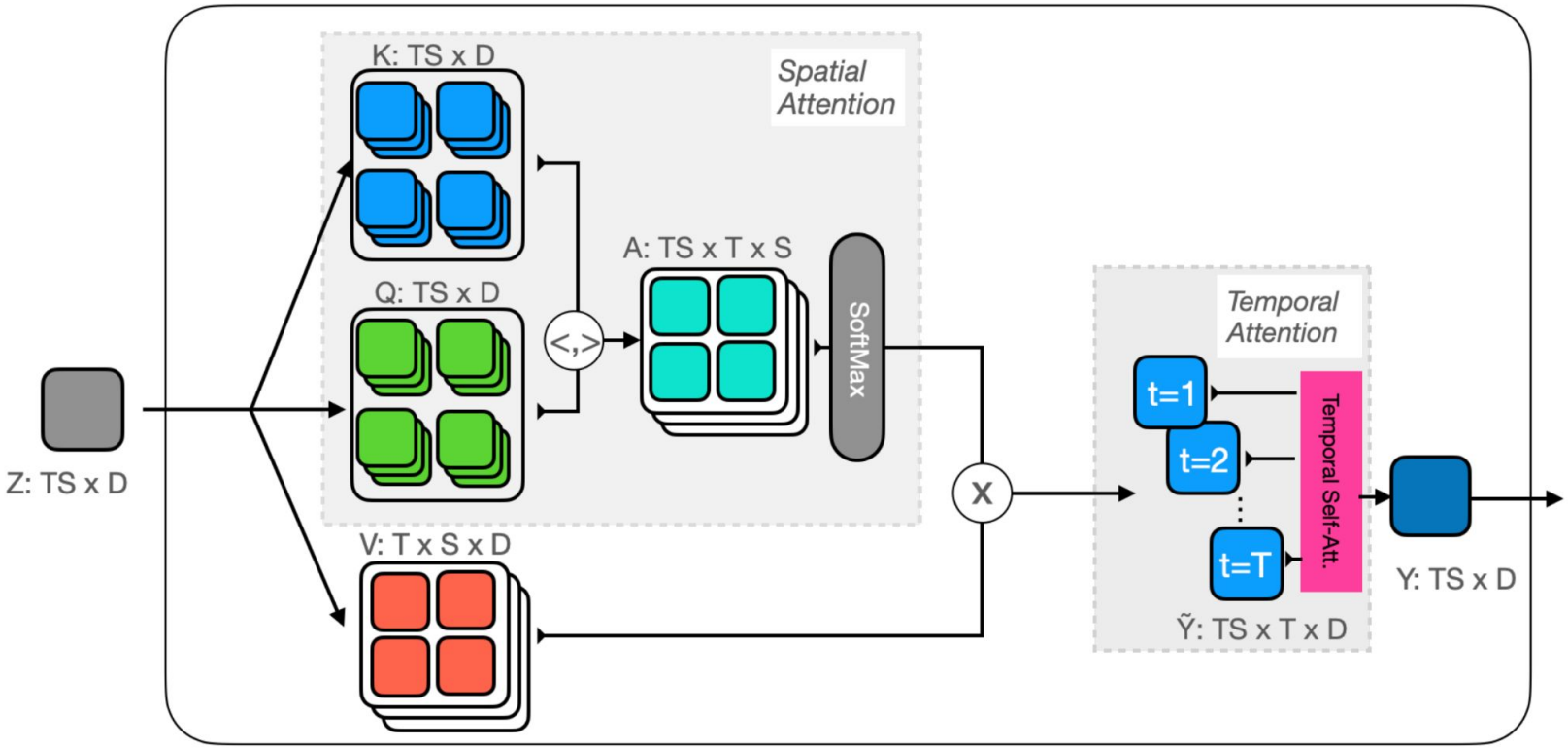

**#VidTr** Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, Joseph Tighe, "VidTr: Video Transformer Without Convolutions". ICCV 2021.

# Transformer: Video



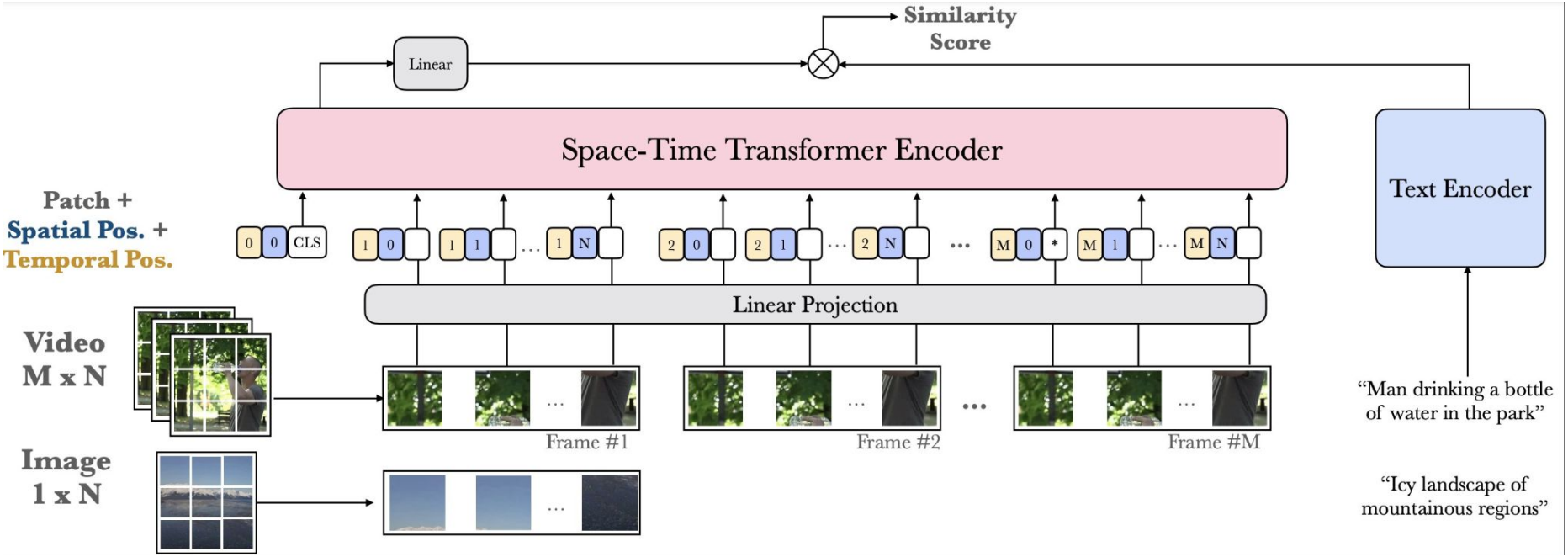

**#MViT** Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., & Feichtenhofer, C. Multiscale vision transformers. ICCV 2021.

# Transformer: Trajectory Estimation



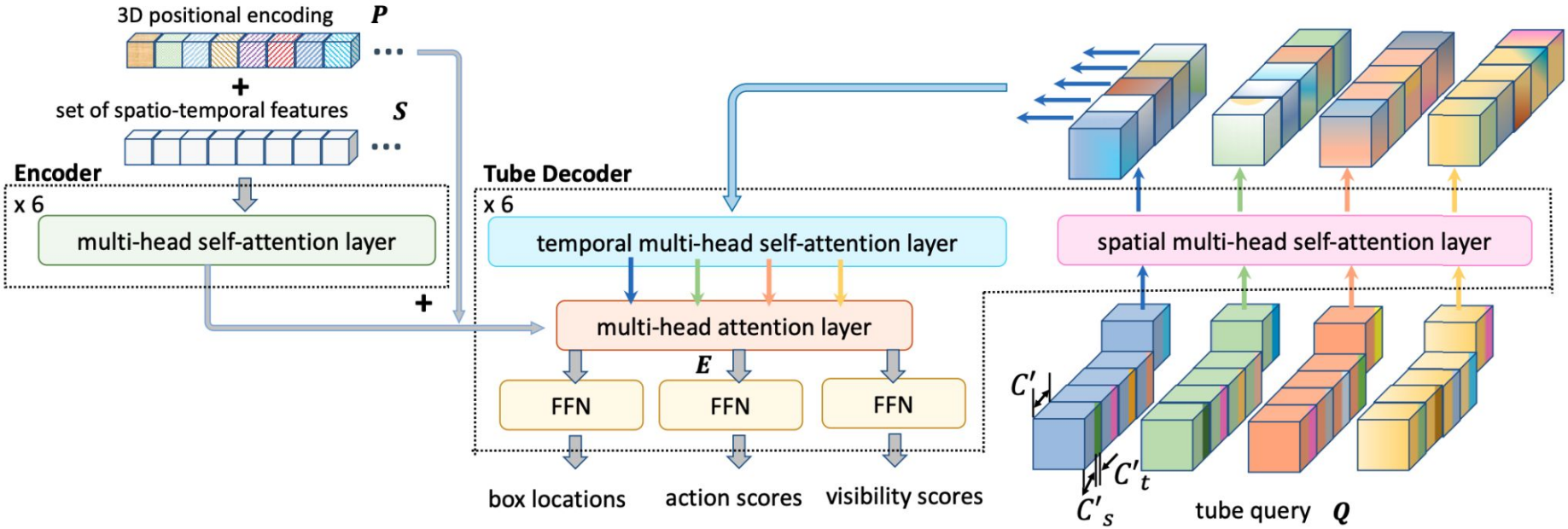

**#MotionFormer** Patrick, M., Campbell, D., Asano, Y., Misra, I., Metze, F., Feichtenhofer, C., Vedaldi, A. and Henriques, J.F., 2021. Keeping your eye on the ball: Trajectory attention in video transformers. NeurIPS 2021. [code]

# Transformer: Retrieval




Max Bain, Arsha Nagrani, Gül Varol, Andrew Zisserman, Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. ICCV 2021. [tweet]

# Transformer: Action Detection




**#TubeR** Zhao, J., Li, A., Liu, C., Bing, S., Chen, H., Snoek, C. G., & Tighe, J. (2021). TubeR: Tube-Transformer for Action Detection. arXiv preprint arXiv:2104.00969.

# Exercise: Context-aware tokens

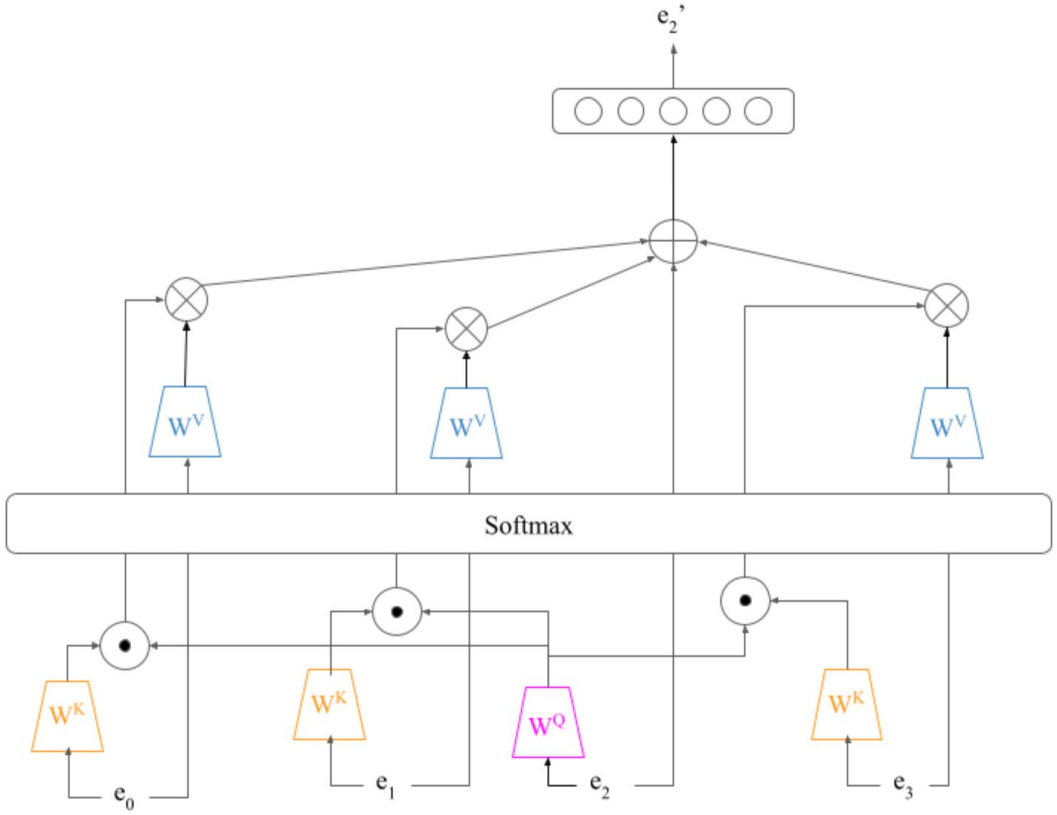
Draw a temporally unfolded representation of a self-attention mechanism to compute the context vector $e'_2$, assuming a single head of projection matrices $W^Q$, $W^K$ and $W^V$ for the queries, keys and values, respectively.

Consider as inputs the set $e_0$, $e_1$, $e_2$ and $e_3$ tokens.

# Solution: Context-aware tokens

Draw a temporally unfolded representation of a self-attention mechanism to compute the context vector $e'_2$, assuming a single head of projection matrices $W^Q$, $W^K$ and $W^V$ for the queries, keys and values, respectively.

Consider as inputs the set $e_0$, $e_1$, $e_2$ and $e_3$ tokens.

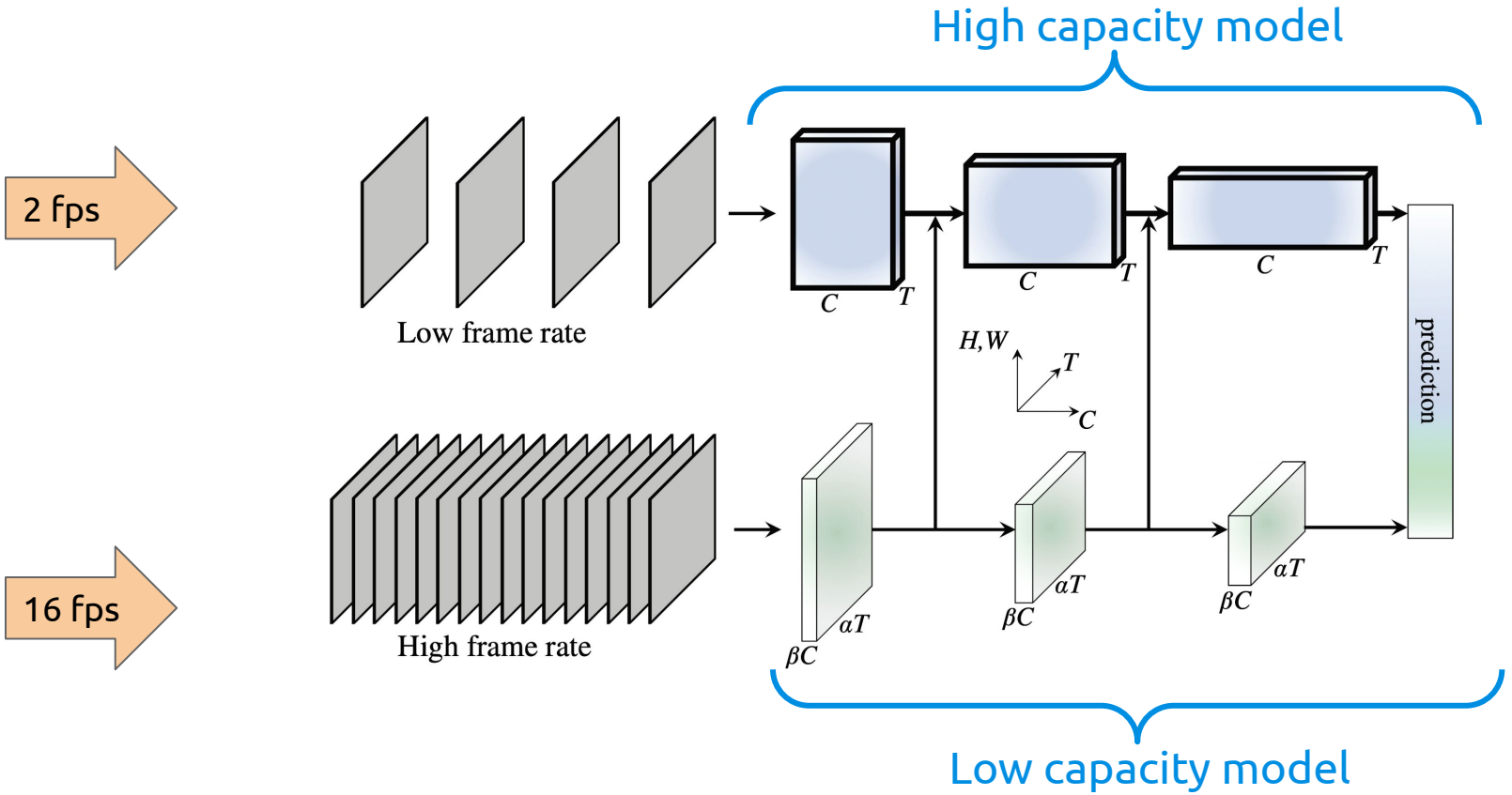


# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. RGB + Optical Flow
5. Transformers
6. **Miscellaneous**

# Dual slow and fast frame rates




**#SlowFast** Feichtenhofer, Christoph, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition." ICCV 2019. [blog] [code]

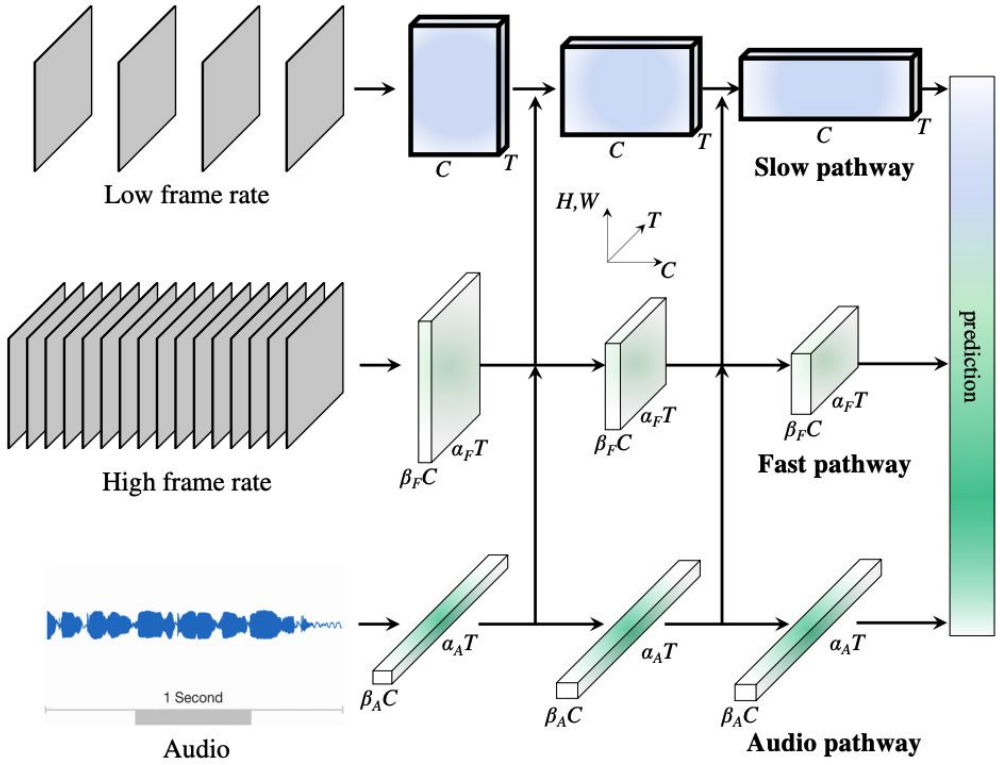
# Dual slow and fast frame rates

Comparison with the state of the art on Kinetics-400

| model | flow | pretrain | top-1 | top-5 | GFLOPs×views |
|---|---|---|---|---|---|
| I3D [5] | | ImageNet | 72.1 | 90.3 | 108 × N/A |
| Two-Stream I3D [5] | ✓ | ImageNet | 75.7 | 92.0 | 216 × N/A |
| S3D-G [61] | ✓ | ImageNet | 77.2 | 93.0 | 143 × N/A |
| Nonlocal R50 [56] | | ImageNet | 76.5 | 92.6 | 282 × 30 |
| Nonlocal R101 [56] | | ImageNet | 77.7 | 93.3 | 359 × 30 |
| R(2+1)D Flow [50] | ✓ | - | 67.5 | 87.2 | 152 × 115 |
| STC [9] | | - | 68.7 | 88.5 | N/A × N/A |
| ARTNet [54] | | - | 69.2 | 88.3 | 23.5 × 250 |
| S3D [61] | | - | 69.4 | 89.1 | 66.4 × N/A |
| ECO [63] | | - | 70.0 | 89.4 | N/A × N/A |
| I3D [5] | ✓ | - | 71.6 | 90.0 | 216 × N/A |
| R(2+1)D [50] | | - | 72.0 | 90.0 | 152 × 115 |
| R(2+1)D [50] | ✓ | - | 73.9 | 90.9 | 304 × 115 |
| **SlowFast** 4×16, R50 | | - | 75.6 | 92.1 | 36.1 × 30 |
| **SlowFast** 8×8, R50 | | - | 77.0 | 92.6 | 65.7 × 30 |
| **SlowFast** 8×8, R101 | | - | 77.9 | 93.2 | 106 × 30 |
| **SlowFast** 16×8, R101 | | - | 78.9 | 93.5 | 213 × 30 |
| **SlowFast** 16×8, R101+NL | | - | **79.8** | **93.9** | 234 × 30 |

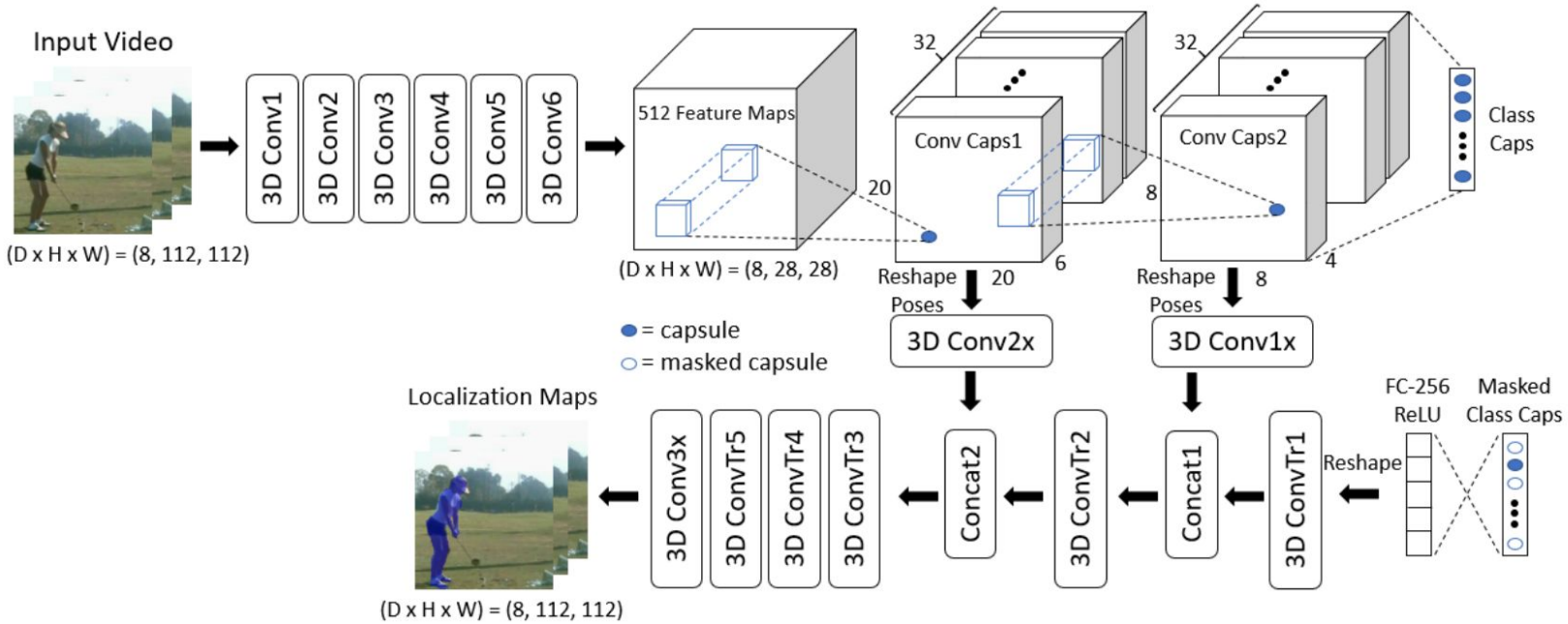
**#SlowFast** Feichtenhofer, Christoph, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition." ICCV 2019. [blog] [code]

# Dual slow and fast frame rates




**#AVSlowFast** Xiao, Fanyi, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. "Audiovisual SlowFast Networks for Video Recognition." arXiv preprint arXiv:2001.08740 (2020).

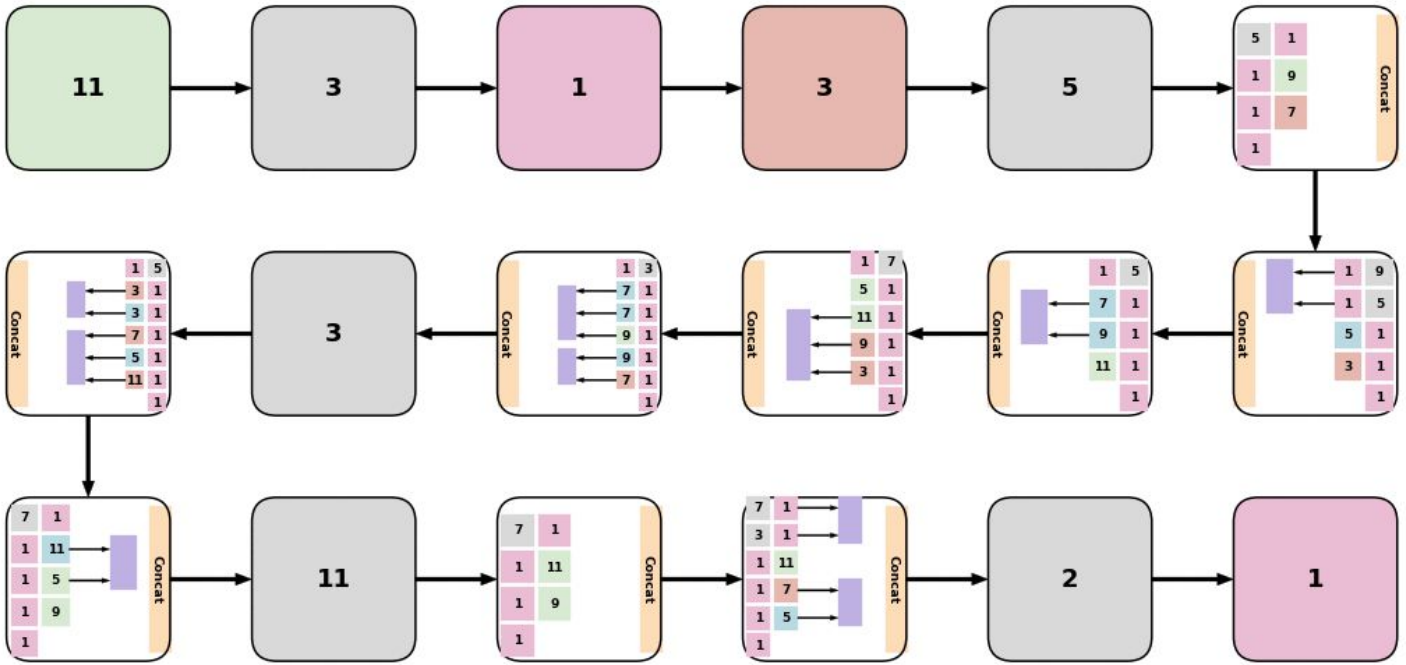
# Capsule Networks




**#Videocapsulenet** Duarte, Kevin, Yogesh Rawat, and Mubarak Shah. "Videocapsulenet: A simplified network for action detection." NeurIPS 2018. [code]

# Neural Architecture Search

Evolutionary strategy to find the best combination of building blocks.

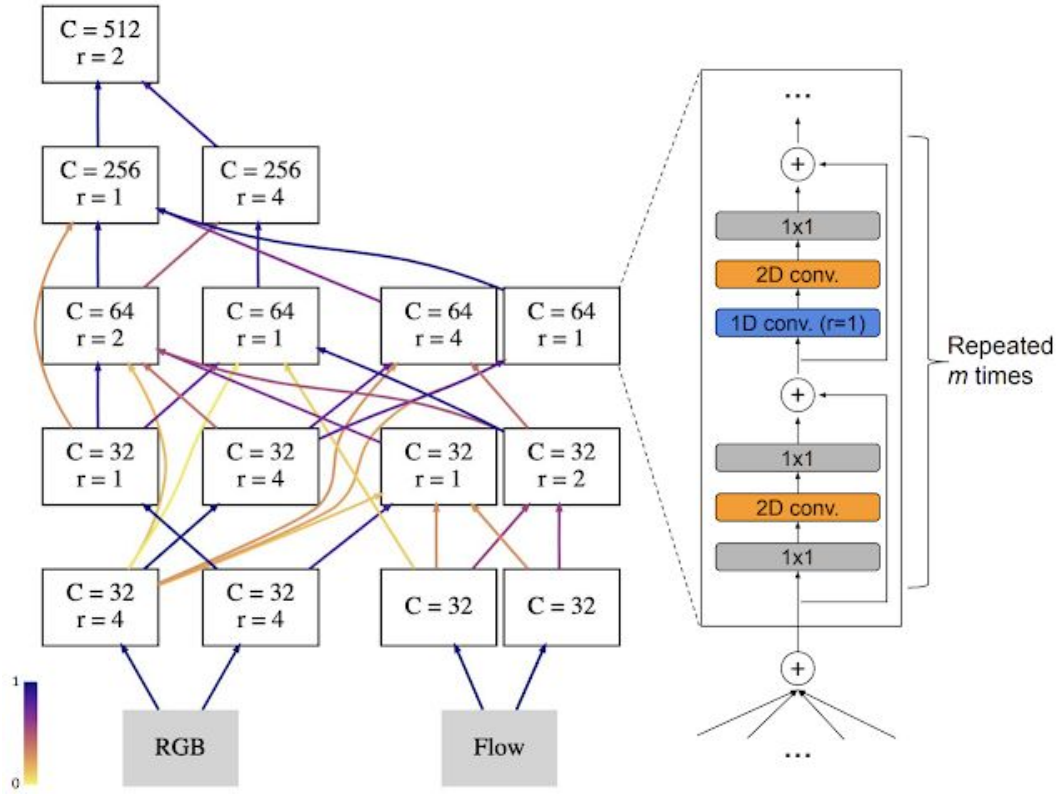



**#EvaNet** Piergiovanni, A. J., et al. "Evolving space-time neural architectures for videos." ICCV 2019. [blog] [code]

# Neural Architecture Search

**Four-stream** architecture with various intermediate connections for the first time:

- 2 streams per RGB and optical flow…
- ...each one at different temporal resolutions.

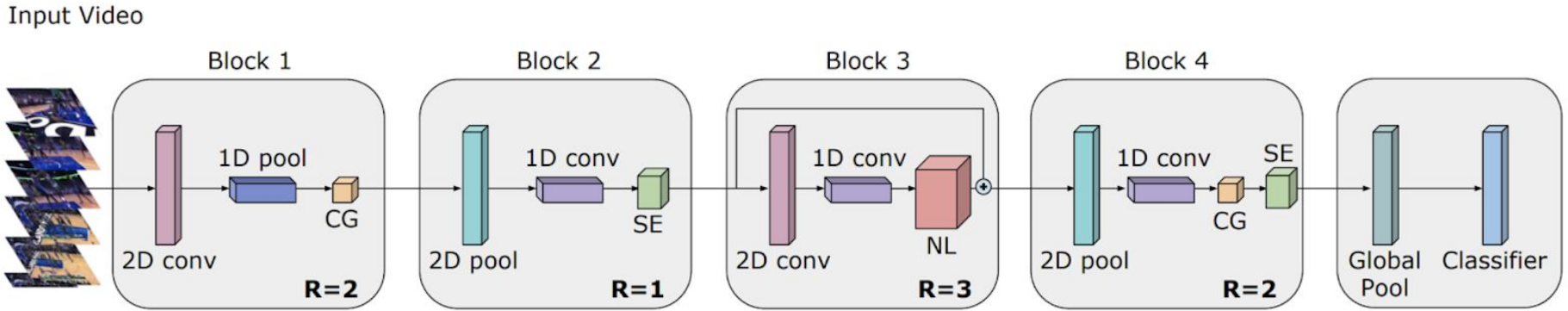



**#AssembleNet** Ryoo, M.S., Piergiovanni, A.J., Tan, M. and Angelova, A.,. Assemblenet: Searching for multi-stream neural connectivity in video architectures. ICLR 2020.

# Neural Architecture Search

Computationally efficient architectures suitable for mobile devices.

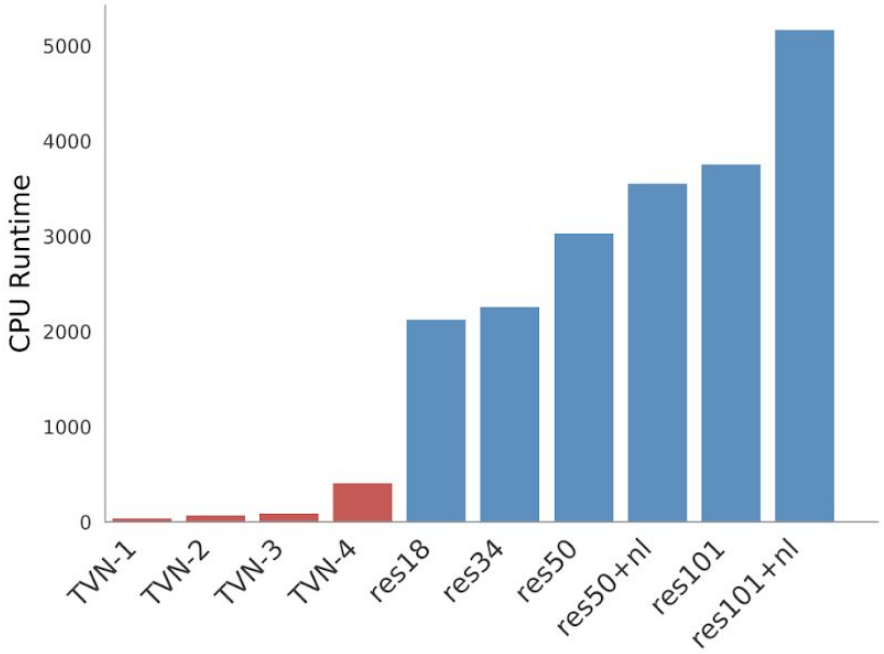
TVN-1

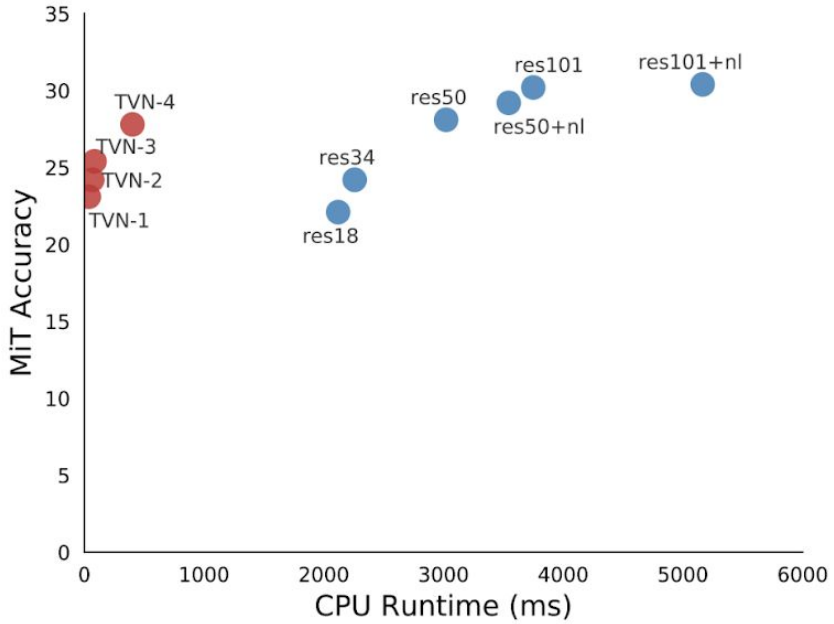



**#TVN** Piergiovanni, A. J., Anelia Angelova, and Michael S. Ryoo. "Tiny Video Networks." arXiv preprint arXiv:1910.06961 (2019).

# Neural Architecture Search

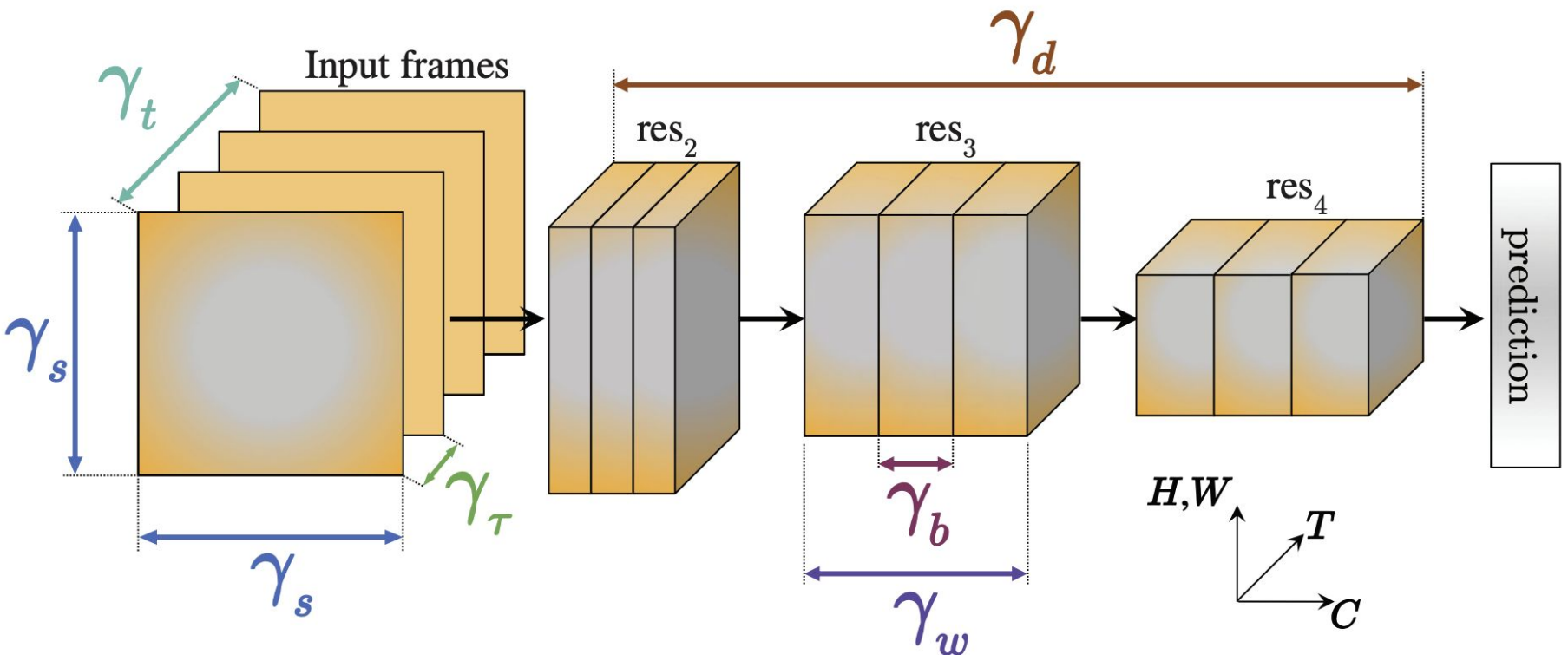
Computationally efficient architectures suitable for mobile devices.




**#TVN** Piergiovanni, A. J., Anelia Angelova, and Michael S. Ryoo. "Tiny Video Networks." arXiv preprint arXiv:1910.06961 (2019).

# Neural Architecture Search




**#X3D** Feichtenhofer, Christoph. "X3D: Expanding architectures for efficient video recognition." CVPR 2020.

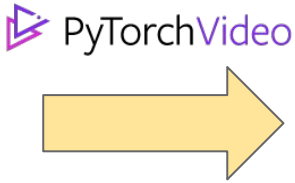
# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. RGB + Optical Flow
5. Transformers
6. Miscellaneous

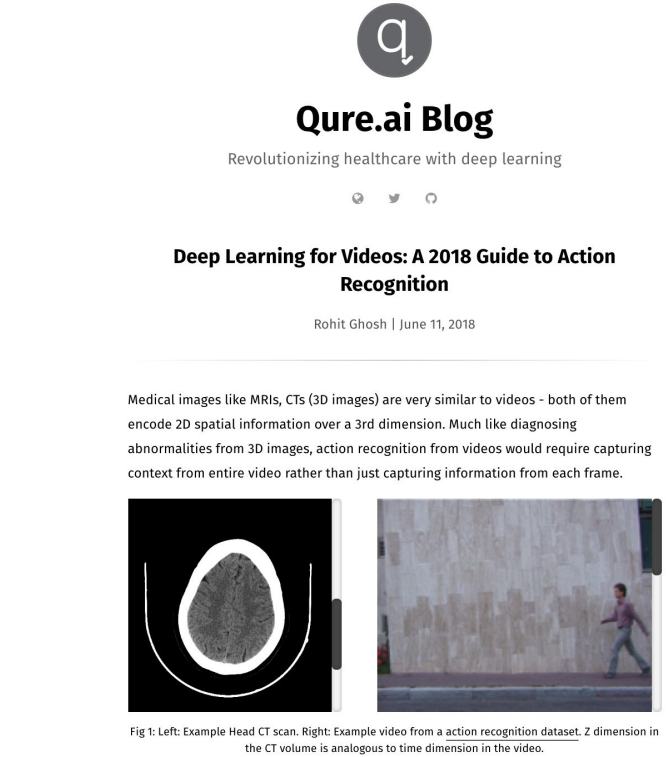
# Implementations

- PyTorchVideo
- Facebook FAIR
  - SlowFast
  - SlowOnly
  - C2D
  - I3D
  - Non-local Network
  - X3D
  - Multiscale Vision Transformers
- Torchvision
  - ResNet 3D 18
  - ResNet MC 18
  - ResNet(2+1)D
- PyTorch Lightning
- Microsoft Computer Vision Recipes
  - R(2+1)D




| arch | depth | pretrain | frame length x sample rate | top 1 | top 5 | Flops (G) x views | Params (M) | Model |
|---|---|---|---|---|---|---|---|---|
| C2D | R50 | - | 8x8 | 71.46 | 89.68 | 25.89 x 3 x 10 | 24.33 | link |
| I3D | R50 | - | 8x8 | 73.27 | 90.70 | 37.53 x 3 x 10 | 28.04 | link |
| Slow | R50 | - | 4x16 | 72.40 | 90.18 | 27.55 x 3 x 10 | 32.45 | link |
| Slow | R50 | - | 8x8 | 74.58 | 91.63 | 54.52 x 3 x 10 | 32.45 | link |
| SlowFast | R50 | - | 4x16 | 75.34 | 91.89 | 36.69 x 3 x 10 | 34.48 | link |
| SlowFast | R50 | - | 8x8 | 76.94 | 92.69 | 65.71 x 3 x 10 | 34.57 | link |
| SlowFast | R101 | - | 8x8 | 77.90 | 93.27 | 127.20 x 3 x 10 | 62.83 | link |
| SlowFast | R101 | - | 16x8 | 78.70 | 93.61 | 215.61 x 3 x 10 | 53.77 | link |
| CSN | R101 | - | 32x2 | 77.00 | 92.90 | 75.62 x 3 x 10 | 22.21 | link |
| R(2+1)D | R50 | - | 16x4 | 76.01 | 92.23 | 76.45 x 3 x 10 | 28.11 | link |
| X3D | XS | - | 4x12 | 69.12 | 88.63 | 0.91 x 3 x 10 | 3.79 | link |
| X3D | S | - | 13x6 | 73.33 | 91.27 | 2.96 x 3 x 10 | 3.79 | link |
| X3D | M | - | 16x5 | 75.94 | 92.72 | 6.72 x 3 x 10 | 3.79 | link |
| X3D | L | - | 16x5 | 77.44 | 93.31 | 26.64 x 3 x 10 | 6.15 | link |
| MViT | B | - | 16x4 | 78.85 | 93.85 | 70.80 x 1 x 5 | 36.61 | link |
| MViT | B | - | 32x3 | 80.30 | 94.69 | 170.37 x 1 x 5 | 36.61 | link |

# Learn more



**Qure.ai Blog**

Revolutionizing healthcare with deep learning

**Deep Learning for Videos: A 2018 Guide to Action Recognition**

Rohit Ghosh | June 11, 2018

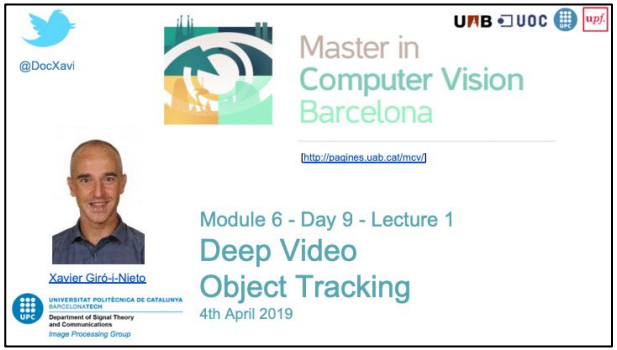
Medical images like MRIs, CTs (3D images) are very similar to videos - both of them encode 2D spatial information over a 3rd dimension. Much like diagnosing abnormalities from 3D images, action recognition from videos would require capturing context from entire video rather than just capturing information from each frame.

Fig 1: Left: Example Head CT scan. Right: Example video from a action recognition dataset. Z dimension in the CT volume is analogous to time dimension in the video.

Rohit Ghosh, "Deep Learning for Videos: A 2018 Guide for Action recognition" (2018)


Visual Recognition for Images, Video, and 3D

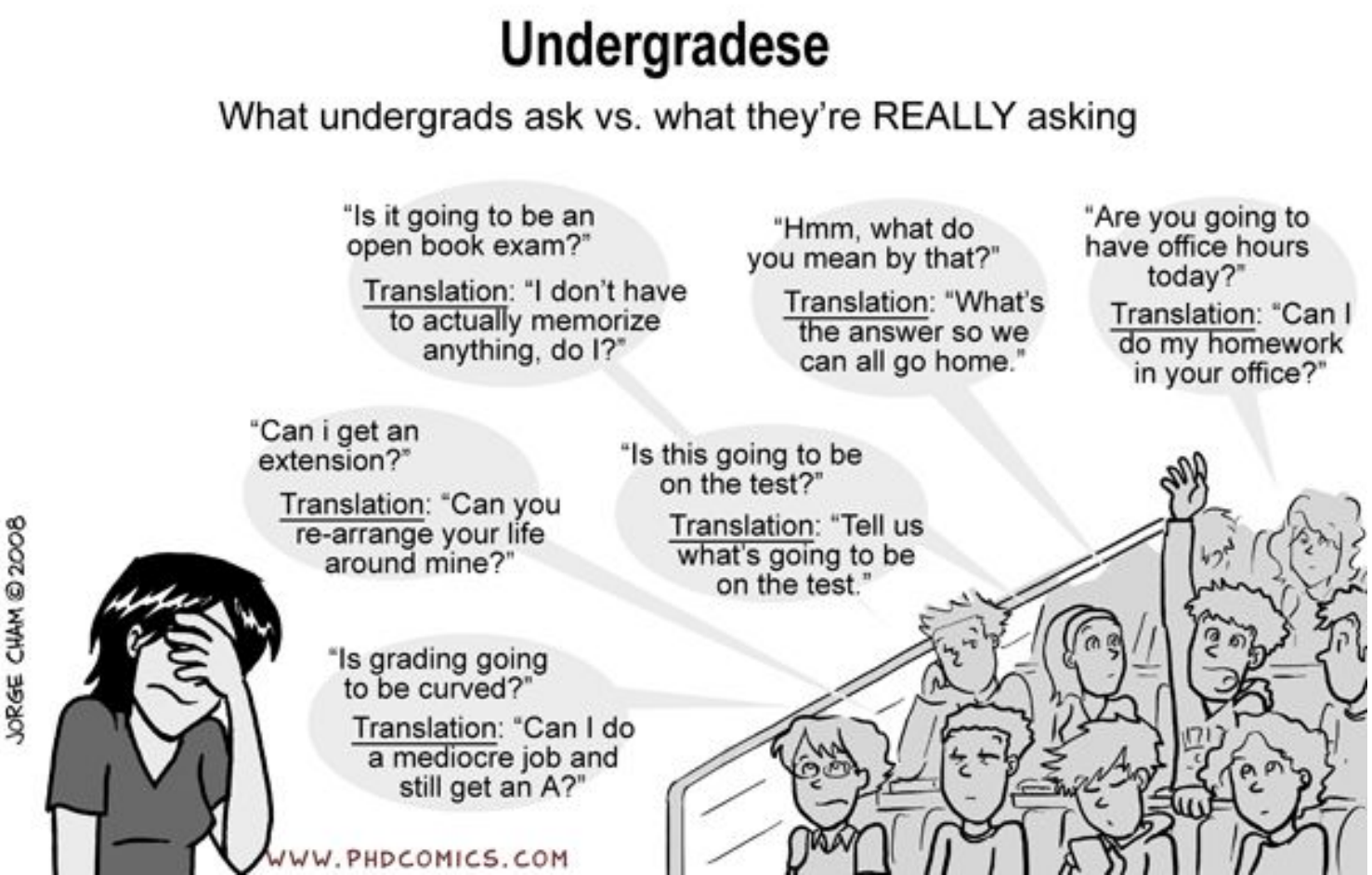
# Additional content [index]




Object tracking
[slides] [video]


Video Object segmentation
[slides] [video]

# Questions ?


Undergradese
What undergrads ask vs. what they're REALLY asking
"Is it going to be an open book exam?"
Translation: "I don't have to actually memorize anything, do I?"
"Hmm, what do you mean by that?"
Translation: "What's the answer so we can all go home."
"Are you going to have office hours today?"
Translation: "Can I do my homework in your office?"
"Can i get an extension?"
Translation: "Can you re-arrange your life around mine?"
"Is this going to be on the test?"
Translation: "Tell us what's going to be on the test."
"Is grading going to be curved?"
Translation: "Can I do a mediocre job and still get an A?"
JORGE CHAM © 2008
WWW.PHDCOMICS.COM

# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. RGB + Optical Flow
5. Transformers
6. Miscellaneous
7. **Annex I: Exercises**
8. Annex II: Datasets

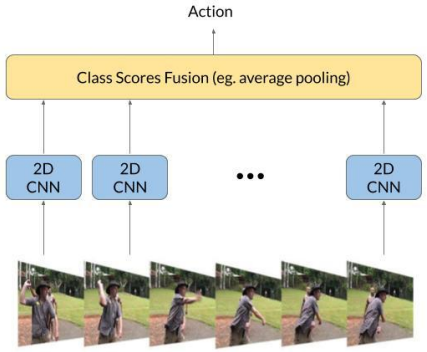
# Exercise 1

Consider a deep neural network able to solve the action recognition task in a video clip composed of K=6 frames. Draw a scheme depicting a basic architecture for each of the following set ups:

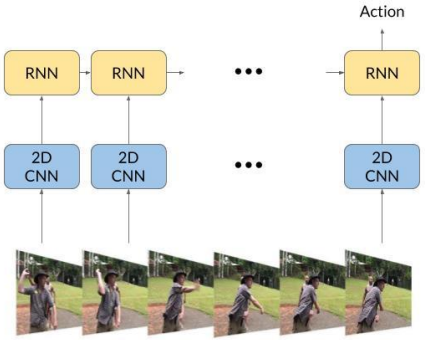
Single frame model
CNN + RNN
3D CNN
Two-Stream

# Exercise 1

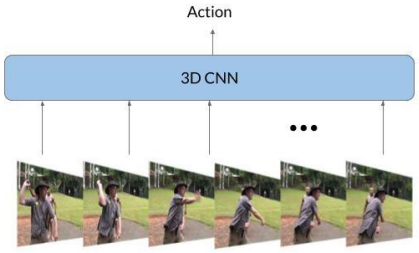
a) Single frame model



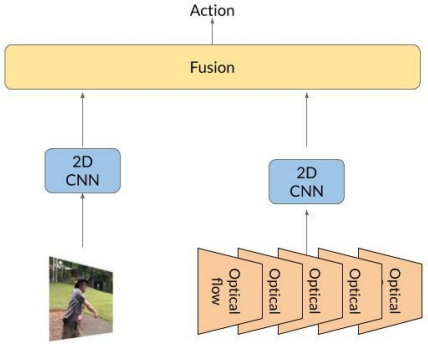

b) CNN+RNN




c) 3D CNN




d) Two-stream



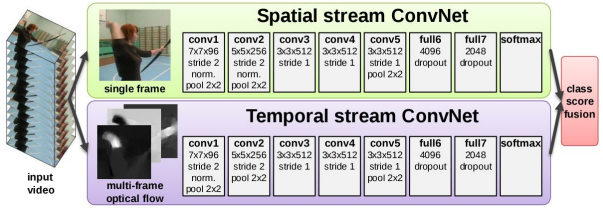
# Exercise 2

Draw a scheme of the two-stream deep convolutional network used for action recognition in videos, as presented by Symonian and Zisserman (NIPS 2014) and Feichtenhofer, Pinz and Zisserman (CVPR 2016). Indicate the main difference between the two proposals.

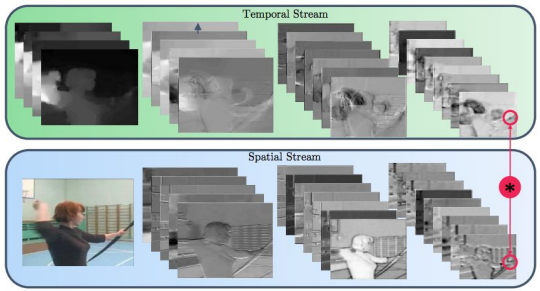
# Exercise 2

Draw a scheme of the two-stream deep convolutional network used for action recognition in videos, as presented by Symonian and Zisserman (NIPS 2014) and Feichtenhofer, Pinz and Zisserman (CVPR 2016). Indicate the main difference between the two proposals.

Symonian and Zisserman (NIPS 2014)




Feichtenhofer, Pinz and Zisserman (CVPR 2016)



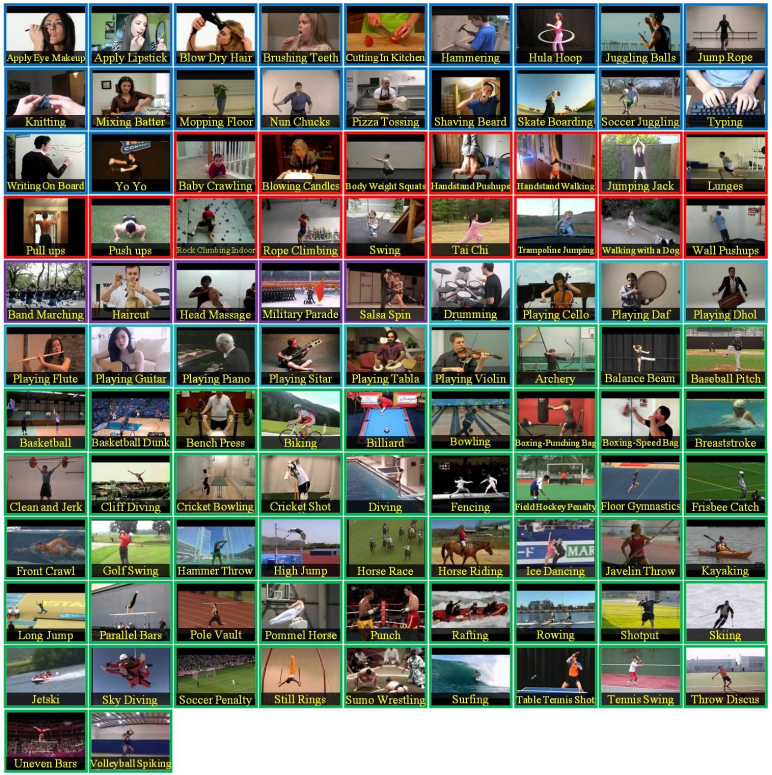

The main difference between the two solutions is how the spatial and temporal streams were fused. In NIPS 2014 the final class scores where fused, while in CVPR 2016 the fusion was performed at the last convolutional layer.

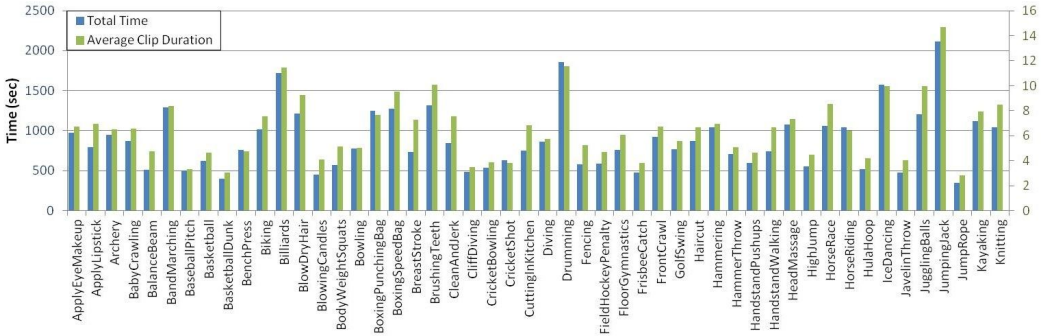
# Deep Video Architectures

Basic deep architectures for video:

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. RGB + Optical Flow
5. Transformers
6. Advanced solutions
7. Annex I: Exercises
8. **Annex II: Datasets**

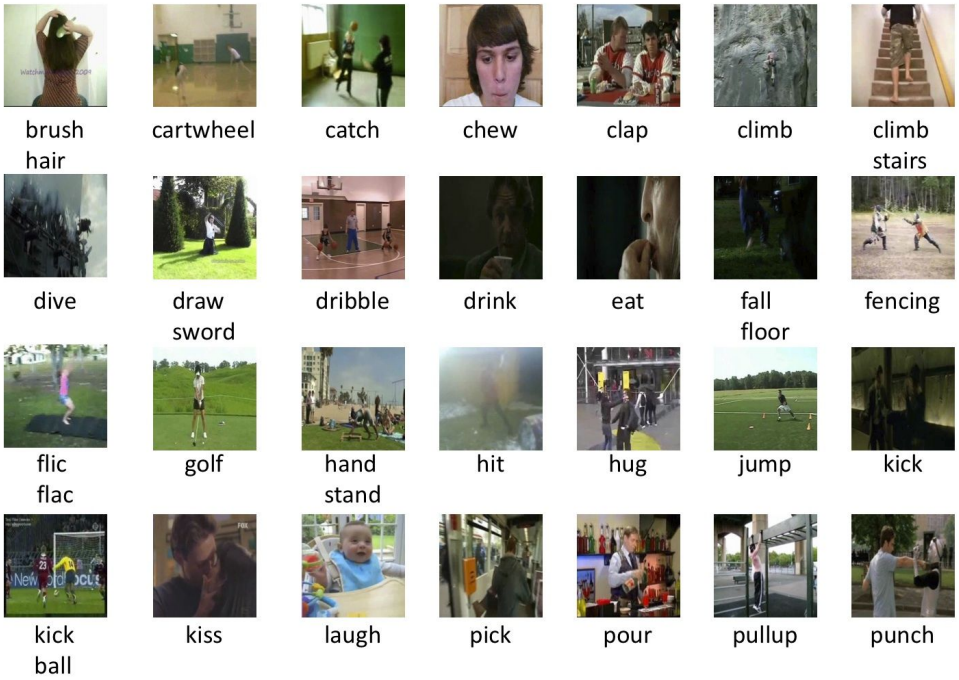
# Datasets: UCF-101



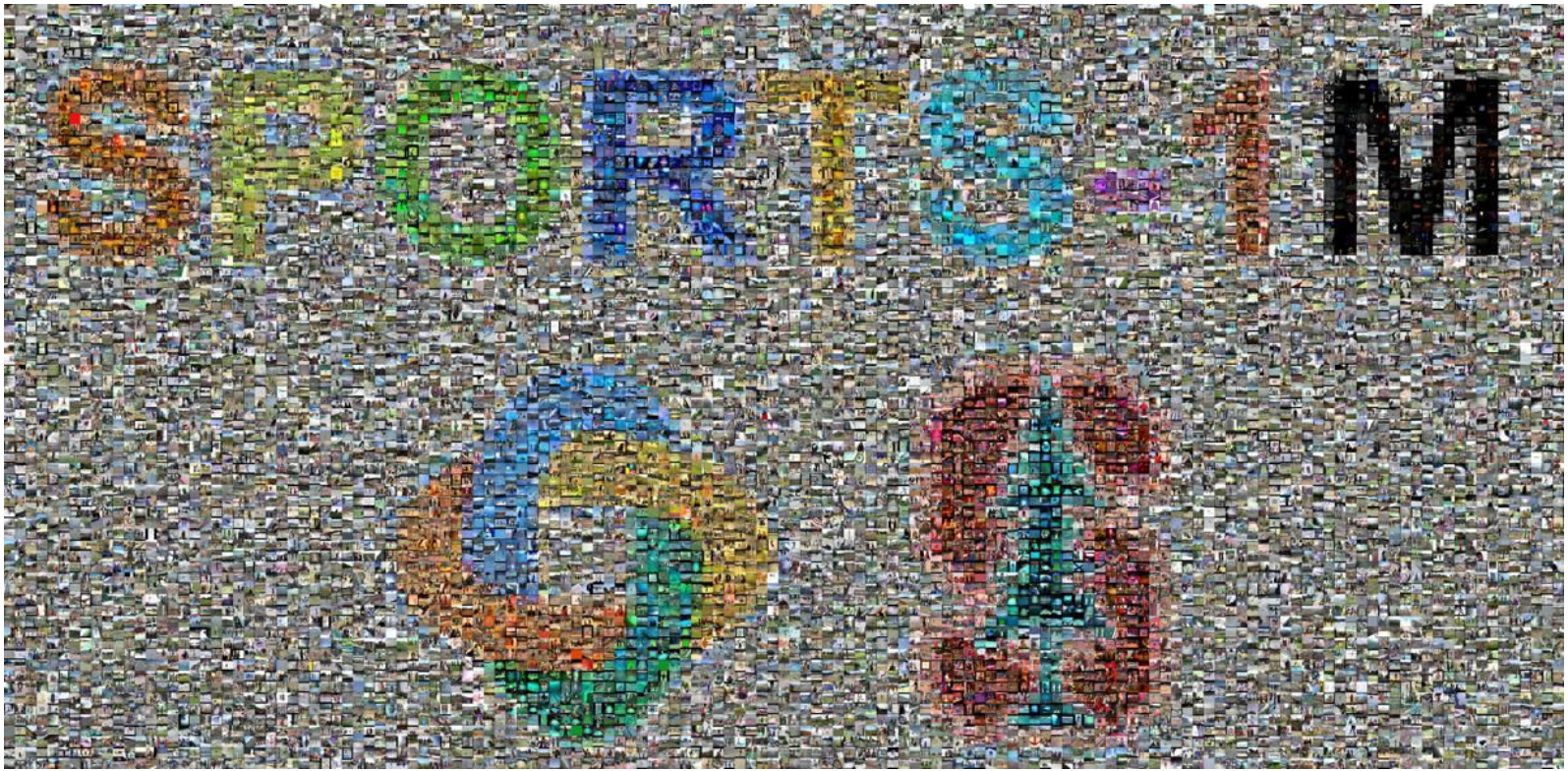

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

# Datasets: HDMB51 (Brown University)



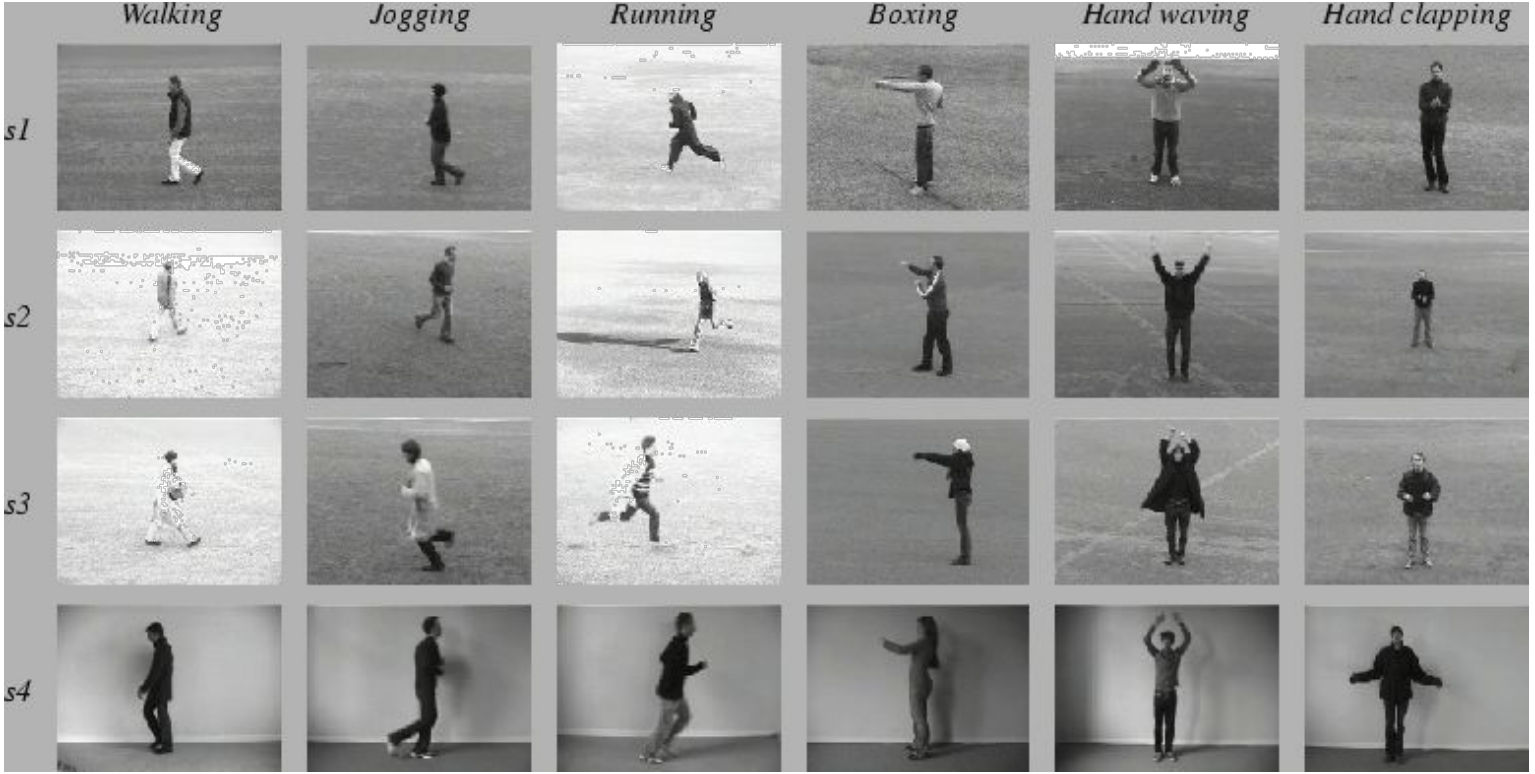

Kuehne, Hildegard, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. "HMDB: a large video database for human motion recognition." ICCV 2011.

# Datasets: Sports 1M (Stanford)



Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. Large-scale video classification with convolutional neural networks. CVPR 2014.

# Datasets: KTH

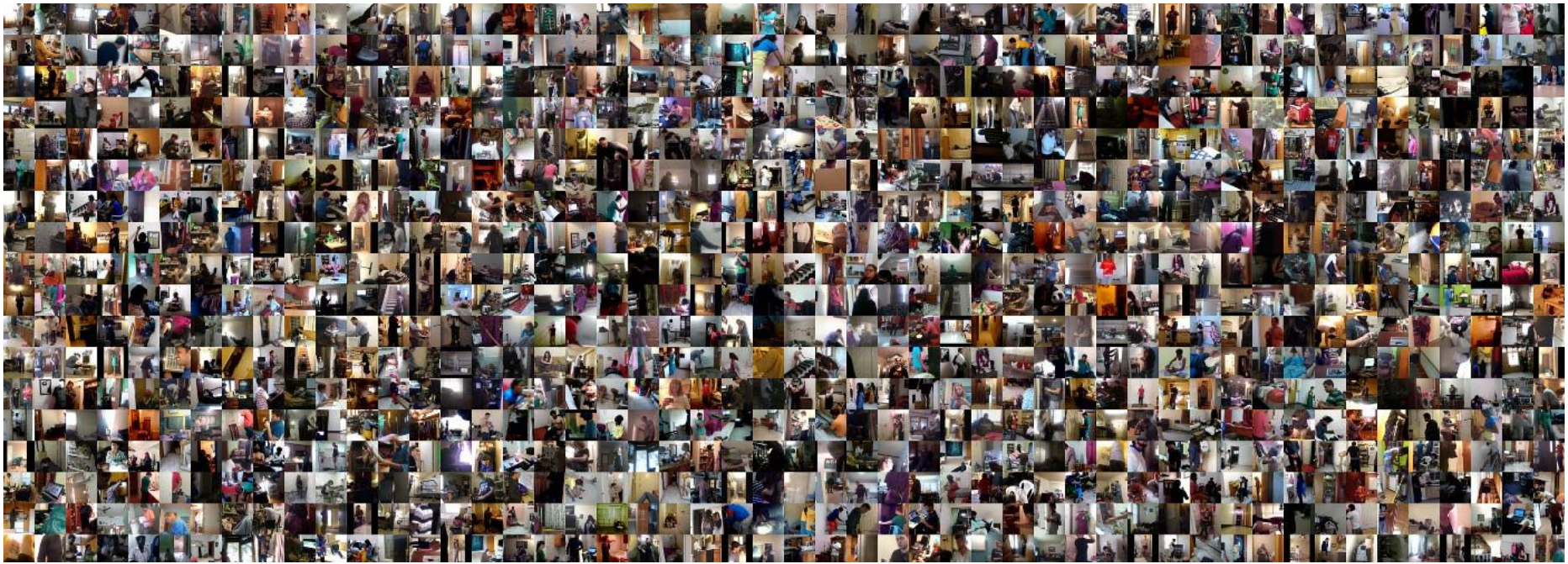



Schuldt, Christian, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." In Pattern Recognition, 2004. ICPR 2004.

# Datasets: ActivityNet

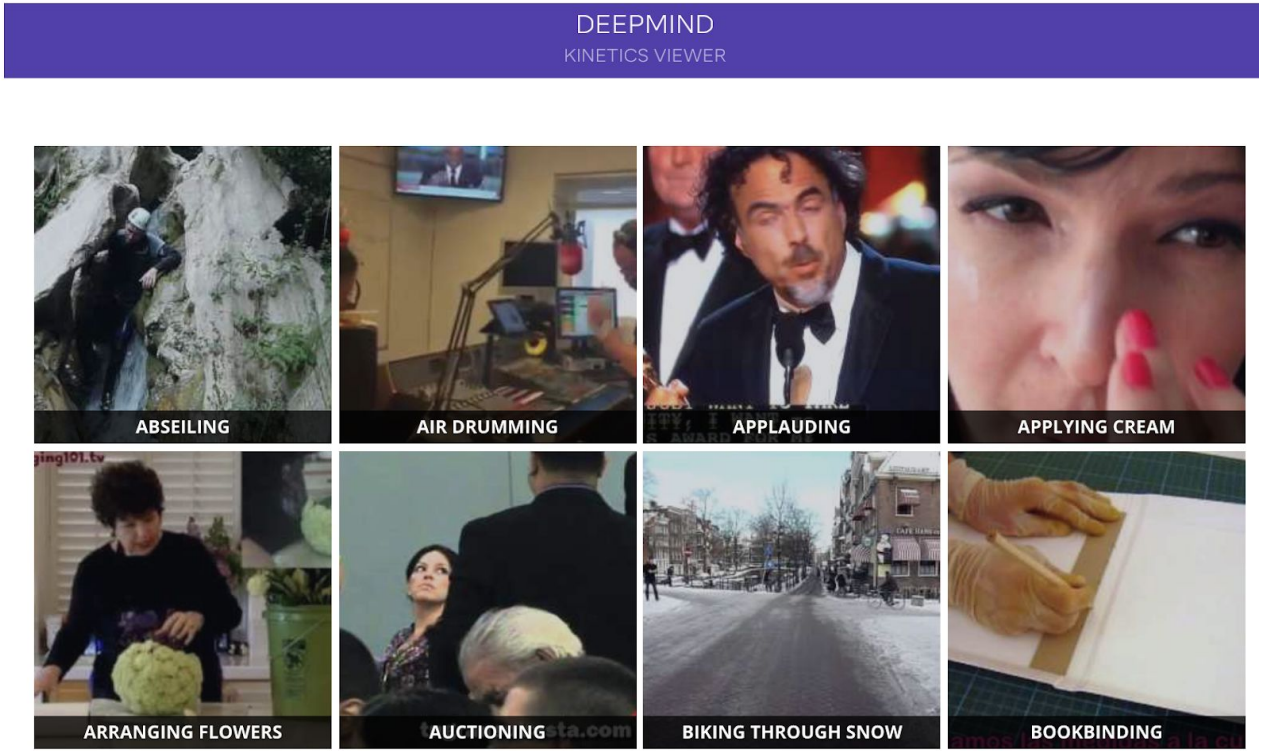

Heilbron, F.C., Escorcia, V., Ghanem, B. and Niebles, J.C.,. "Activitynet: A large-scale video benchmark for human activity understanding". CVPR 2015.

# Datasets: Charades (Allen AI)



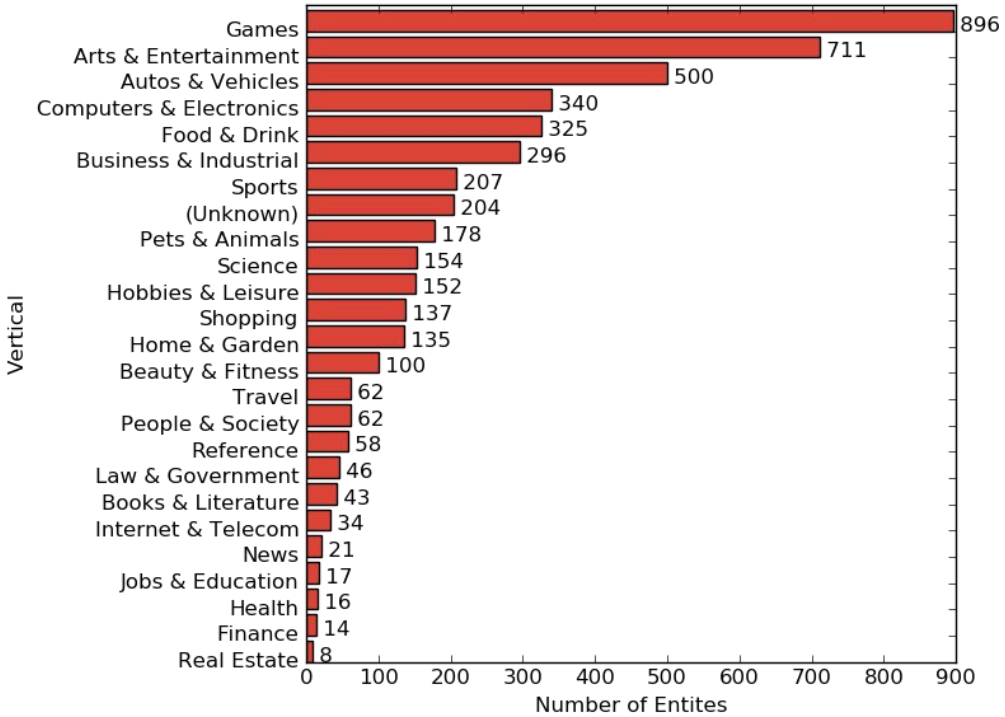
Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., & Gupta, A. (2016, October). Hollywood in homes: Crowdsourcing data collection for activity understanding. ECCV 2016. [Dataset] [Code]

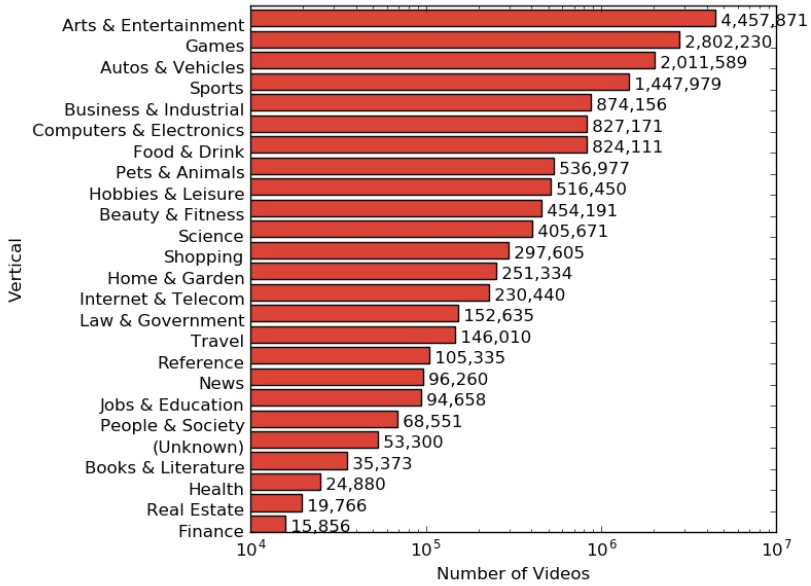
# Datasets: Kinectics (DeepMind)




Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... & Suleyman, M. (2017). The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.

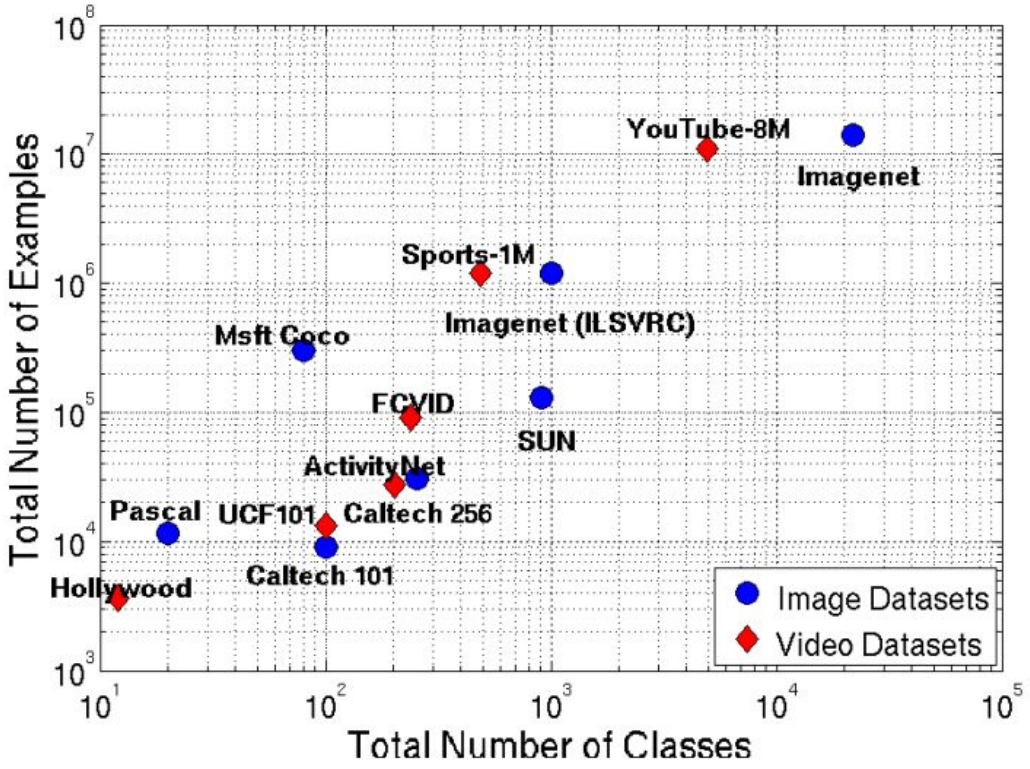
# Datasets: YouTube-8M (Google)



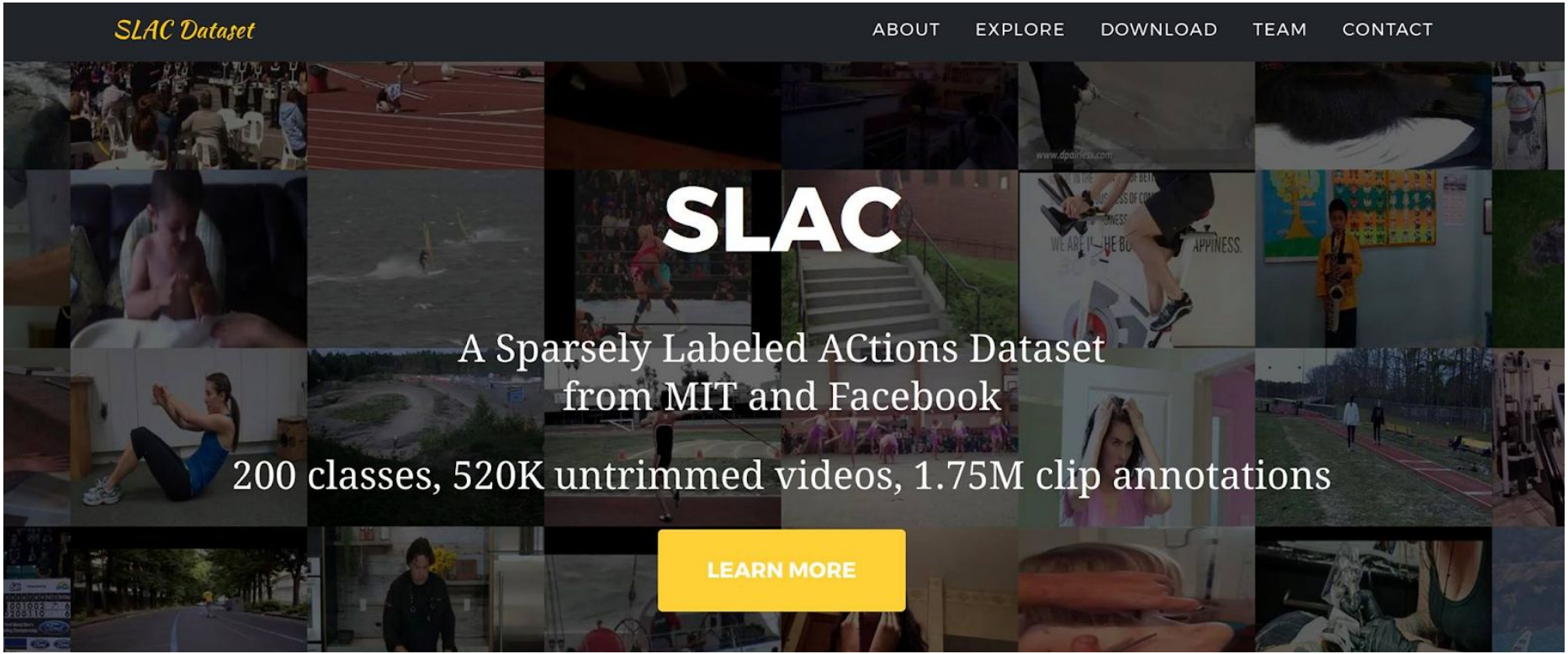

(Slides by Dídac Surís) Abu-El-Haija, Sami, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675 (2016). [project]

# Activity Recognition: Datasets



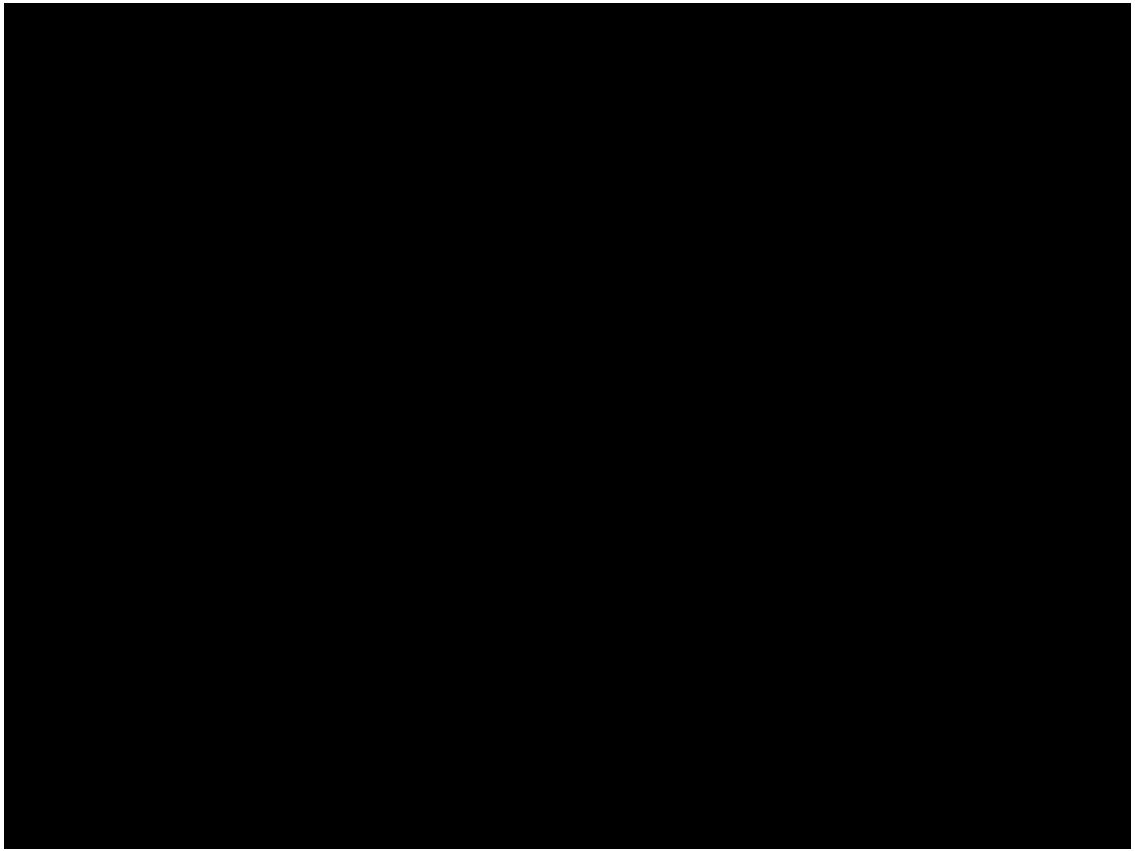

(Slides by Dídac Surís) Abu-El-Haija, Sami, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675 (2016). [project]

# Datasets: SLAC (MIT & Facebook)




Hang Zhao, Zhicheng Yan, Heng Wang, Lorenzo Torresani, Antonio Torralba, "SLAC: A Sparsely Labeled Dataset for Action Classification and Localization" arXiv 2017 [project page]

# Datasets: Moments in Time (MIT & IBM)



Monfort, Mathew, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa Brown et al. "Moments in Time Dataset: one million videos for event understanding." arXiv preprint arXiv:1801.03150 (2018).

# Datasets: DALY (INRIA)


DALY contains the following spatial annotations:

- bounding box around the action
- upper body pose annotation, including a bounding box around the head
- bounding box around object(s) involved in the action

Weinzaepfel, Philippe, Xavier Martin, and Cordelia Schmid. "Human Action Localization with Sparse Spatial Supervision." (2017).

# Datasets: AVA (Berkeley & Google)

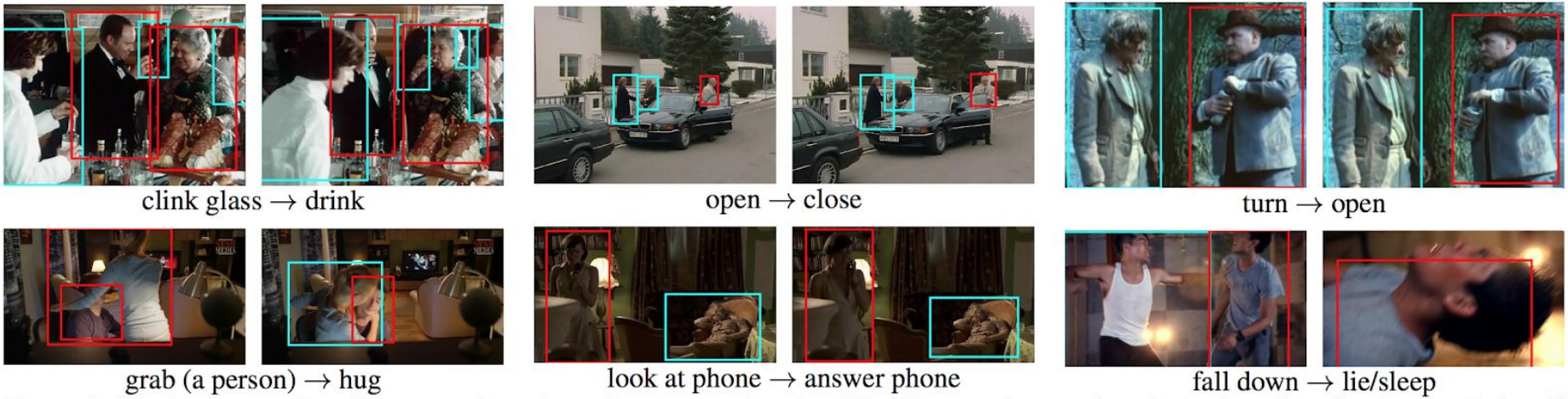



Figure 4. We show examples of how atomic actions change over time in AVA. The text shows pairs of atomic actions for the people in red bounding boxes. Temporal information is key for recognizing many of the actions and appearance can substantially vary within an action category, such as opening a door or bottle.

Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D. A., Toderici, G., ... & Malik, J. (2017). AVA: A video dataset of spatio-temporally localized atomic visual actions. arXiv preprint arXiv:1705.08421.