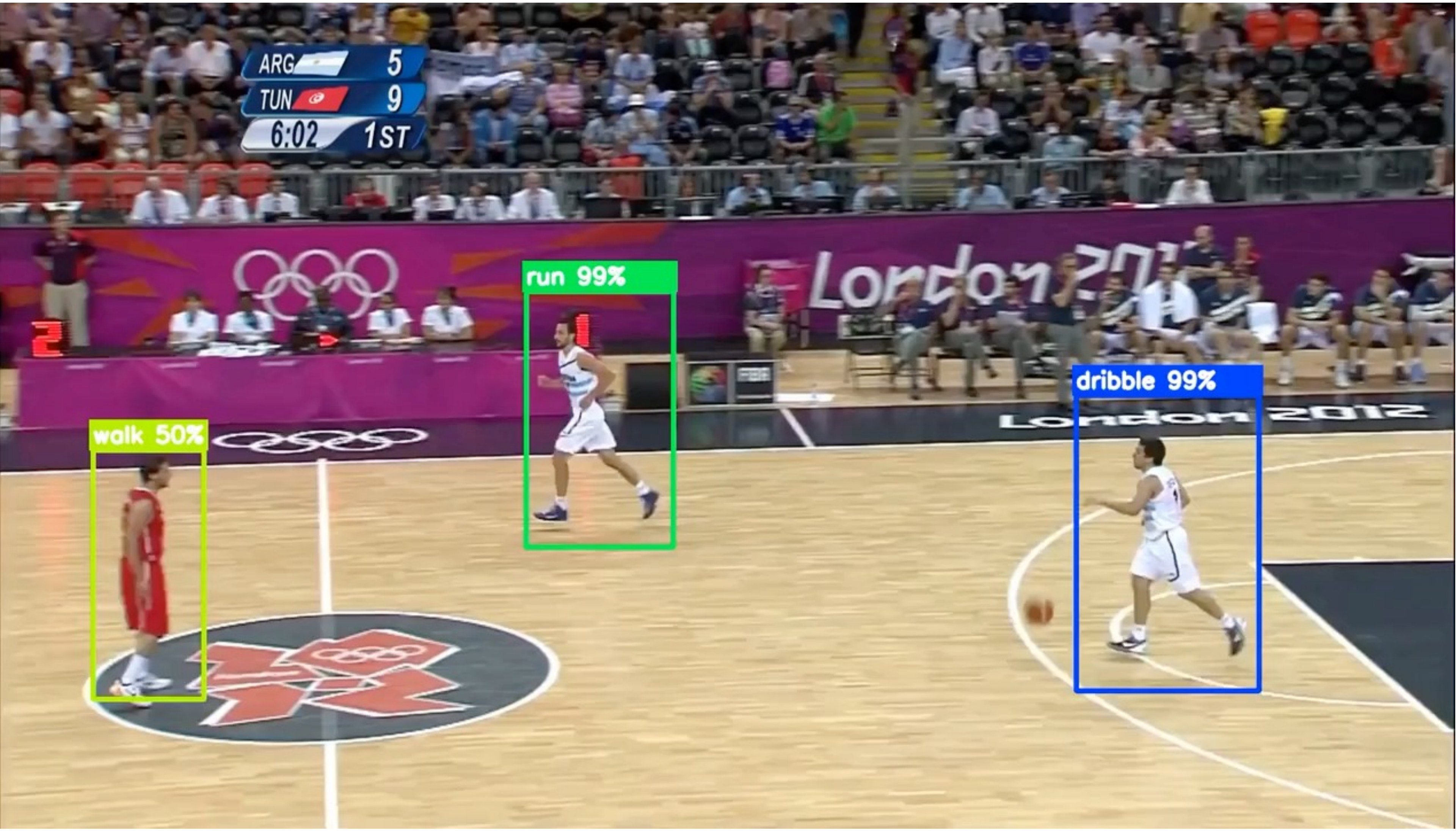
Computer Vision:
Video understanding

Siheng Chen 陈思衡

Action Recognition in Videos



Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Action Recognition in Videos

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Video Understanding: Action recognition

Hand-crafted features → CNN

DeepVideo

Two-stream CNN

Two-stream → TSN

3D CNN

c3d → i3d → r(2+1)d → nonlocal neural networks

Video Transformer

TimeSFormer

CNN

DeepVideo

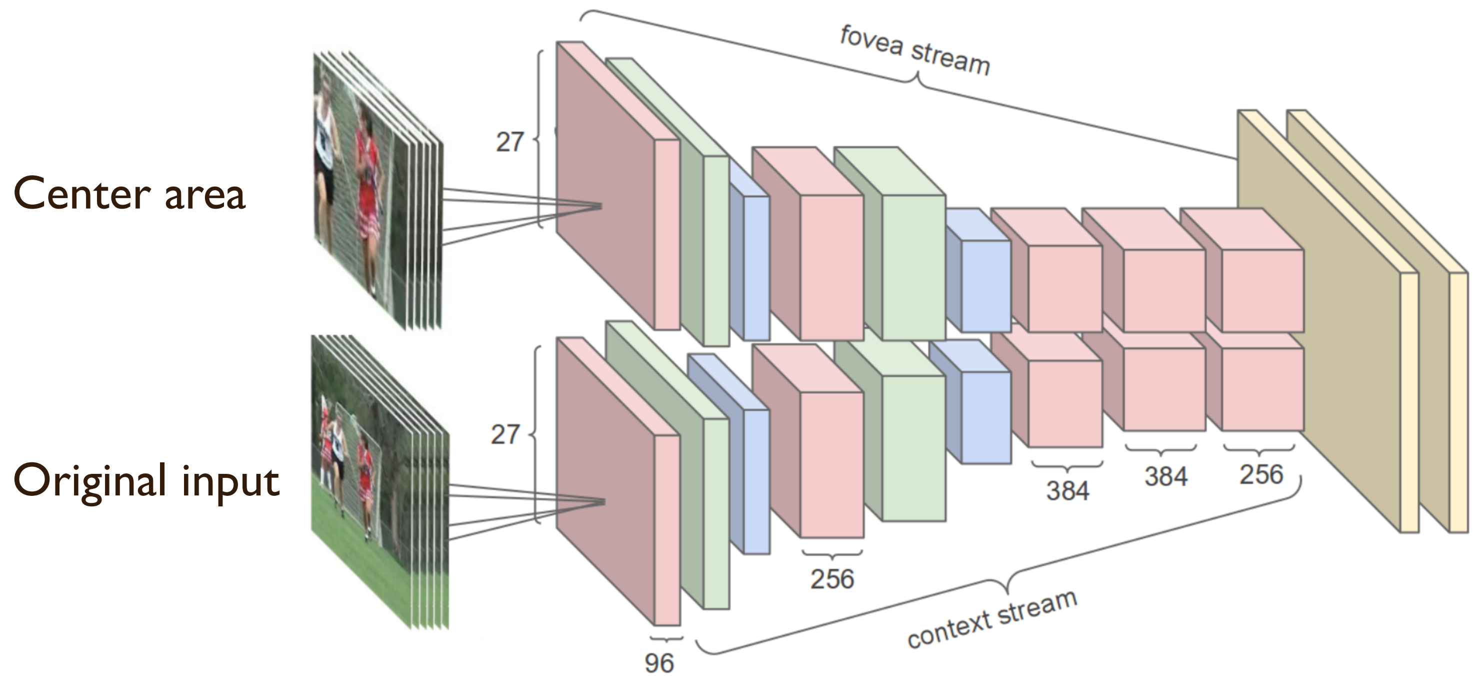
Large-scale Video Classification with Convolutional Neural Networks

Andrej Karpathy[1,2] karpathy@cs.stanford.edu
George Toderici[1] gtoderici@google.com
Sanketh Shetty[1] sanketh@google.com
Thomas Leung[1] leungt@google.com
Rahul Sukthankar[1] sukthankar@google.com
Li Fei-Fei[2] feifeili@cs.stanford.edu
[1]Google Research [2]Computer Science Department, Stanford University
http://cs.stanford.edu/people/karpathy/deepvideo

Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

DeepVideo: Trial #1

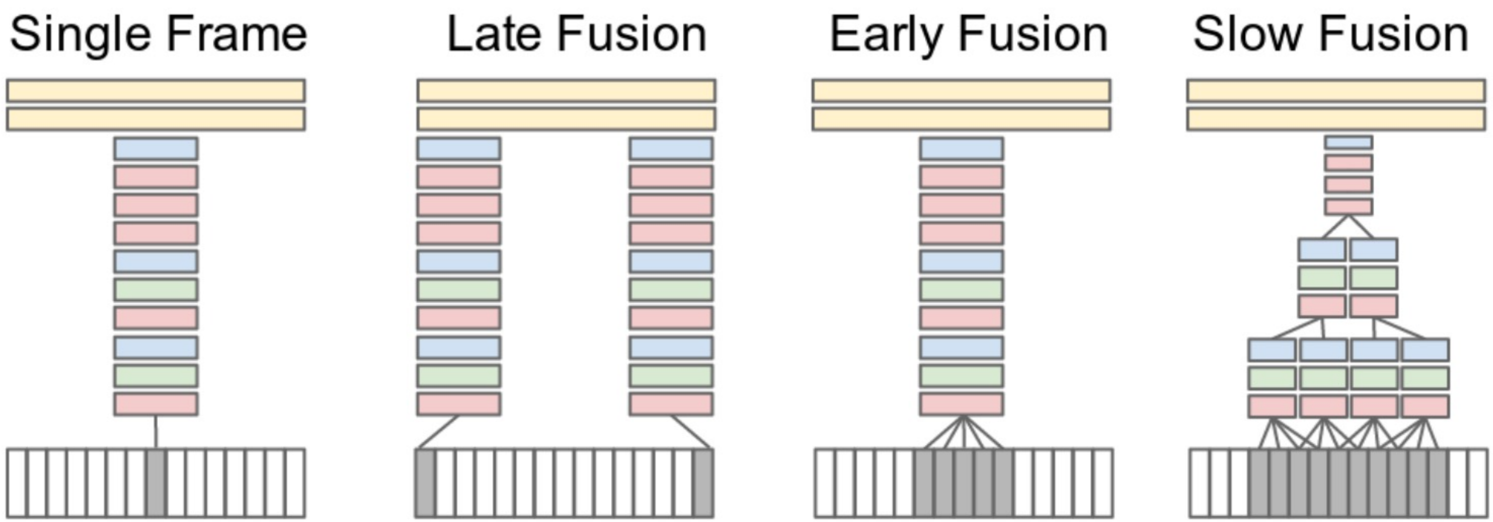


Multiresolution CNN

Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

DeepVideo: Trial #2



random pick one frame
Image classification

two separate single-frame networks with shared parameters a distance of 15 frames apart and then merges the two streams in the first fully connected layer

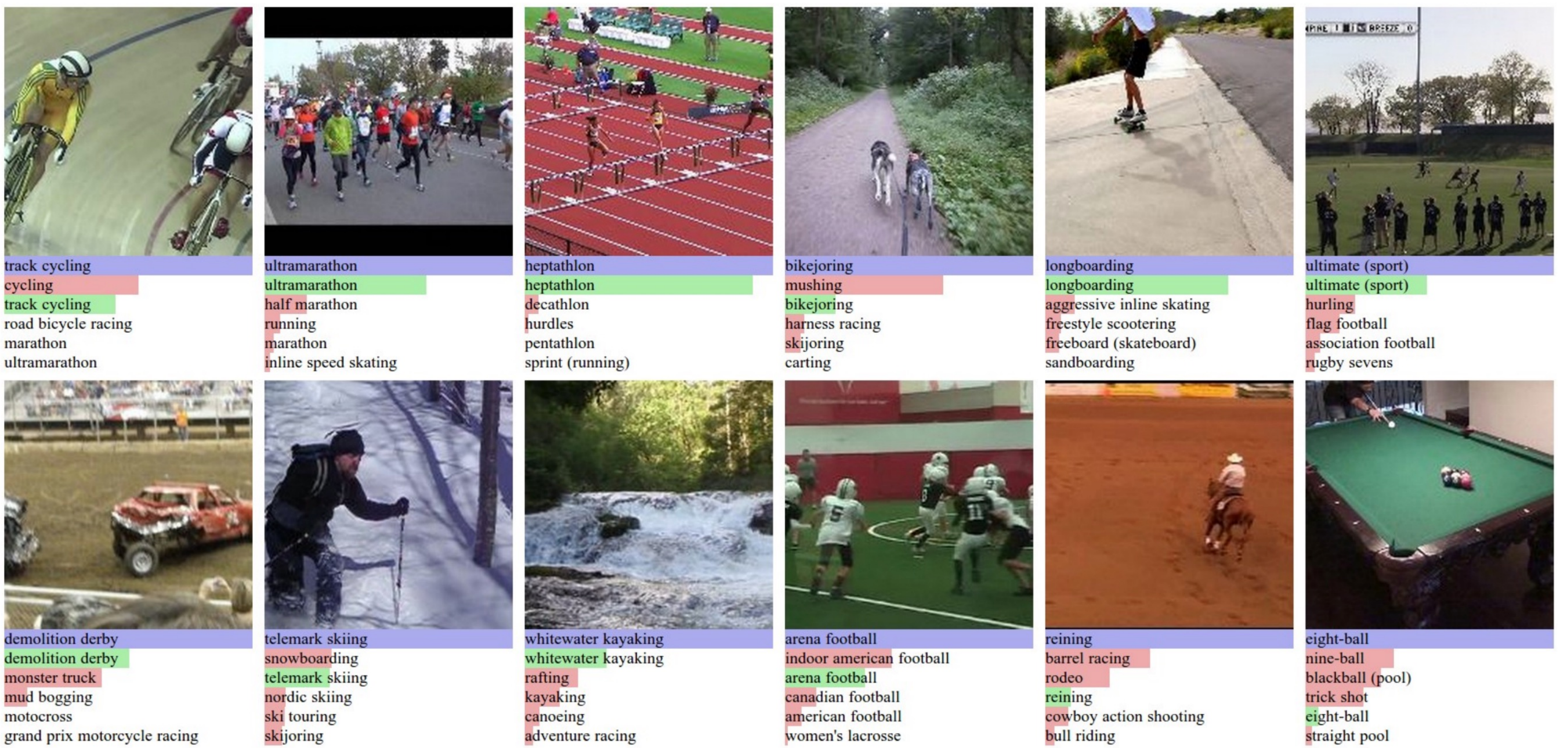
combine information across an entire time window immediately on the pixel level

balanced mix between the two approaches that slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions

Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

Dataset for video action recognition. The Sports-1M dataset consists of 1 million YouTube videos annotated with 487 classes



Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

DeepVideo: Results on the 200,000 videos of the Sports-1M test set.

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
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

Relatively minor improvement

Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

DeepVideo: Results on the 200,000 videos of the Sports-1M test set.

Model	3-fold Accuracy
Soomro et al [22]	43.9%
Feature Histograms + Neural Net	59.0%
Train from scratch	41.3%
Fine-tune top layer	64.1%
Fine-tune top 3 layers	**65.4%**
Fine-tune all layers	62.2%

87%
handcrafted features

Table 3: Results on UCF-101 for various Transfer Learning approaches using the Slow Fusion network.

Pretrain on Sports-1M and transfer learning on UCF-101

Problem: CNN does not work well on video understanding?

Andrej Karpathy, et al, Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

CNN

Comparison with handcrafted features

Table 4: **Mean accuracy (over three splits) on UCF-101 and HMDB-51.**

	Method	UCF-101	HMDB-51
Traditional method	Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
	IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
	IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
CNN	Spatio-temporal HMAX network [11, 16]	-	22.8%
	"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-

the results of CNNs were significantly worse than those of the best hand-crafted shallow representations

Hypothesis: Motion is critical for video understanding

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

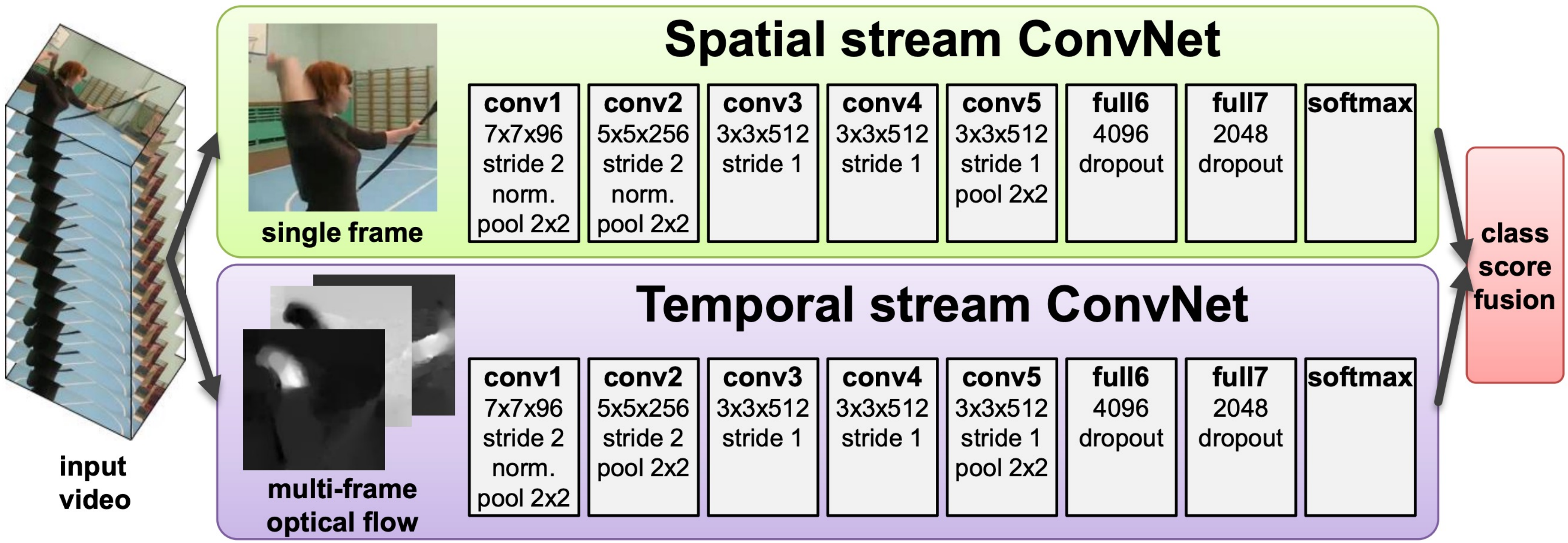
Two-Stream Convolutional Networks for Action Recognition in Videos

Karen Simonyan **Andrew Zisserman**
Visual Geometry Group, University of Oxford
{karen,az}@robots.ox.ac.uk

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Spatial stream performs action recognition from still video frames

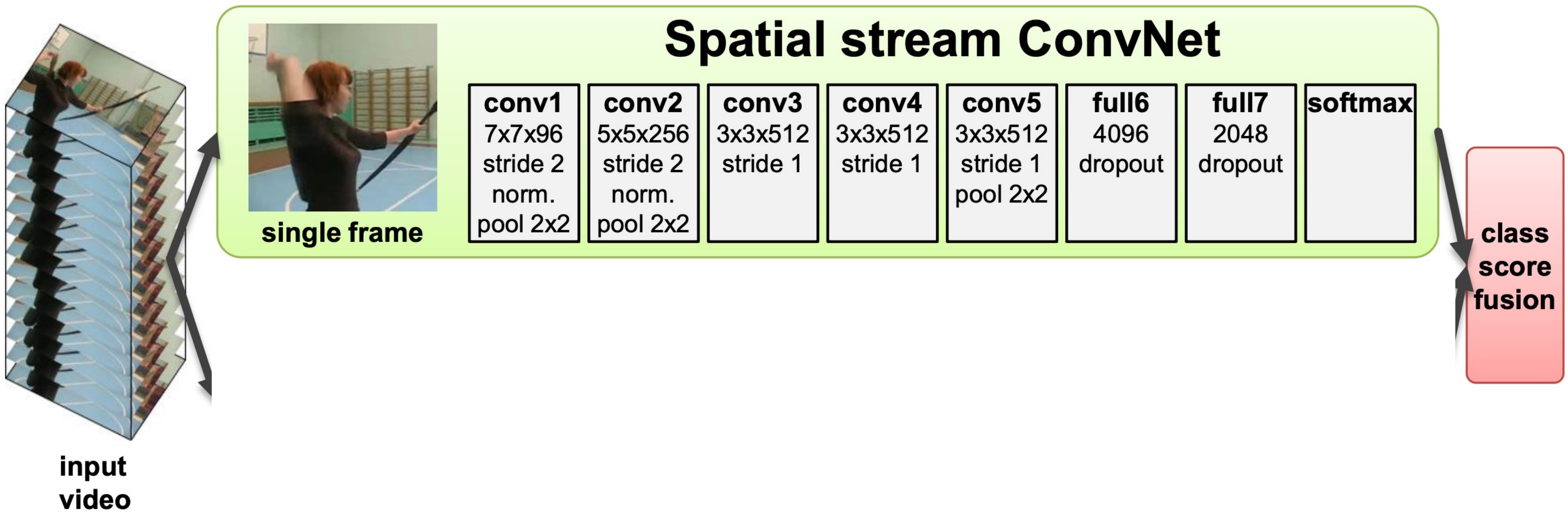


Temporal stream is trained to recognise action from motion in the form of dense optical flow

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Spatial stream performs action recognition from still video frames



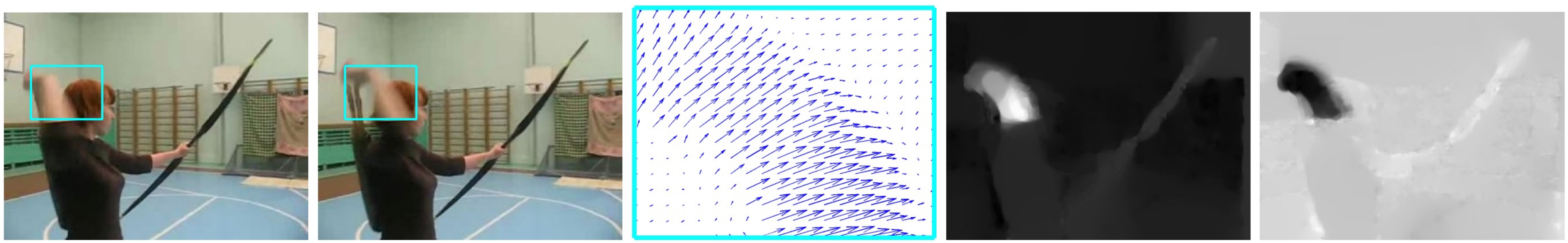
- spatial ConvNet is essentially an image classification architecture, we can build upon the recent advances in large-scale image recognition methods
- pre-train the network on a large image classification dataset, such as the ImageNet challenge dataset

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Temporal-stream CNN

- Input



(a) (b) (c) (d) (e)

Figure 2: **Optical flow.** (a),(b): a pair of consecutive video frames with the area around a moving hand outlined with a cyan rectangle. (c): a close-up of dense optical flow in the outlined area; (d): horizontal component d^x of the displacement vector field (higher intensity corresponds to positive values, lower intensity to negative values). (e): vertical component d^y. Note how (d) and (e) highlight the moving hand and bow. The input to a ConvNet contains multiple flows (Sect. 3.1).

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Temporal-stream CNN

- Input
  - Bi-directional optical flow
  - Mean flow subtraction

Disadvantages of optical flow:

i) too slow

ii) too large

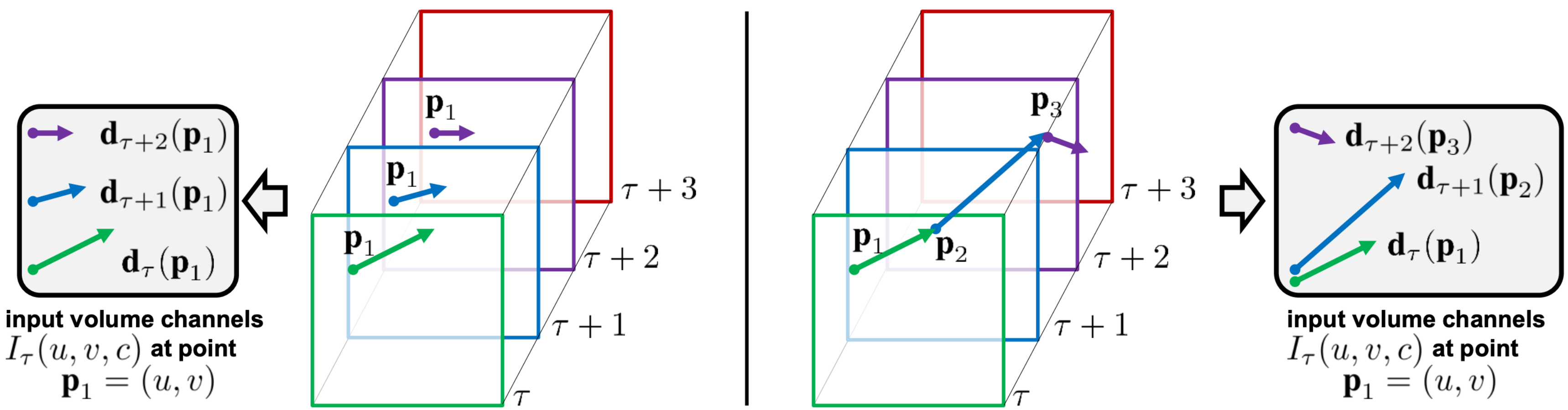
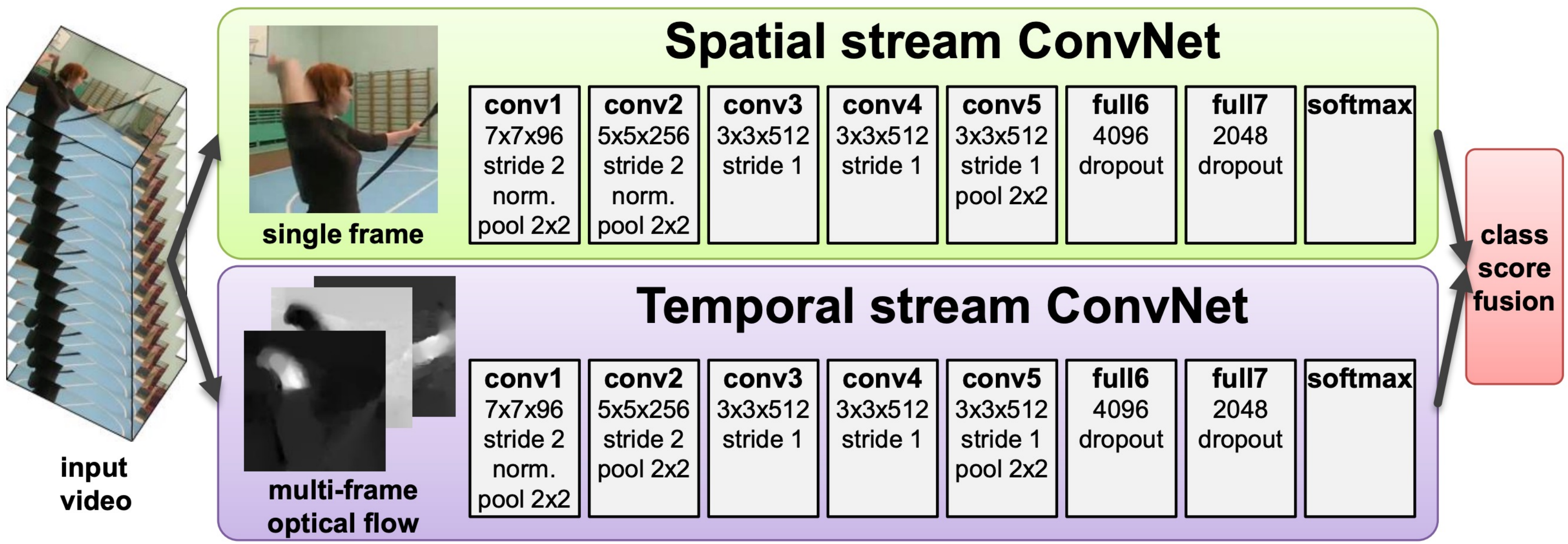


Figure 3: **ConvNet input derivation from the multi-frame optical flow.** *Left:* optical flow stacking (1) samples the displacement vectors **d** at the same location in multiple frames. *Right:* trajectory stacking (2) samples the vectors along the trajectory. The frames and the corresponding displacement vectors are shown with the same colour.

Two-stream CNN

- Pretraining on on ImageNet ILSVRC-2012
- Multi-task learning: Combining several datasets



Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Two-stream CNN

Table 4: **Mean accuracy (over three splits) on UCF-101 and HMDB-51.**

	Method	UCF-101	HMDB-51
Traditional method	Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
	IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
	IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
CNN	Spatio-temporal HMAX network [11, 16]	-	22.8%
	"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-
Two-stream CNN	Spatial stream ConvNet	73.0%	40.5%
	Temporal stream ConvNet	83.7%	54.6%
	Two-stream model (fusion by averaging)	86.9%	58.0%
	Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

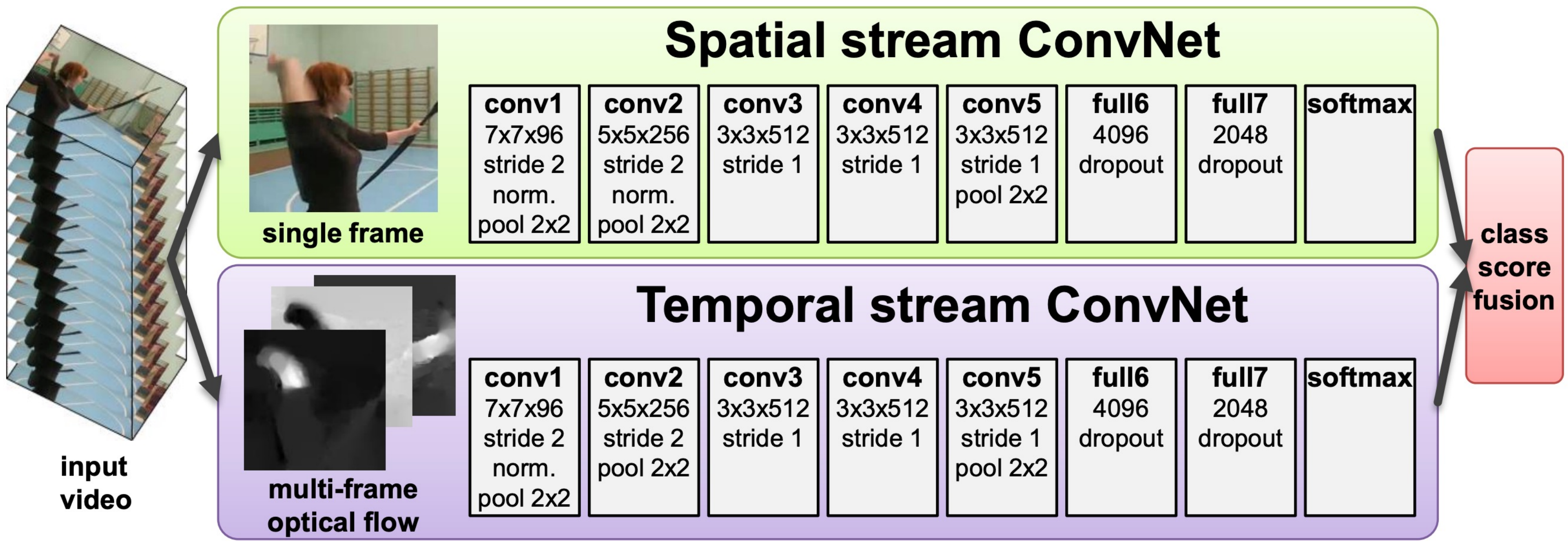
Table 2: **Temporal ConvNet accuracy on HMDB-51 (split 1 with additional training data).**

Training setting	Accuracy
Training on HMDB-51 without additional data	46.6%
Fine-tuning a ConvNet, pre-trained on UCF-101	49.0%
Training on HMDB-51 with classes added from UCF-101	52.8%
Multi-task learning on HMDB-51 and UCF-101	**55.4%**

Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NeurIPS 2014

Two-stream CNN

Problem: Only handle a short video, how to deal with a long video?



Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

Two-stream CNN

Problem: Only handle a short video, how to deal with a long video?

Solution #1: Pooling or LSTM

Beyond Short Snippets: Deep Networks for Video Classification

Joe Yue-Hei Ng[1] yhng@umiacs.umd.edu
Matthew Hausknecht[2] mhauskn@cs.utexas.edu
Sudheendra Vijayanarasimhan[3] svnaras@google.com
Oriol Vinyals[3] vinyals@google.com
Rajat Monga[3] rajatmonga@google.com
George Toderici[3] gtoderici@google.com

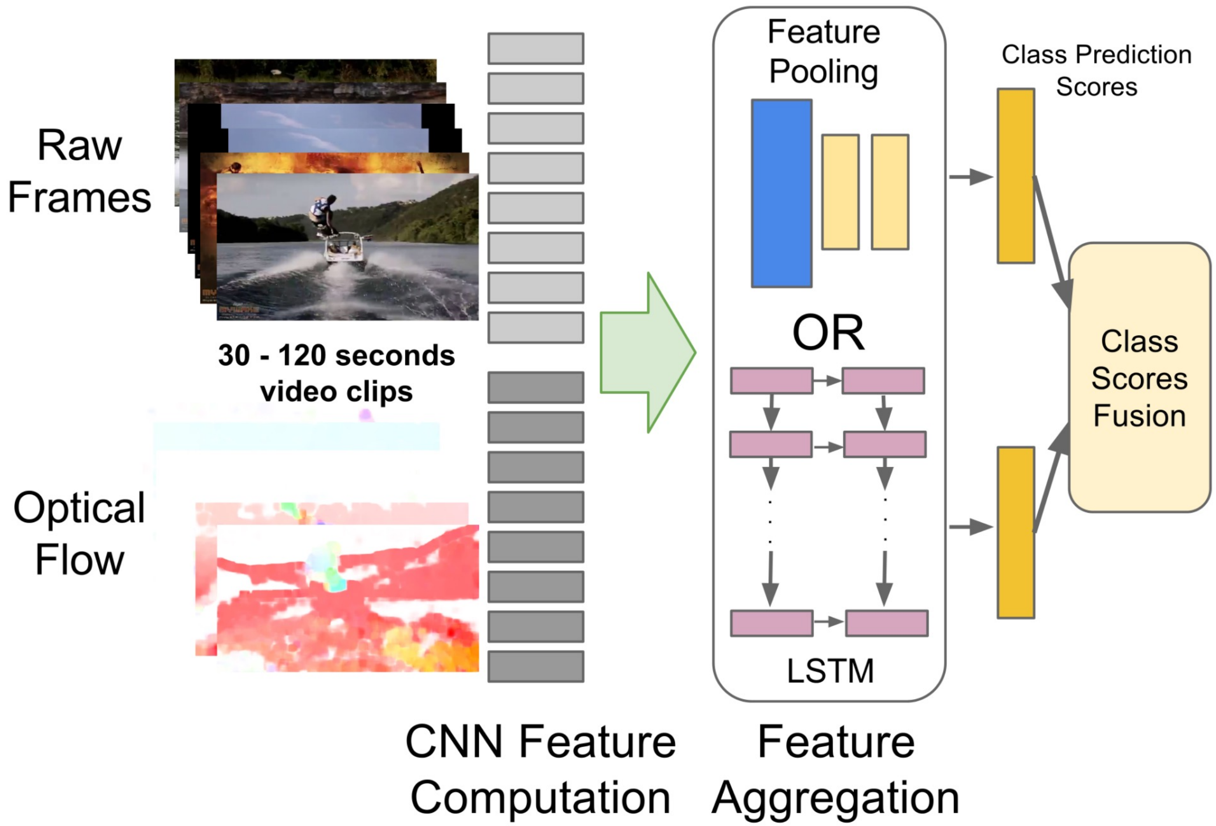
[1]University of Maryland, College Park [2]University of Texas at Austin [3]Google, Inc.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

Two-stream CNN

Problem: Only handle a short video, how to deal with a long video?

Solution #1: Pooling or LSTM

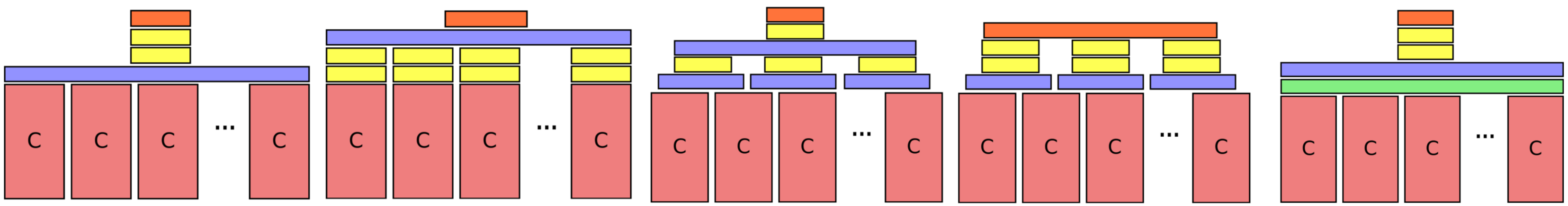


Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

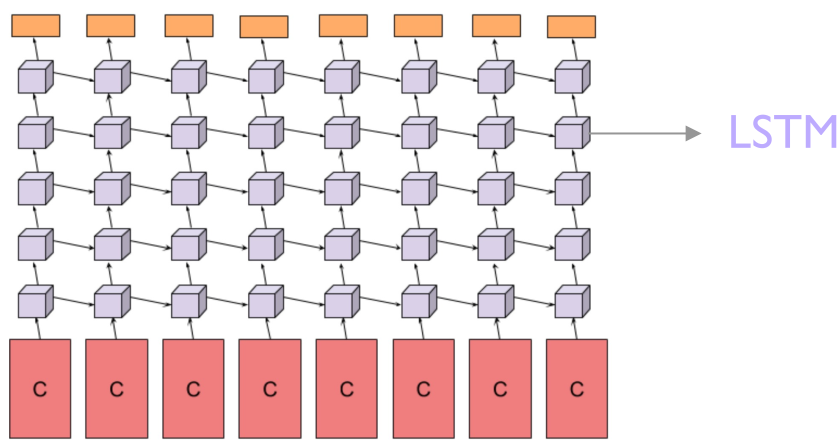
Two-stream CNN

Pooling

max-pooling
time-domain convolutional
fully-connected
softmax layers



(a) Conv Pooling (b) Late Pooling (c) Slow Pooling (d) Local Pooling (e) Time-Domain Convolution



Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

Two-stream CNN

Pooling: Experimental results on Sports-1M

Category	Method	Frames	Clip Hit@1	Hit@1	Hit@5
Prior Results [14]	Single Frame	1	41.1	59.3	77.7
	Slow Fusion	15	41.9	60.9	80.2
Conv Pooling	Image and Optical Flow	120	**70.8**	72.4	**90.8**
LSTM	Image and Optical Flow	30	N/A	**73.1**	90.5

Table 5: Leveraging global video-level descriptors, LSTM and Conv-Pooling achieve a 20% increase in Hit@1 compared to prior work on the in Sports-1M dataset. Hit@1, and Hit@5 are computed at video level.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

Two-stream CNN

Experimental results on UCF 101

Method	3-fold Accuracy (%)
Improved Dense Trajectories (IDTF)s [23]	87.9
Slow Fusion CNN [14]	65.4
Single Frame CNN Model (Images) [19]	73.0
Single Frame CNN Model (Optical Flow) [19]	73.9
Two-Stream CNN (Optical Flow + Image Frames, Averaging) [19]	86.9
Two-Stream CNN (Optical Flow + Image Frames, SVM Fusion) [19]	88.0
Our Single Frame Model	73.3
Conv Pooling of Image Frames + Optical Flow (30 Frames)	87.6
Conv Pooling of Image Frames + Optical Flow (120 Frames)	**88.2**
LSTM with 30 Frame Unroll (Optical Flow + Image Frames)	**88.6**

Table 7: UCF-101 results. The bold-face numbers represent results that are higher than previously reported results.

Minor improvement

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

Two-stream CNN

Problem: Only handle a short video, how to deal with a long video?

Solution #2: Temporal segment

Temporal Segment Networks: Towards Good Practices for Deep Action Recognition

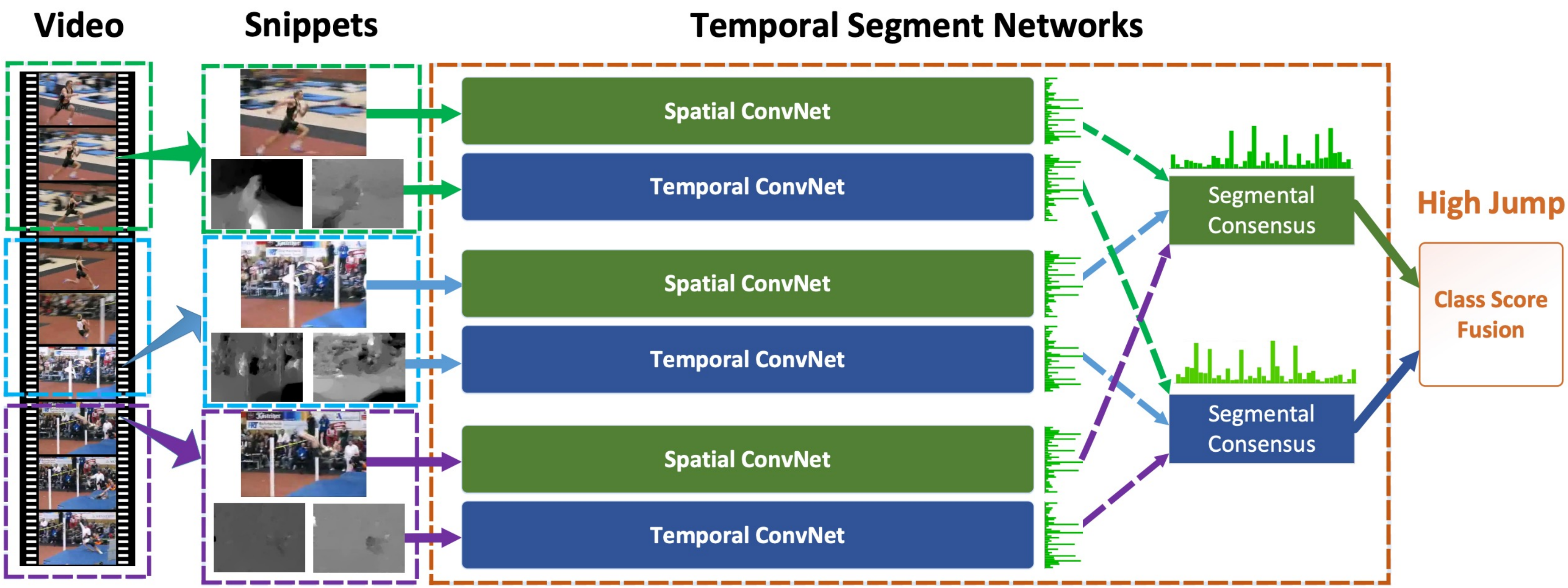
Limin Wang[1], Yuanjun Xiong[2], Zhe Wang[3], Yu Qiao[3], Dahua Lin[2], Xiaoou Tang[2], and Luc Van Gool[1]

[1]Computer Vision Lab, ETH Zurich, Switzerland
[2]Department of Information Engineering, The Chinese University of Hong Kong
[3]Shenzhen Institutes of Advanced Technology, CAS, China

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, ECCV 2016

Two-stream CNN

Temporal Segment Network (TSN): Long video to multiple snippets



Share parameters

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, ECCV 2016

Two-stream CNN

Temporal Segment Network (TSN): Long video to multiple snippets

HMDB51		UCF101	
DT+MVSV [37]	55.9%	DT+MVSV [37]	83.5%
iDT+FV [2]	57.2%	iDT+FV [38]	85.9%
iDT+HSV [25]	61.1%	iDT+HSV [25]	87.9%
MoFAP [39]	61.7%	MoFAP [39]	88.3%
Two Stream [1]	59.4%	Two Stream [1]	88.0%
VideoDarwin [18]	63.7%	C3D (3 nets) [13]	85.2%
MPR [40]	65.5%	Two stream +LSTM [4]	88.6%
$\mathrm{F_{ST}CN}$ (SCI fusion) [28]	59.1%	$\mathrm{F_{ST}CN}$ (SCI fusion) [28]	88.1%
TDD+FV [5]	63.2%	TDD+FV [5]	90.3%
LTC [19]	64.8%	LTC [19]	91.7%
KVMF [41]	63.3%	KVMF [41]	93.1%
TSN (2 modalities)	68.5%	TSN (2 modalities)	94.0%
TSN (3 modalities)	**69.4**%	TSN (3 modalities)	**94.2**%

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, ECCV 2016

Two-stream CNN

Convolutional Two-Stream Network $\boxed{\text{Fusion}}$ for Video Action Recognition

Christoph Feichtenhofer
Graz University of Technology
feichtenhofer@tugraz.at

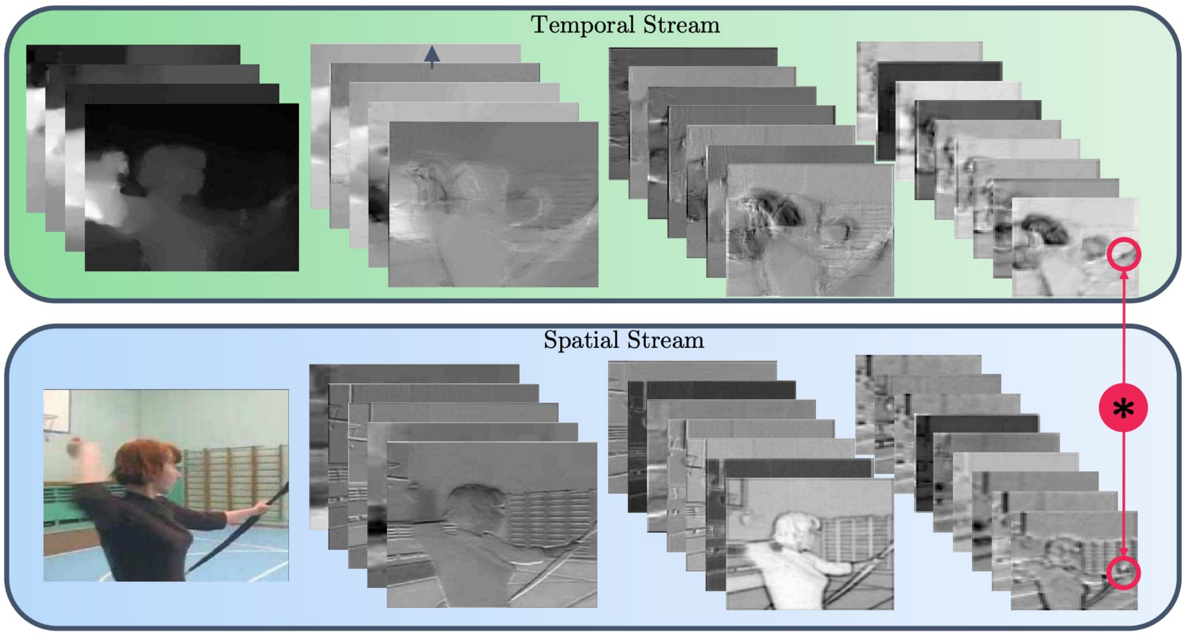
Axel Pinz
Graz University of Technology
axel.pinz@tugraz.at

Andrew Zisserman
University of Oxford
az@robots.ox.ac.uk

Two-stream CNN

All about fusion

- How to fuse in the spatial dimension?
- How to fuse in the temporal dimension?
- Which layer to fuse?



Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

Spatial fusion

- Sum fusion $$\mathbf{y}^{\text{sum}} = f^{\text{sum}}(\mathbf{x}^a, \mathbf{x}^b) \qquad y^{\text{sum}}_{i,j,d} = x^a_{i,j,d} + x^b_{i,j,d}$$
- Max fusion $$\mathbf{y}^{\max} = f^{\max}(\mathbf{x}^a, \mathbf{x}^b) \qquad y^{\max}_{i,j,d} = \max\{x^a_{i,j,d}, x^b_{i,j,d}\}$$
- Concatenation fusion $$\mathbf{y}^{\text{cat}} = f^{\text{cat}}(\mathbf{x}^a, \mathbf{x}^b) \qquad y^{\text{cat}}_{i,j,2d} = x^a_{i,j,d} \qquad y^{\text{cat}}_{i,j,2d-1} = x^b_{i,j,d}$$
- Conv fusion $$\mathbf{y}^{\text{conv}} = f^{\text{conv}}(\mathbf{x}^a, \mathbf{x}^b) \qquad \mathbf{y}^{\text{conv}} = \mathbf{y}^{\text{cat}} * \mathbf{f} + b$$
- Bilinear fusion $$\mathbf{y}^{\text{bil}} = f^{\text{bil}}(\mathbf{x}^a, \mathbf{x}^b) = \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{x}^{a\top}_{i,j} \otimes \mathbf{x}^b_{i,j} \in \mathbb{R}^{D^2}$$

Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

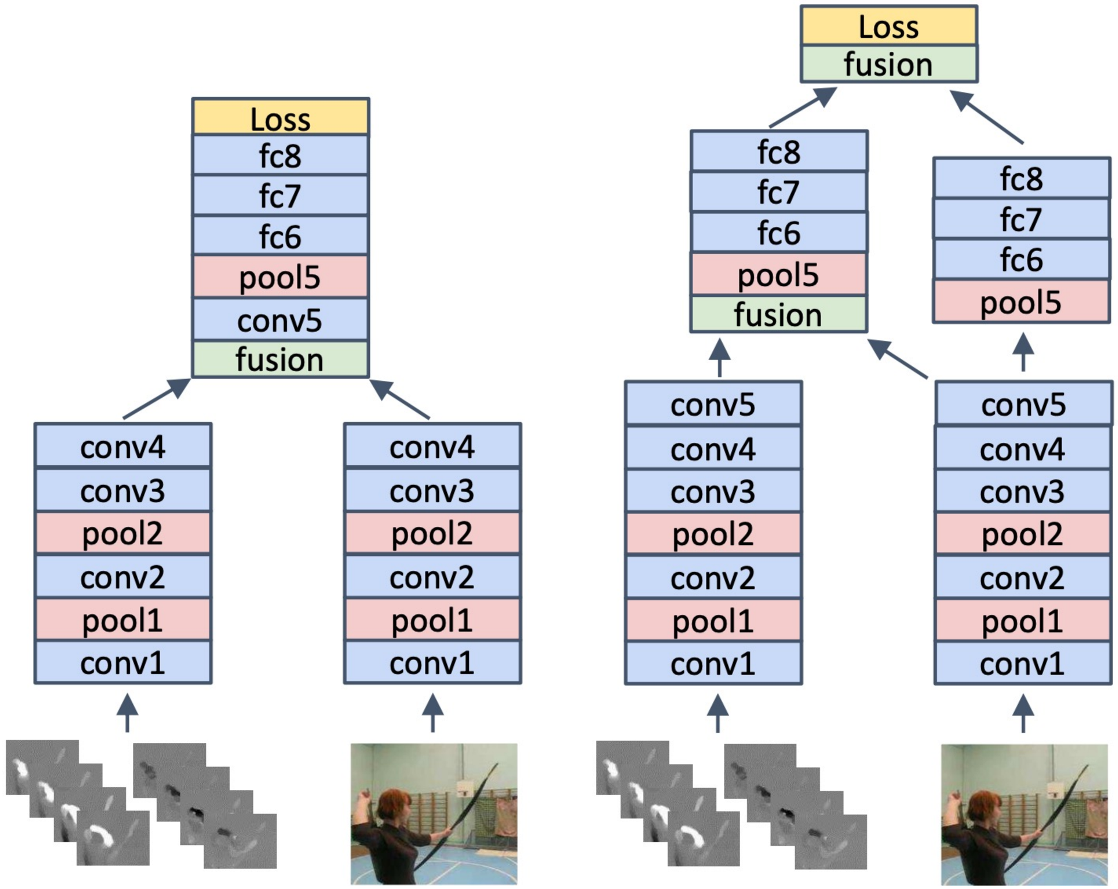
Spatial fusion: Results

Fusion Method	Fusion Layer	Acc.	#layers	#parameters
Sum [22]	Softmax	85.6%	16	181.42M
Sum (ours)	Softmax	85.94%	16	181.42M
Max	ReLU5	82.70%	13	97.31M
Concatenation	ReLU5	83.53%	13	172.81M
Bilinear [15]	ReLU5	85.05%	10	6.61M+SVM
Sum	ReLU5	85.20%	13	97.31M
Conv	ReLU5	85.96%	14	97.58M

Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

Where to fuse



Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

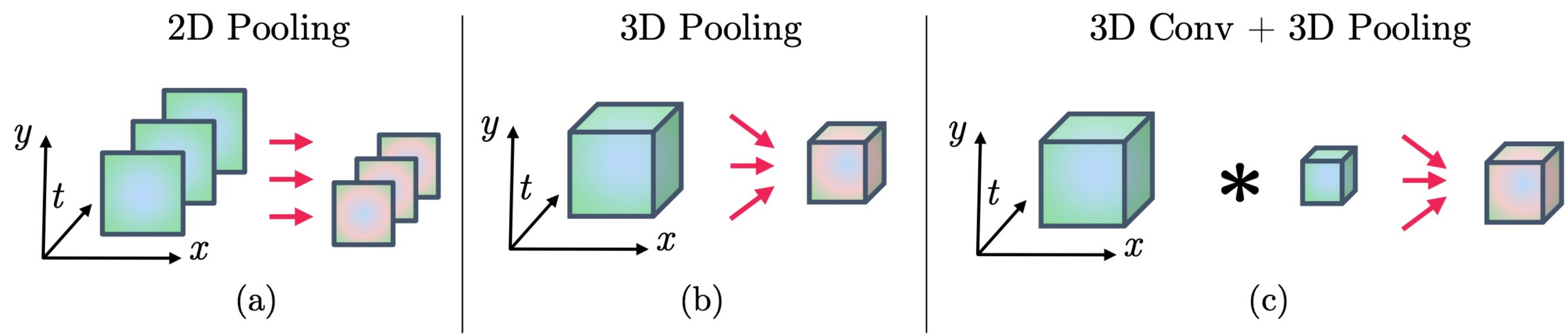
Where to fuse: Results

Fusion Layers	Accuracy	#layers	#parameters
ReLU2	82.25%	11	91.90M
ReLU3	83.43%	12	93.08M
ReLU4	82.55%	13	95.48M
ReLU5	85.96%	14	97.57M
ReLU5 + FC8	86.04%	17	181,68M
ReLU3 + ReLU5 + FC6	81.55%	17	190,06M

Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

Temporal fusion



2D pooling ignores time and simply pools over spatial neighbourhoods to individually shrink the size of the feature maps for each temporal sample

3D pooling pools from local spatiotemporal neighbourhoods by first stacking the feature maps across time and then shrinking this spatiotemporal cube

3D conv + 3D pooling additionally performs a convolution with a fusion kernel that spans the feature channels, space and time before 3D pooling

Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

Temporal fusion: Results

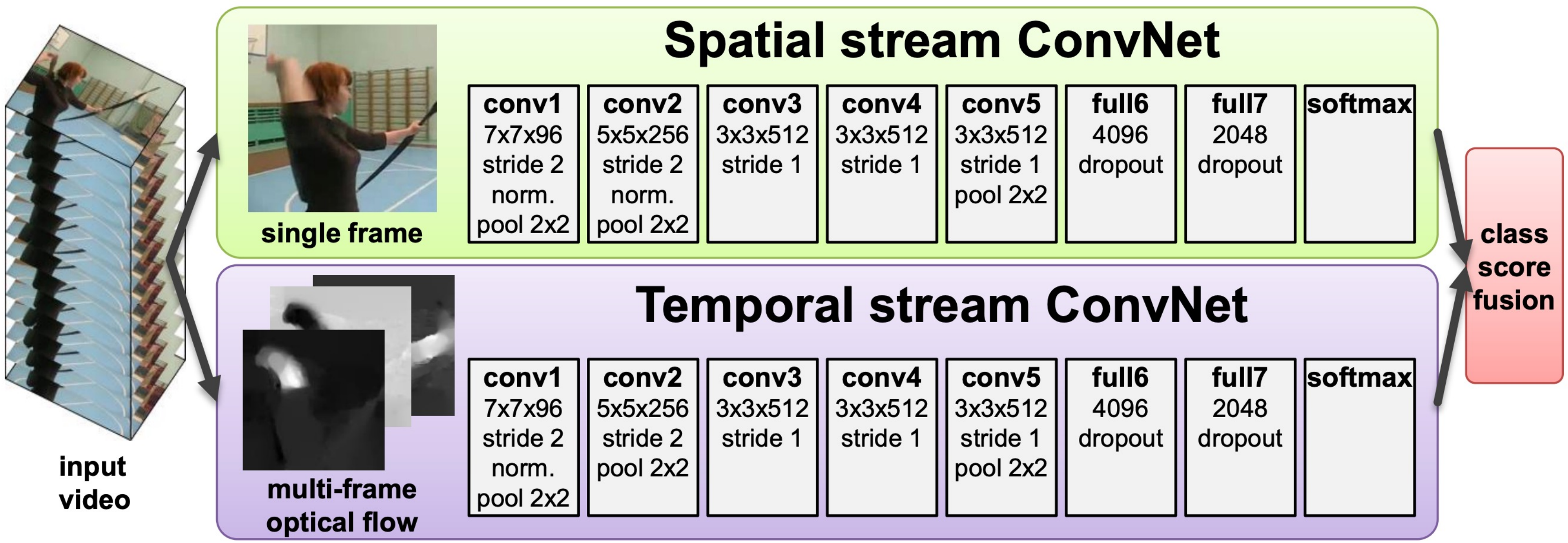
Fusion Method	Pooling	Fusion Layers	UCF101	HMDB51
2D Conv	2D	ReLU5 +	89.35%	56.93%
2D Conv	3D	ReLU5 +	89.64%	57.58%
3D Conv	3D	ReLU5 +	90.40%	58.63%

Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional Two-Stream Network Fusion for Video Action Recognition, CVPR 2016

Two-stream CNN

Problem: Precompute optical flow   Slow and expensive to store!

0.06s per frame	UCF 101	10K videos	6s	30 Hz	1.5 day
	K400	240K videos	10s	30 Hz	50 day



Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici, Beyond Short Snippets: Deep Networks for Video Classification, CVPR 2015

3D CNN

C3D

Learning Spatiotemporal Features with 3D Convolutional Networks

Du Tran[1,2], Lubomir Bourdev[1], Rob Fergus[1], Lorenzo Torresani[2], Manohar Paluri[1]
[1]Facebook AI Research, [2]Dartmouth College
{dutran,lorenzo}@cs.dartmouth.edu {lubomir,robfergus,mano}@fb.com

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks, ICCV 2015

3D CNN

C3D

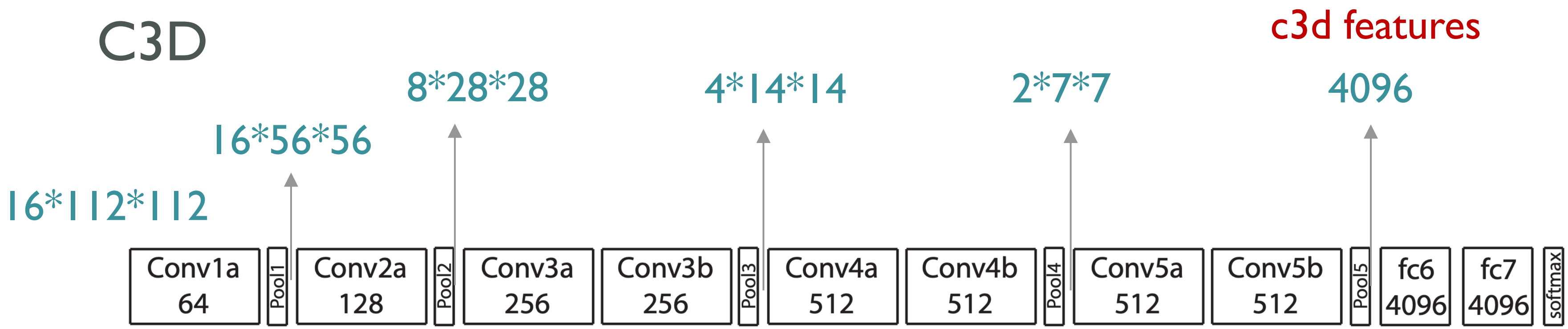


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling kernels are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

VGG16

Training for one month

Method	Number of Nets	Clip hit@1	Video hit@1	Video hit@5
DeepVideo's Single-Frame + Multires [18]	3 nets	42.4	60.0	78.5
DeepVideo's Slow Fusion [18]	1 net	41.9	60.9	80.2
Convolution pooling on 120-frame clips [29]	3 net	**70.8***	**72.4**	**90.8**
C3D (trained from scratch)	1 net	44.9	60.0	84.4
C3D (fine-tuned from I380K pre-trained model)	1 net	46.1	61.1	85.2

Table 2. **Sports-1M classification result**. C3D outperforms [18] by 5% on top-5 video-level accuracy. (*)We note that the method of [29] uses long clips, thus its clip-level accuracy is not directly comparable to that of C3D and DeepVideo.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks, ICCV 2015

3D CNN

Two-stream Inflated 3D ConvNet (I3D)

Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

João Carreira[†] Andrew Zisserman[†,*]
joaoluis@google.com zisserman@google.com
[†]DeepMind [*]Department of Engineering Science, University of Oxford

Abstract

The paucity of videos in current action classification datasets (UCF-101 and HMDB-51) has made it difficult to identify good video architectures, as most methods obtain similar performance on existing small-scale benchmarks. This paper re-evaluates state-of-the-art architectures in light of the new Kinetics Human Action Video dataset. Kinetics has two orders of magnitude more data, with 400 human action classes and over 400 clips per class, and is collected from realistic, challenging YouTube videos. We provide an analysis on how current architectures fare on the task of action classification on this dataset and how much performance improves on the smaller benchmark datasets after pre-training on Kinetics.

We also introduce a new Two-Stream Inflated 3D ConvNet (I3D) that is based on 2D ConvNet inflation: filters and pooling kernels of very deep image classification ConvNets are expanded into 3D, making it possible

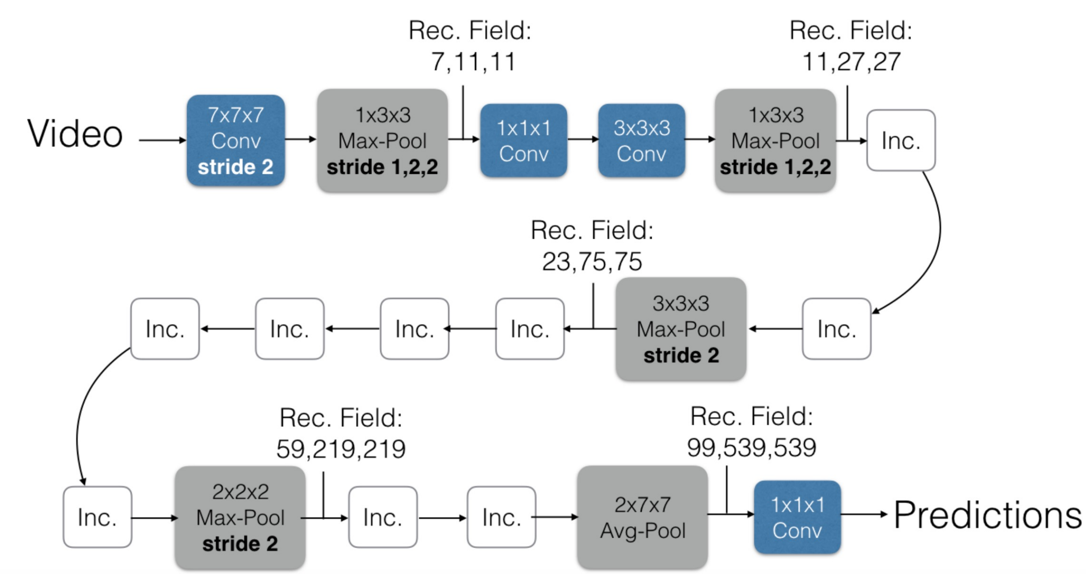
Figure 1. **A still from 'Quo Vadis' (1951).** Where is this going? Are these actors about to kiss each other, or have they just done so? More importantly, where is action recognition going? Actions can be ambiguous in individual frames, but the limitations of exist-

Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, CVPR 2017

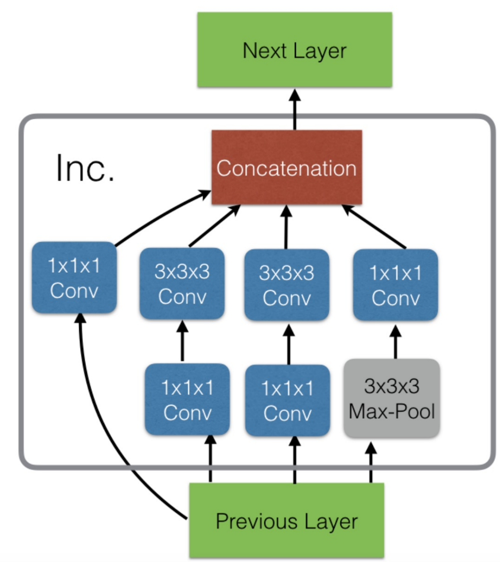
3D CNN

Two-stream inflated 3D ConvNet (I3D)

Inflated Inception-V1



Inception Module (Inc.)



Inflating 2D ConvNets into 3D

This can be done by starting with a 2D architecture, and inflating all the filters and pooling kernels – endowing them with an additional temporal dimension. Filters are typically square and we just make them cubic – N × N filters become N × N × N.

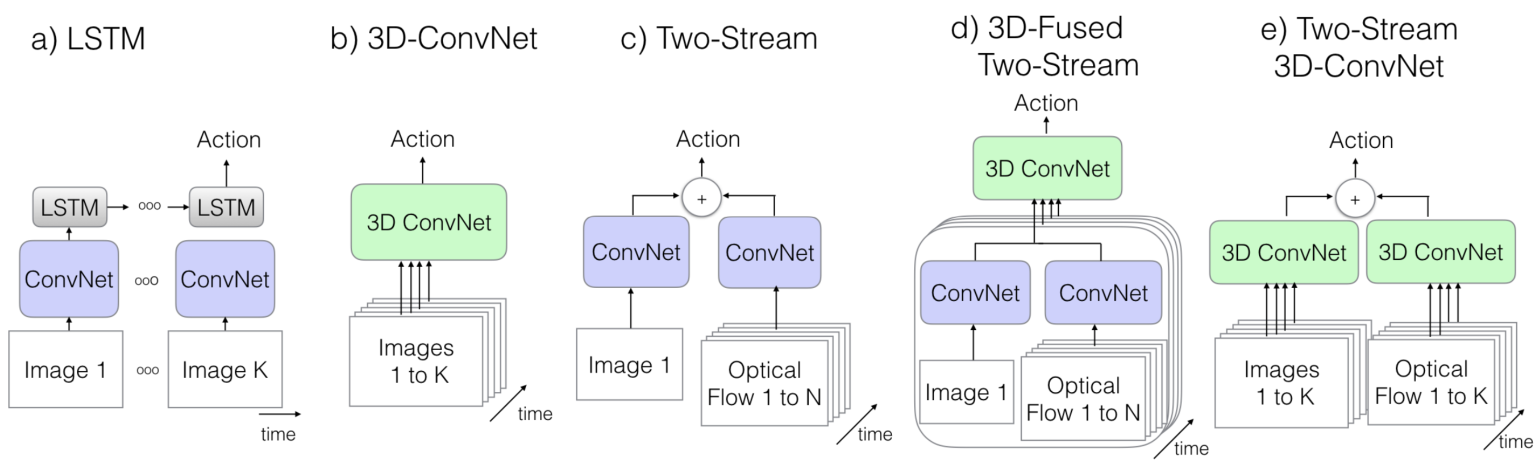
Bootstrapping 3D filters from 2D Filters

repeat the weights of the 2D filters N times along the time dimension, and rescale them by dividing by N

Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, CVPR 2017

3D CNN

Two-stream inflated 3D ConvNet (I3D)



Architecture	UCF-101			HMDB-51			miniKinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	69.9	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	60.0	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	70.1	58.4	72.9
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	71.4	61.0	74.0
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**74.1**	**69.6**	**78.7**

Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, CVPR 2017

3D CNN

Two-stream inflated 3D ConvNet (I3D)

Model	UCF-101	HMDB-51
Two-Stream [25]	88.0	59.4
IDT [30]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [31]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [32]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [29], Sports 1M pre-training	82.3	-
C3D ensemble [29], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [29], Sports 1M pre-training	90.1	-
RGB-I3D, miniKinetics pre-training	91.8	66.4
RGB-I3D, Kinetics pre-training	95.4	74.5
Flow-I3D, miniKinetics pre-training	94.7	72.4
Flow-I3D, Kinetics pre-training	95.4	74.6
Two-Stream I3D, miniKinetics pre-training	96.9	76.3
Two-Stream I3D, Kinetics pre-training	**97.9**	**80.2**

Table 4. Comparison with state-of-the-art on the UCF-101 and HMDB-51 datasets, averaged over three splits. First set of rows contains results of models trained without labeled external data.

Terminate UCF-101 & HMDB-51

Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, CVPR 2017

3D CNN

Two-stream inflated 3D ConvNet (I3D)

InceptionNet	→	3D InceptionNet
ResNet	→	ResNet3D
ResNext	→	MFNet
SENet	→	STCNet

Joao Carreira, Andrew Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, CVPR 2017

3D CNN

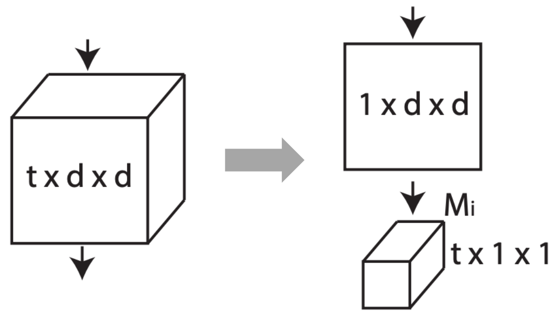
R(2+1)D：

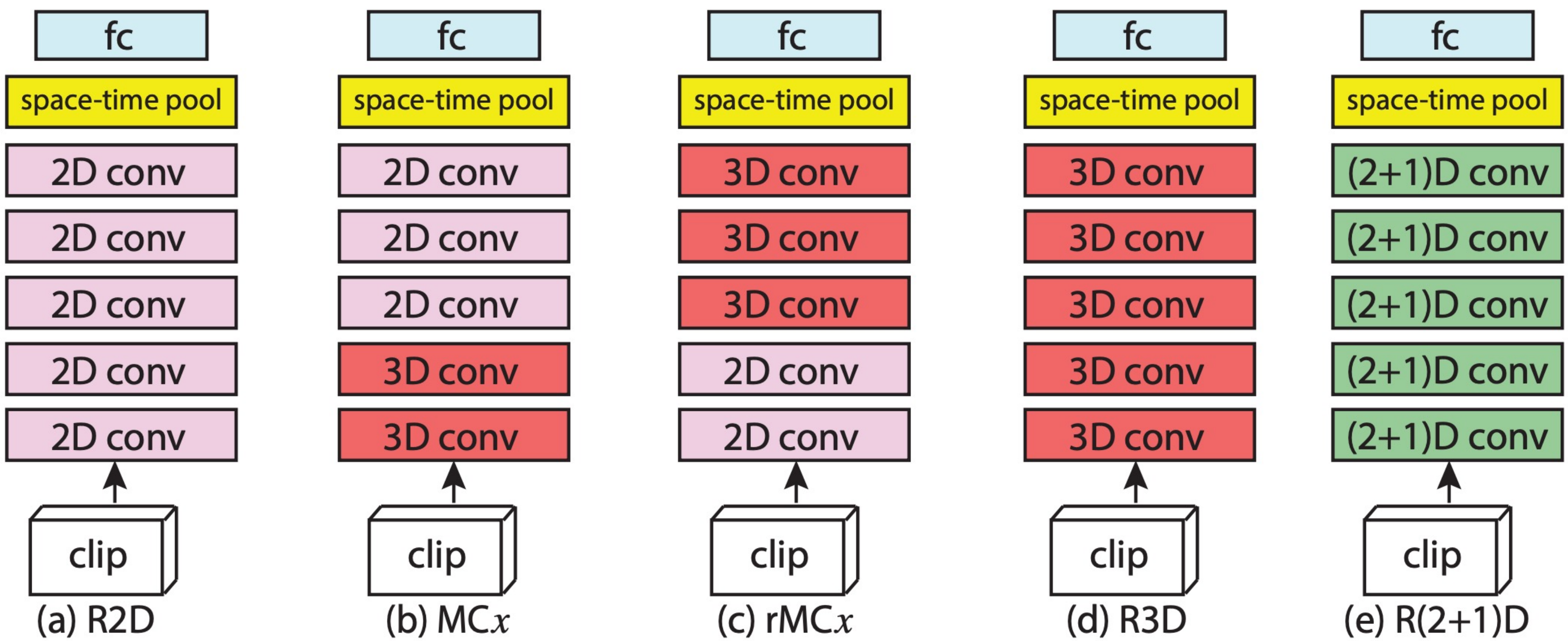
A Closer Look at Spatiotemporal Convolutions for Action Recognition

Du Tran[1], Heng Wang[1], Lorenzo Torresani[1,2], Jamie Ray[1], Yann LeCun[1], Manohar Paluri[1]
[1]Facebook Research [2]Dartmouth College
{trandu,hengwang,torresani,jamieray,yann,mano}@fb.com

3D CNN

R(2+1)D：





Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, Manohar Paluri

3D CNN

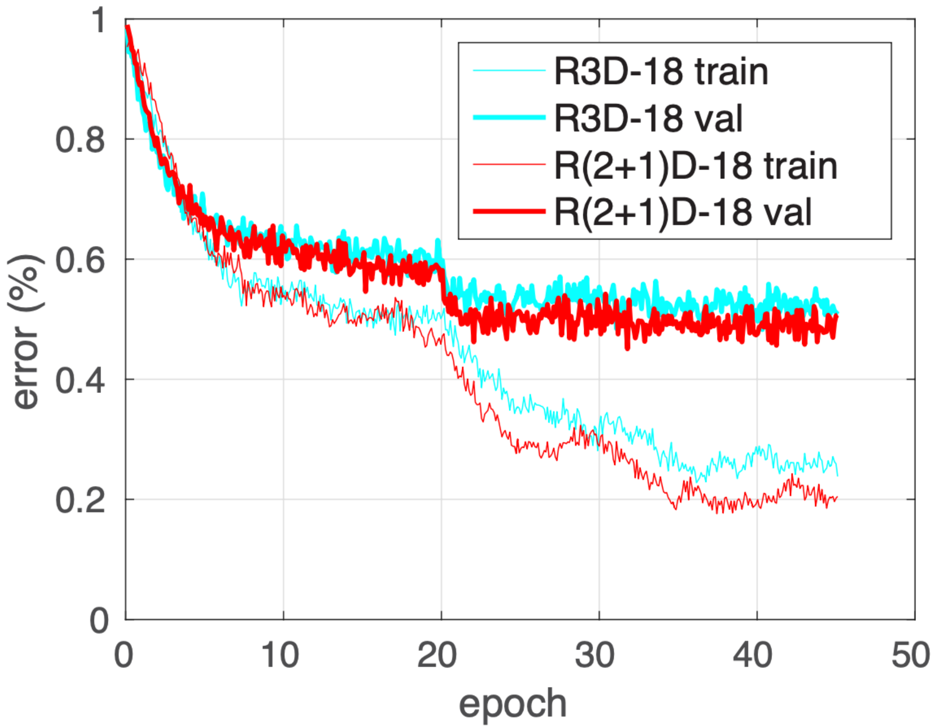
R(2+1)D：

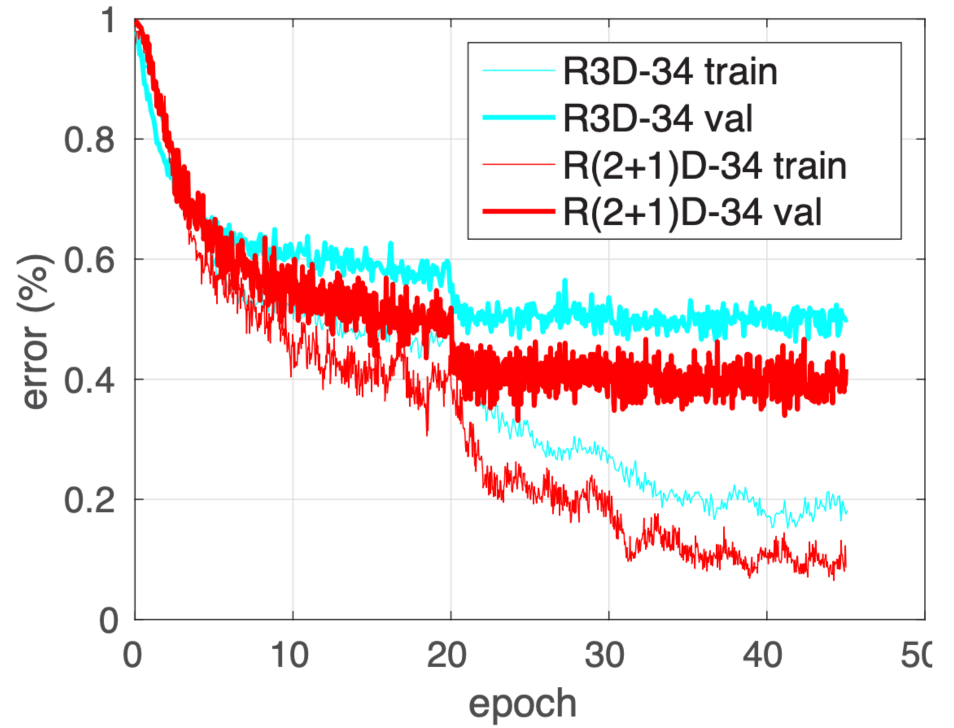
Net	# params	Clip@1	Video@1	Clip@1	Video@1
Input		$8\times112\times112$		$16\times112\times112$	
R2D	11.4M	46.7	59.5	47.0	58.9
f-R2D	11.4M	48.1	59.4	50.3	60.5
R3D	33.4M	49.4	61.8	52.5	64.2
MC2	11.4M	50.2	62.5	53.1	64.2
MC3	11.7M	50.7	62.9	53.7	64.7
MC4	12.7M	50.5	62.5	53.7	65.1
MC5	16.9M	50.3	62.5	53.7	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
R(2+1)D	33.3M	**52.8**	**64.8**	**56.8**	**68.0**

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, Manohar Paluri

3D CNN

R(2+1)D :





Easier to optimize

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, Manohar Paluri

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

Non-local Neural Networks

Xiaolong Wang[1,2*] Ross Girshick[2] Abhinav Gupta[1] Kaiming He[2]
[1]Carnegie Mellon University [2]Facebook AI Research

Abstract

Both convolutional and recurrent operations are building blocks that process one local neighborhood at a time. In this paper, we present non-local operations as a generic family of building blocks for capturing long-range dependencies. Inspired by the classical non-local means method [4] in computer vision, our non-local operation computes the response at a position as a weighted sum of the features at all positions. This building block can be plugged into many computer vision architectures. On the task of video classification, even without any bells and whistles, our non-local models can compete or outperform current competition

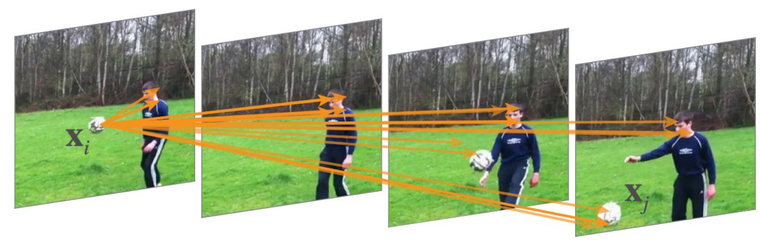
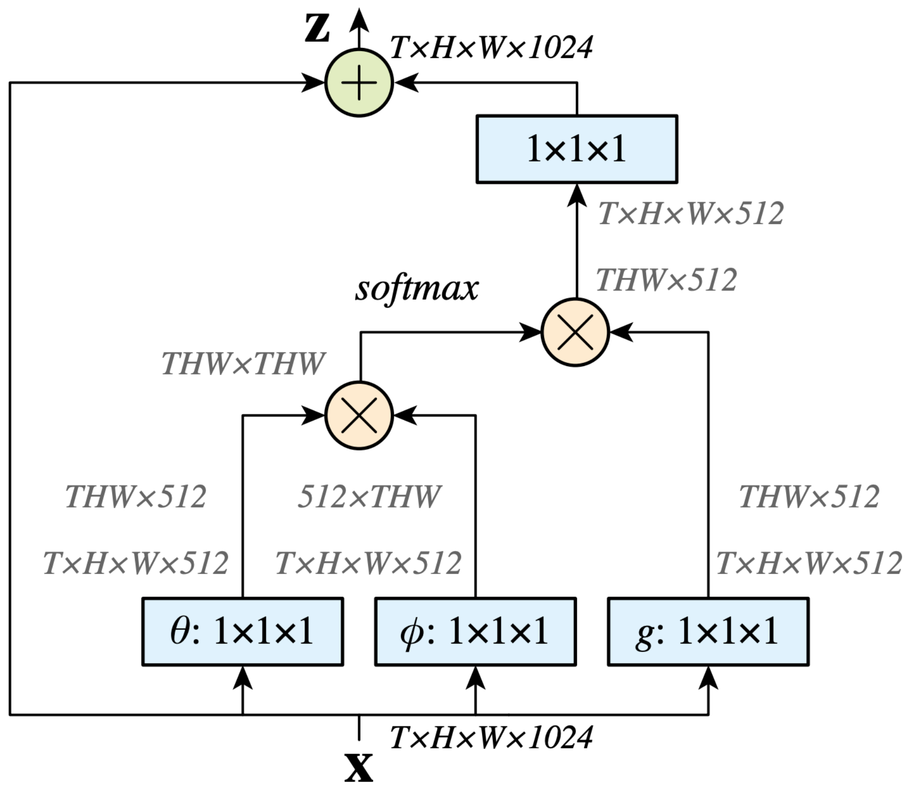


Figure 1. A spacetime ***non-local*** operation in our network trained for video classification in Kinetics. A position $\mathbf{x}_i$'s response is computed by the weighted average of the features of *all* positions $\mathbf{x}_j$ (only the highest weighted ones are shown here). In this example computed by our model, note how it relates the ball in the first frame to the ball in the last two frames. More examples are in Figure 3.

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks



Spacetime non-local block (self-attention layer)

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

model, R50	top-1	top-5
C2D baseline	71.8	89.7
Gaussian	72.5	90.2
Gaussian, embed	72.7	**90.5**
dot-product	**72.9**	90.3
concatenation	72.8	**90.5**

(a) **Instantiations**: 1 non-local block of different types is added into the C2D baseline. All entries are with ResNet-50.

model, R50	top-1	top-5
baseline	71.8	89.7
res_2	72.7	90.3
res_3	**72.9**	90.4
res_4	72.7	**90.5**
res_5	72.3	90.1

(b) **Stages**: 1 non-local block is added into different stages. All entries are with ResNet-50.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

model		top-1	top-5
R50	baseline	71.8	89.7
	1-block	72.7	90.5
	5-block	73.8	91.0
	10-block	**74.3**	**91.2**
R101	baseline	73.1	91.0
	1-block	74.3	91.3
	5-block	**75.1**	**91.7**
	10-block	**75.1**	91.6

(c) **Deeper non-local models**: we compare 1, 5, and 10 non-local blocks added to the C2D baseline. We show ResNet-50 (top) and ResNet-101 (bottom) results.

model		top-1	top-5
R50	baseline	71.8	89.7
	space-only	72.9	90.8
	time-only	73.1	90.5
	spacetime	**73.8**	**91.0**
R101	baseline	73.1	91.0
	space-only	74.4	91.3
	time-only	74.4	90.5
	spacetime	**75.1**	**91.7**

(d) **Space *vs*. time *vs*. spacetime**: we compare non-local operations applied along space, time, and spacetime dimensions respectively. 5 non-local blocks are used.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

model, R101	params	FLOPs	top-1	top-5
C2D baseline	1×	1×	73.1	91.0
I3D$_{3\times3\times3}$	1.5×	1.8×	74.1	91.2
I3D$_{3\times1\times1}$	**1.2**×	1.5×	74.4	91.1
NL C2D, 5-block	**1.2**×	**1.2**×	**75.1**	**91.7**

(e) **Non-local *vs*. 3D Conv**: A 5-block non-local C2D *vs*. inflated 3D ConvNet (I3D) [7]. All entries are with ResNet-101. The numbers of parameters and FLOPs are relative to the C2D baseline (43.2M and 34.2B).

	model	top-1	top-5
R50	C2D baseline	71.8	89.7
	I3D	73.3	90.7
	NL I3D	**74.9**	**91.6**
R101	C2D baseline	73.1	91.0
	I3D	74.4	91.1
	NL I3D	**76.0**	**92.1**

(f) **Non-local 3D ConvNet**: 5 non-local blocks are added on top of our best I3D models. These results show that non-local operations are complementary to 3D convolutions.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

	model	top-1	top-5
	C2D baseline	73.8	91.2
R50	I3D	74.9	91.7
	NL I3D	**76.5**	**92.6**
	C2D baseline	75.3	91.8
R101	I3D	76.4	92.7
	NL I3D	**77.7**	**93.3**

(g) **Longer clips**: we fine-tune and test the models in Table 2f on the 128-frame clips. The gains of our non-local operations are consistent.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

Nonlocal neural networks: Bring self-attention to vision tasks

model	backbone	modality	top-1 val	top-5 val	top-1 test	top-5 test	avg test†
I3D in [7]	Inception	RGB	72.1	90.3	71.1	89.3	80.2
2-Stream I3D in [7]	Inception	RGB + flow	75.7	92.0	74.2	91.3	82.8
RGB baseline in [3]	Inception-ResNet-v2	RGB	73.0	90.9	-	-	-
3-stream late fusion [3]	Inception-ResNet-v2	RGB + flow + audio	74.9	91.6	-	-	-
3-stream LSTM [3]	Inception-ResNet-v2	RGB + flow + audio	77.1	93.2	-	-	-
3-stream SATT [3]	Inception-ResNet-v2	RGB + flow + audio	77.7	93.2	-	-	-
NL I3D [ours]	ResNet-50	RGB	76.5	92.6	-	-	-
	ResNet-101	RGB	**77.7**	**93.3**	-	-	**83.8**

Kinetics dataset

Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks, CVPR 2018

3D CNN

SlowFast Networks for Video Recognition

Christoph Feichtenhofer   Haoqi Fan   Jitendra Malik   Kaiming He

Facebook AI Research (FAIR)

Abstract

We present SlowFast networks for video recognition. Our model involves (i) a Slow pathway, operating at low frame rate, to capture spatial semantics, and (ii) a Fast pathway, operating at high frame rate, to capture motion at fine temporal resolution. The Fast pathway can be made very lightweight by reducing its channel capacity, yet can learn useful temporal information for video recognition. Our models achieve strong performance for both action classification and detection in video, and large improvements are pin-pointed as contributions by our SlowFast concept. We report state-of-the-art accuracy on major video recognition benchmarks, Kinetics, Charades and AVA. Code has been made available at: `https://github.com/facebookresearch/SlowFast`.

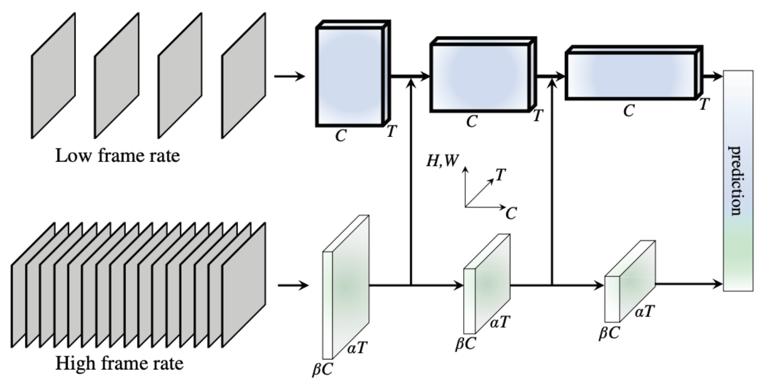


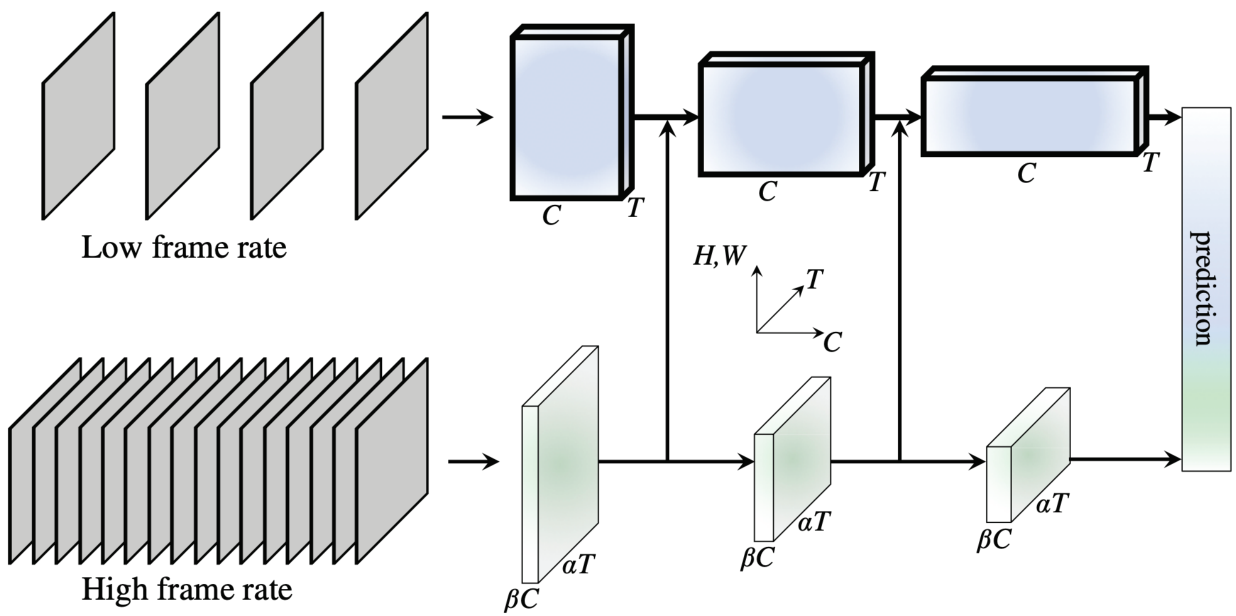
Figure 1. **A SlowFast network** has a low frame rate, low temporal resolution *Slow* pathway and a high frame rate, $\alpha\times$ higher temporal resolution *Fast* pathway. The Fast pathway is lightweight by using a fraction (β, *e.g.*, 1/8) of channels. Lateral connections fuse them.

3D CNN

SlowFast

- P cell: static
- M cells: motion fast few parameters

Slow: less input, bigger model



Fast: more input, smaller model

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

3D CNN

SlowFast

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

Kinetics dataset

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Vision transformer

TimeSFormer

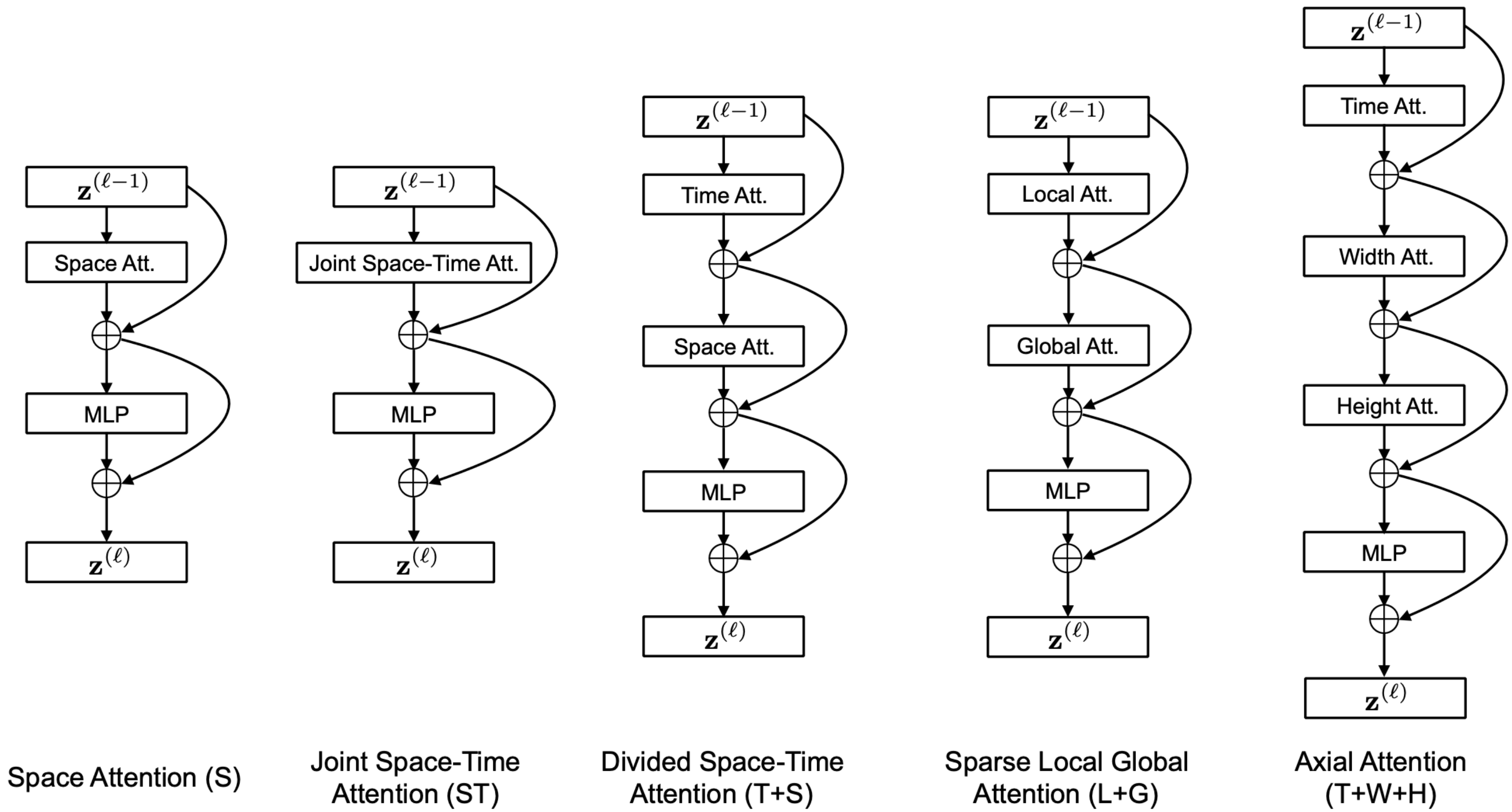
Is Space-Time Attention All You Need for Video Understanding?

Gedas Bertasius[1] **Heng Wang**[1] **Lorenzo Torresani**[1 2]

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Vision transformer

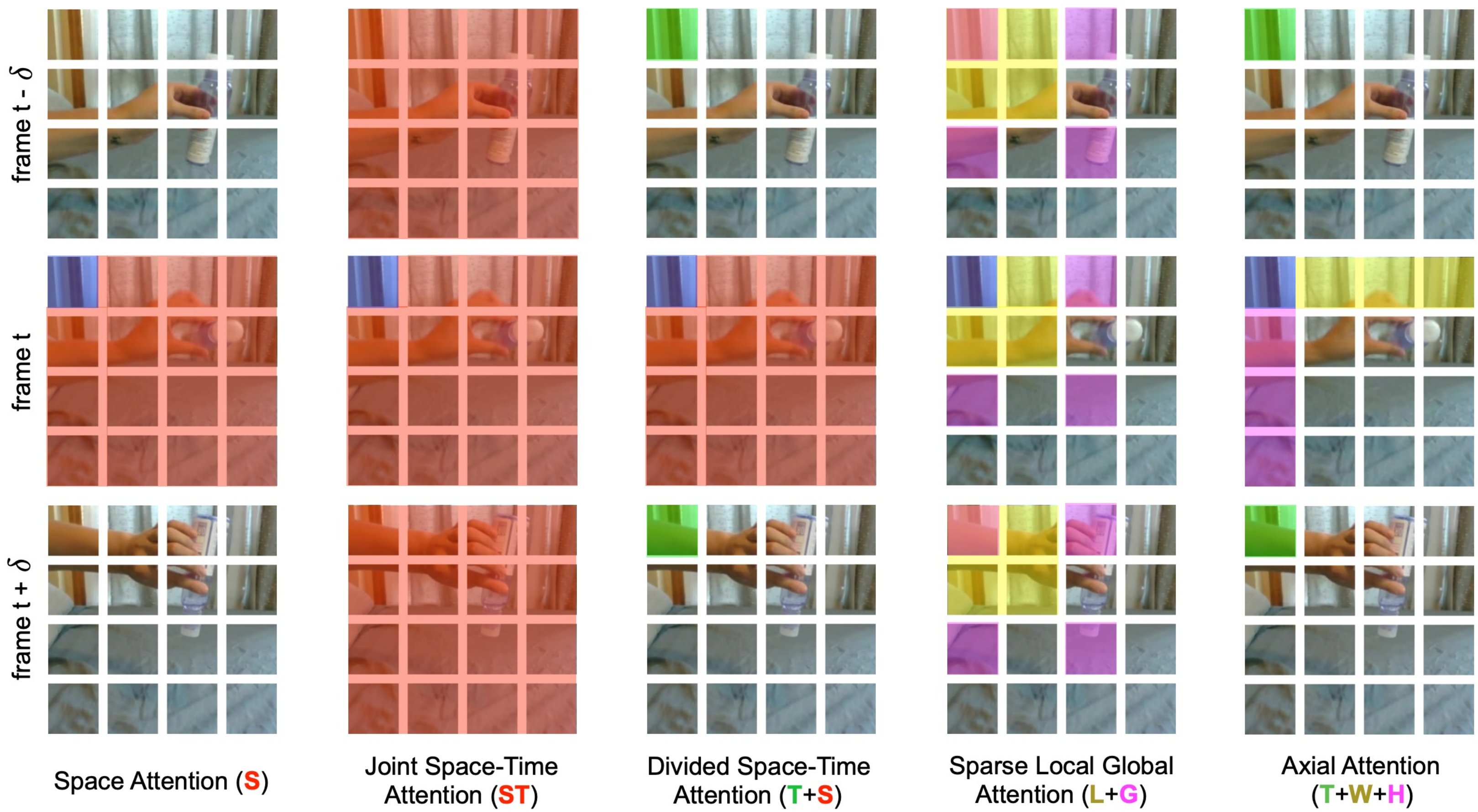
TimeSFormer [R(2+1)d]



Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Vision transformer

TimeSFormer

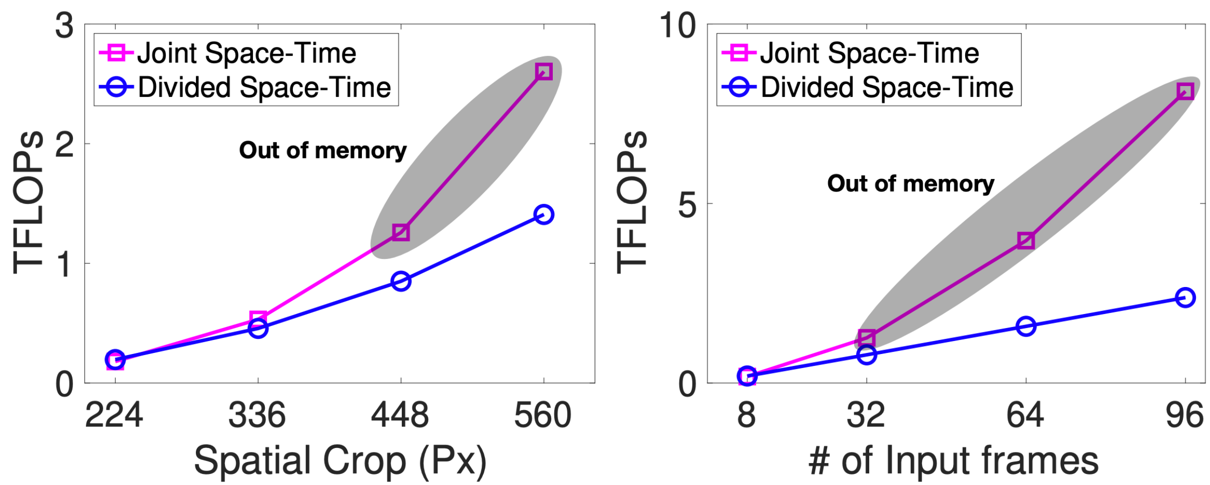


Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Vision transformer

TimeSFormer [R(2+1)d]

Attention	Params	K400	SSv2
Space	85.9M	76.9	36.6
Joint Space-Time	85.9M	77.4	58.5
Divided Space-Time	121.4M	**78.0**	**59.5**
Sparse Local Global	121.4M	75.9	56.3
Axial	156.8M	73.5	56.2



Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Vision transformer

TimeSFormer

Model	Pretrain	K400 Training Time (hours)	K400 Acc.	Inference TFLOPs	Params
I3D 8x8 R50	ImageNet-1K	444	71.0	1.11	28.0M
I3D 8x8 R50	ImageNet-1K	1440	73.4	1.11	28.0M
SlowFast R50	ImageNet-1K	448	70.0	1.97	34.6M
SlowFast R50	ImageNet-1K	3840	75.6	1.97	34.6M
SlowFast R50	N/A	6336	76.4	1.97	34.6M
TimeSformer	ImageNet-1K	**416**	75.8	**0.59**	121.4M
TimeSformer	ImageNet-21K	**416**	**78.0**	**0.59**	121.4M

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, Kaiming He, SlowFast Networks for Video Recognition, ICCV 2019

Video Understanding: Action recognition

Hand-crafted features → CNN

DeepVideo

Two-stream CNN

Two-stream → TSN

3D CNN

c3d → i3d → r(2+1)d → nonlocal neural networks

Video Transformer

TimeSFormer

Next lecture: Self-supervised learning

Y. LeCun

How Much Information is the Machine Given during Learning?

- **“Pure” Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Thank you very much!

sihengc@sjtu.edu.cn