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

CVPR 2017

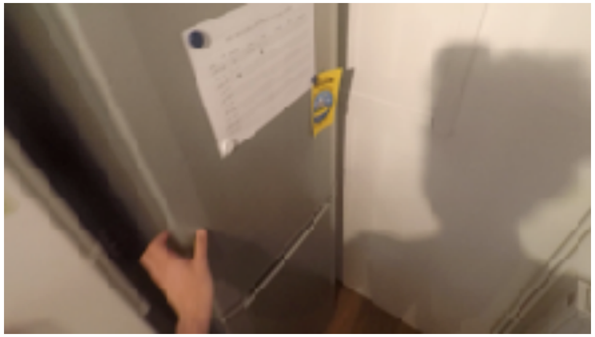
Joao Carreira, Andrew Zisserman

Problem Overview

Given a video, we want to classify it into one of the human action categories.

Cartwheeling

Braiding Hair



Opening a Fridge

Main Challenge

At this time (i.e., ~2014-2017), most action recognition models relied on Imagenet pretraining.

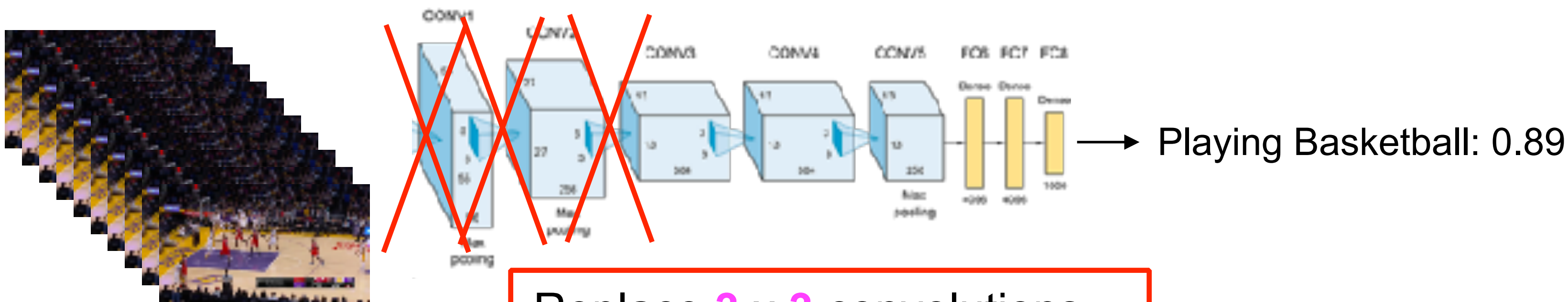
(a) **Spatial ConvNet.**

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

“Two-Stream Convolutional Networks for Action Recognition in Videos“, CVPR 2014

Main Challenges

However, adapting 2D CNNs pretrained on Imagenet to video is not trivial.



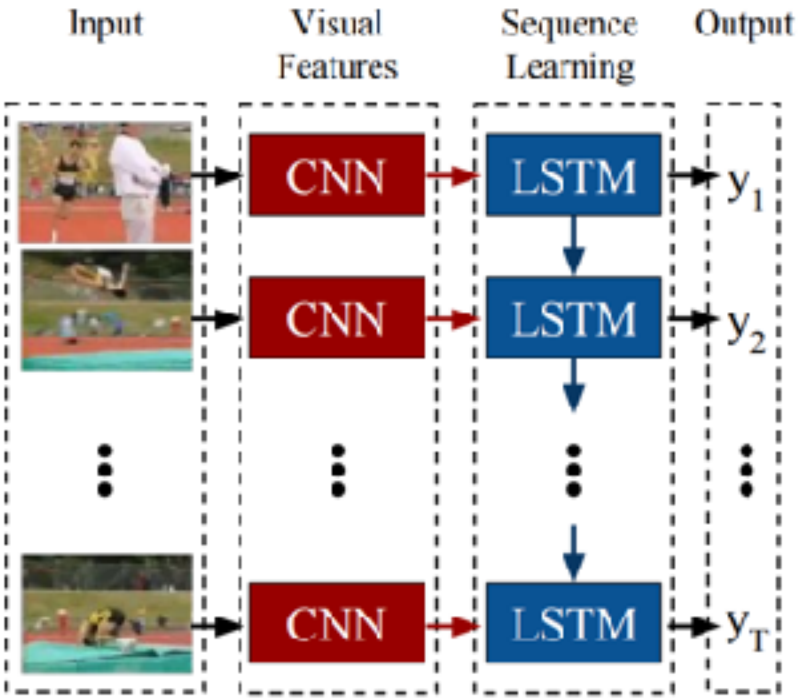
video frames 1, 2, … , 10

Replace 3 x 3 convolutions
with 2 x 3 x 3 convolutions.

“Large-scale Video Classification with Convolutional Neural Networks“, CVPR 2014

Main Challenges

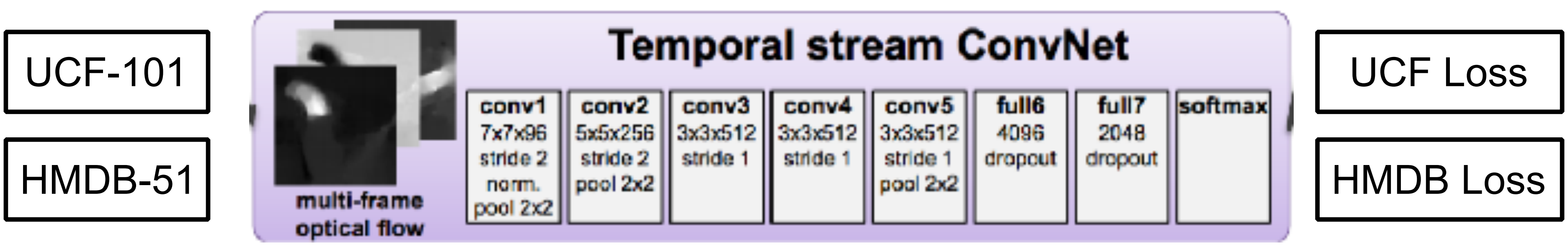
However, adapting 2D CNNs pretrained on Imagenet to video is not trivial.



“Long-term Recurrent Convolutional Networks for Visual Recognition and Description“, CVPR 2015

Main Challenges

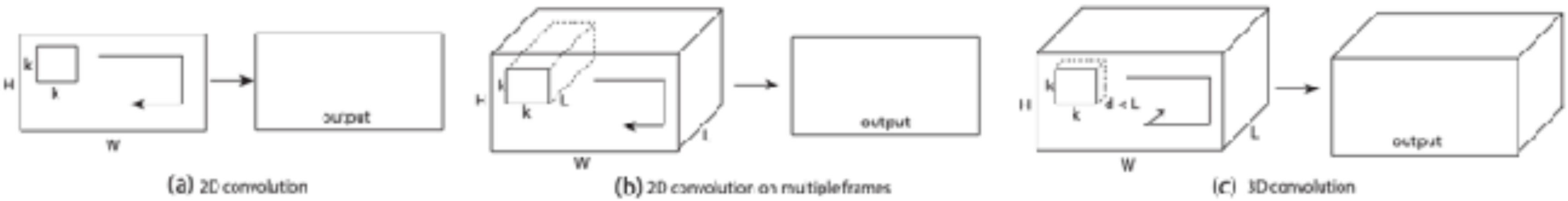
However, adapting 2D CNNs pretrained on Imagenet to video is not trivial.



“Two-Stream Convolutional Networks for Action Recognition in Videos“, CVPR 2014

Main Challenges

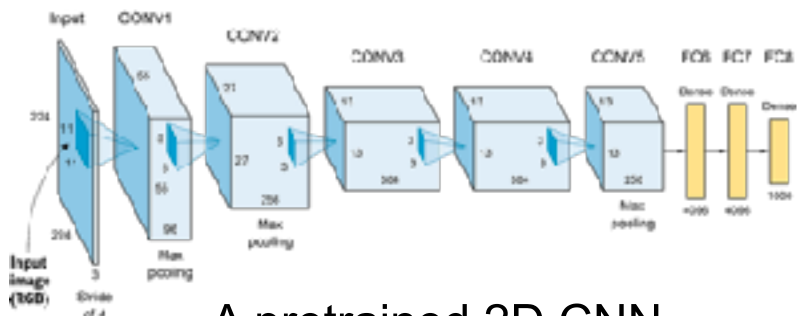
Due to a large number of parameters, it's difficult to train 3D CNNs from scratch.



"Learning Spatiotemporal Features with 3D Convolutional Networks", ICCV 2017

Inflated 3D CNNs

The goal is to transform a pretrained 2D CNN into an equivalent 3D CNN that fully re-uses the learned Imagenet features.

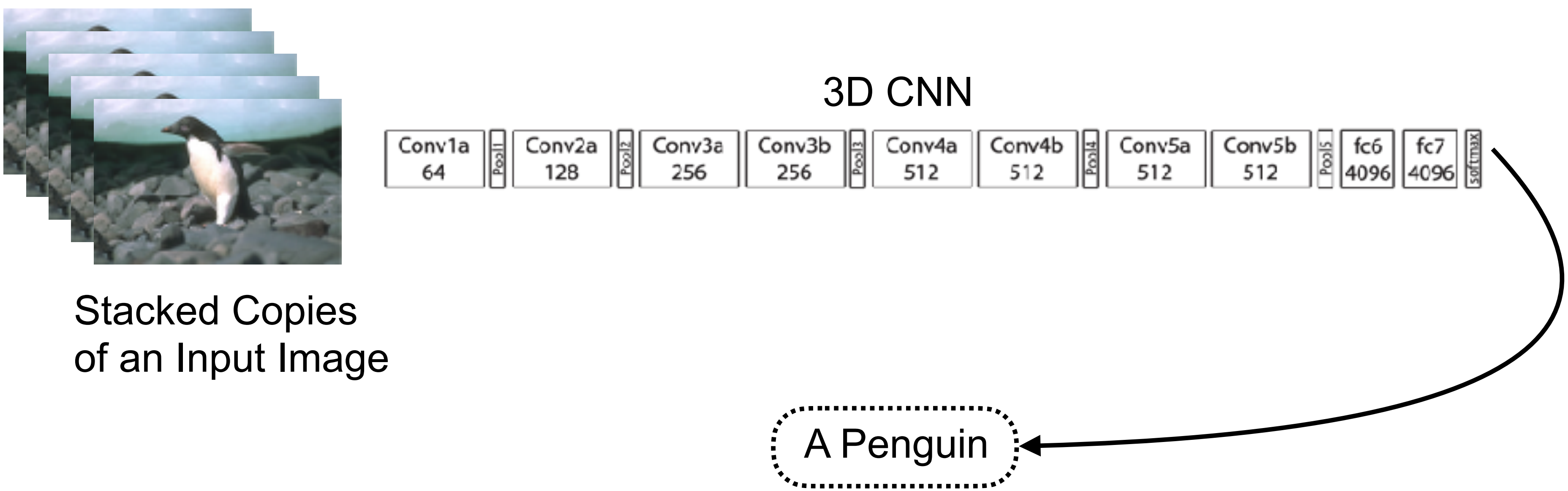


A pretrained 2D CNN

An equivalent 3D CNN

Training 3D CNNs on Imagenet

One could train a 3D CNN on Imagenet on the stacked copies of an input image.



Inflated 3D CNNs

The paper propose to inflate all pretrained 2D filters to 3D.

$$f = \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array}$$

a 2D grid (e.g., an image)

$$g = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & -1 & -2 \\ \hline 1 & 2 & -1 \\ \hline \end{array}$$

2D convolutional filter

$$h = g * f = \boxed{-8}$$

Inflated 3D CNNs

The paper propose to inflate all pretrained 2D filters to 3D.

$$f = \begin{array}{cl} \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t-1} \\ \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t} \\ \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t+1} \end{array}$$

a 3D grid (e.g., a video clip)

$$g = \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & 2 & -1 \end{bmatrix}$$

2D convolutional filter

$$h = g * f = \begin{array}{cl} \boxed{-8} & \text{time t-1} \\ \boxed{-8} & \text{time t} \\ \boxed{-8} & \text{time t+1} \end{array}$$

Inflated 3D CNNs

The paper propose to inflate all pretrained 2D filters to 3D.

$$f = \begin{array}{ll} \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t-1} \\ \\ \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t} \\ \\ \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t+1} \end{array}$$

a 3D grid (e.g., a video clip)

$$g = \begin{array}{ll} \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & -1 & -2 \\ \hline 1 & 2 & -1 \\ \hline \end{array} & \text{time t-1} \\ \\ \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & -1 & -2 \\ \hline 1 & 2 & -1 \\ \hline \end{array} & \text{time t} \\ \\ \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & -1 & -2 \\ \hline 1 & 2 & -1 \\ \hline \end{array} & \text{time t+1} \end{array}$$

3D convolutional filter

$$h = g * f = \boxed{-24}$$

Inflated 3D CNNs

The paper propose to inflate all pretrained 2D filters to 3D.

$$f = \begin{array}{ll} \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t-1} \\ \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t} \\ \begin{bmatrix} 1 & 2 & 3 \\ -5 & 6 & 1 \\ 2 & -2 & -4 \end{bmatrix} & \text{time t+1} \end{array} \qquad g = \begin{array}{ll} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & 2 & -1 \end{bmatrix} / 3 & \text{time t-1} \\ \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & 2 & -1 \end{bmatrix} / 3 & \text{time t} \\ \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & 2 & -1 \end{bmatrix} / 3 & \text{time t+1} \end{array} \qquad h = g * f = \boxed{-8}$$

a 3D grid (e.g., a video clip)

3D convolutional filter

Inflated 3D CNNs

The paper propose to inflate all pretrained 2D filters to 3D.

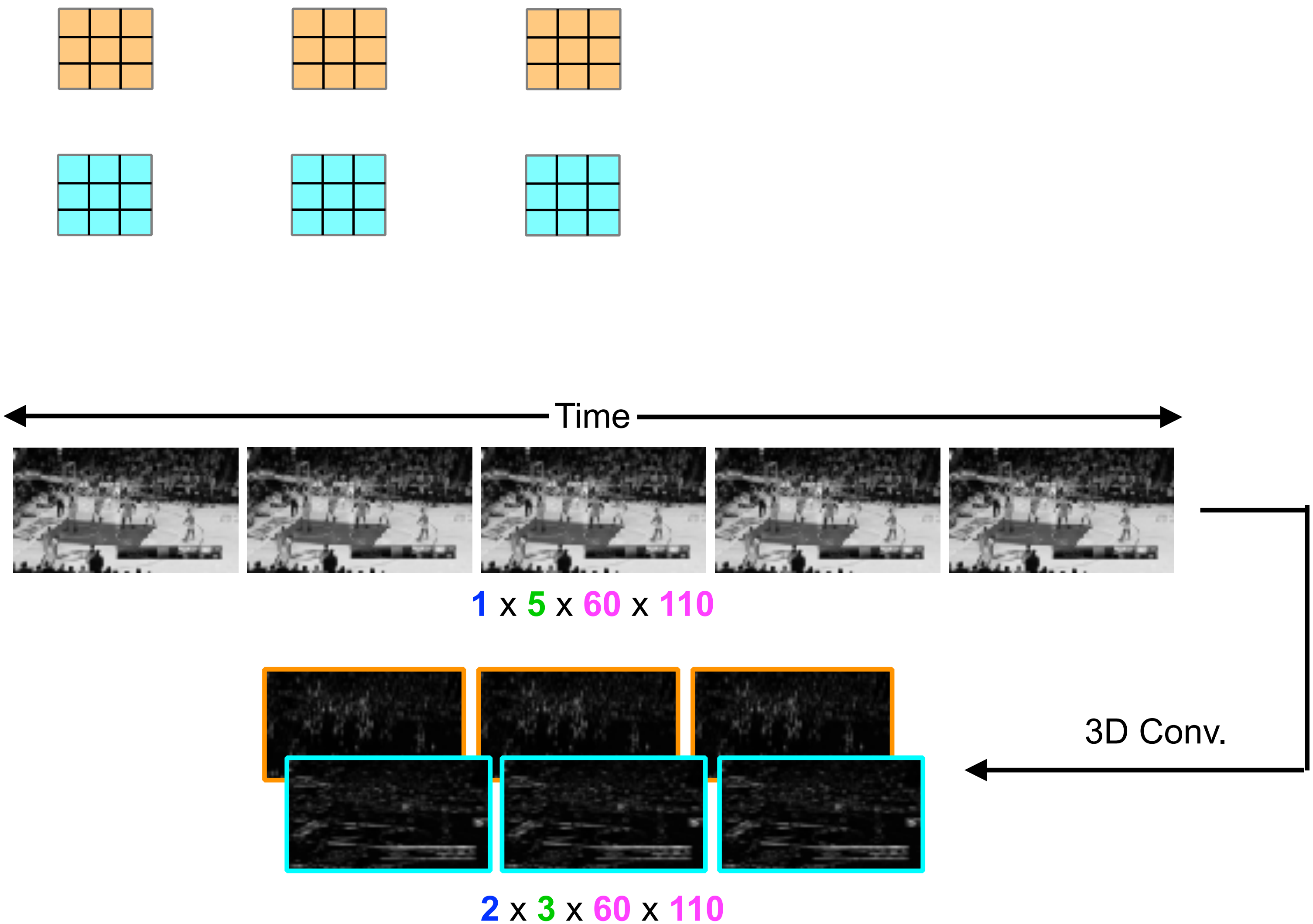
$$f = \begin{array}{cl} \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t-1} \\ \\ \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t} \\ \\ \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline -5 & 6 & 1 \\ \hline 2 & -2 & -4 \\ \hline \end{array} & \text{time t+1} \end{array} \qquad g = \begin{array}{cl} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} & \text{time t-1} \\ \\ \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & -1 & -2 \\ \hline 1 & 2 & -1 \\ \hline \end{array} & \text{time t} \\ \\ \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} & \text{time t+1} \end{array} \qquad h = g * f = \boxed{-8}$$

a 3D grid (e.g., a video clip)

3D convolutional filter

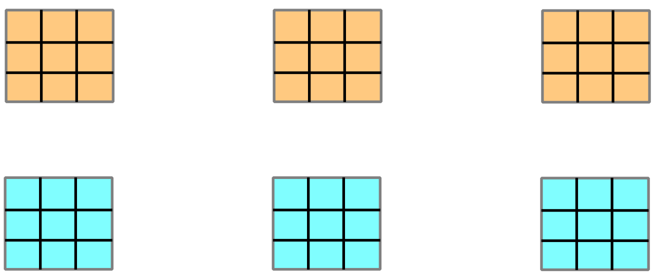
3D Convolution

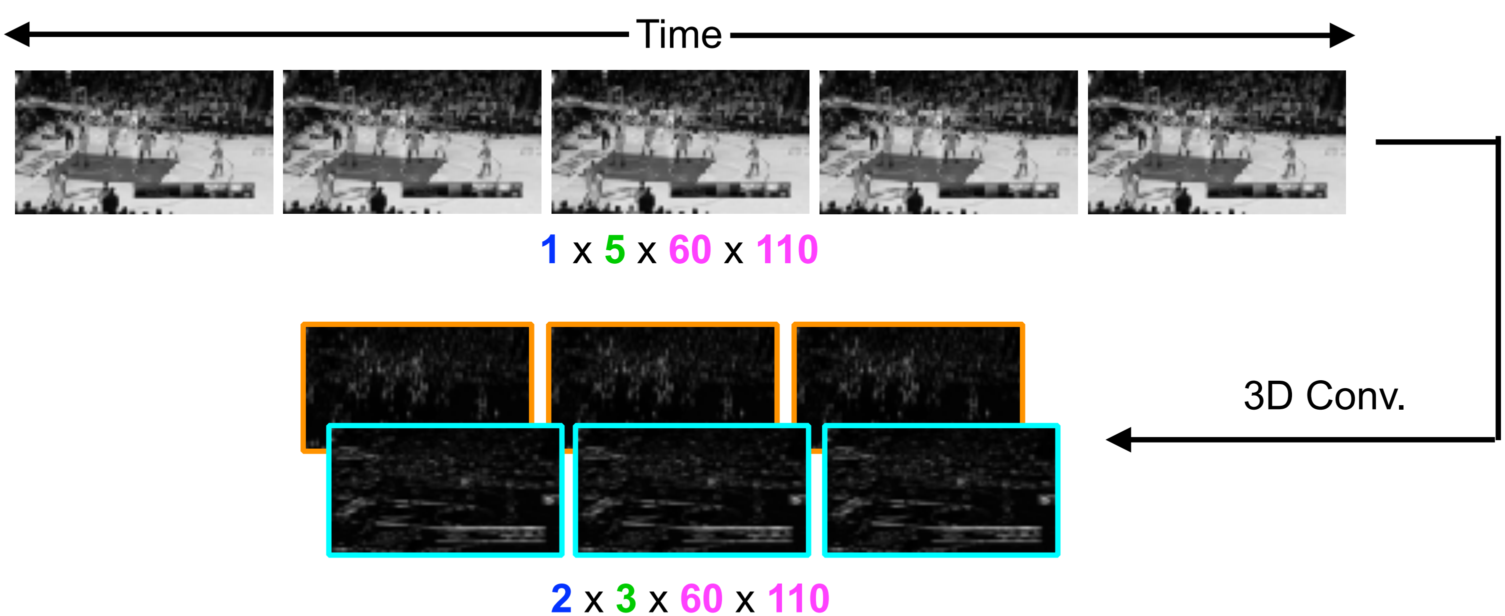
Learnable 3 x 3 x 3 Convolutional Kernels (Temporal, Spatial)



3D Convolution

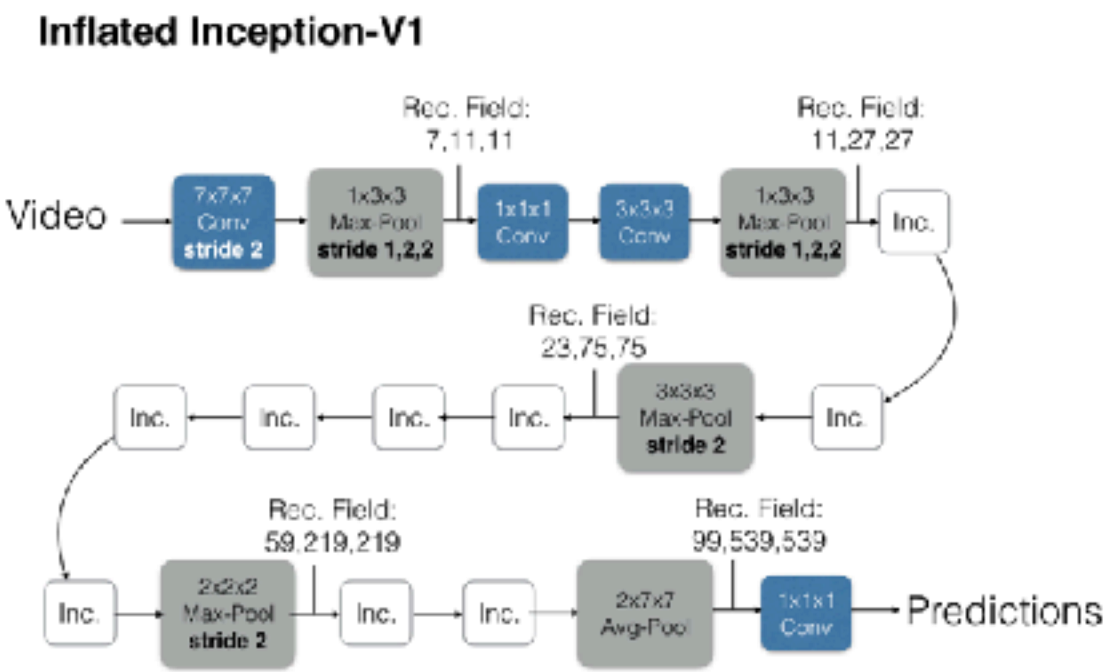
Learnable 3 x 3 x 3 Convolutional Kernels (Temporal, Spatial)

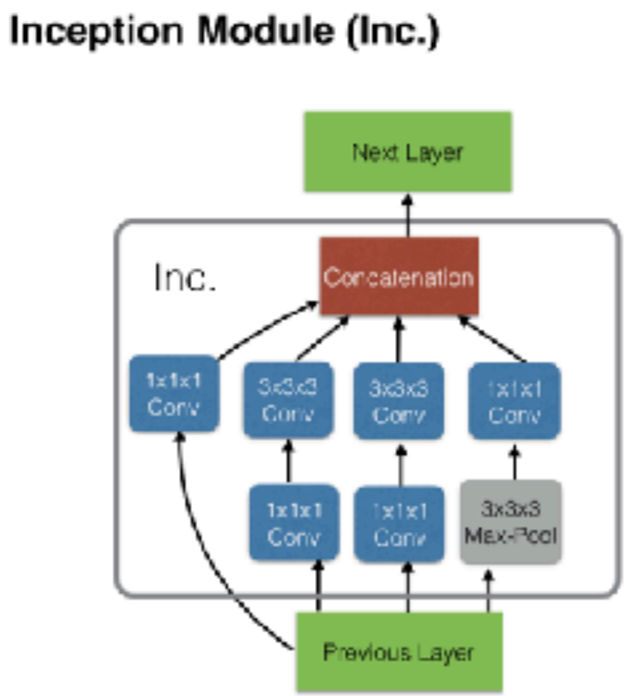




Inflated 3D CNNs

The Inflated Inception-V1 architecture (left) and its detailed inception submodule (right).

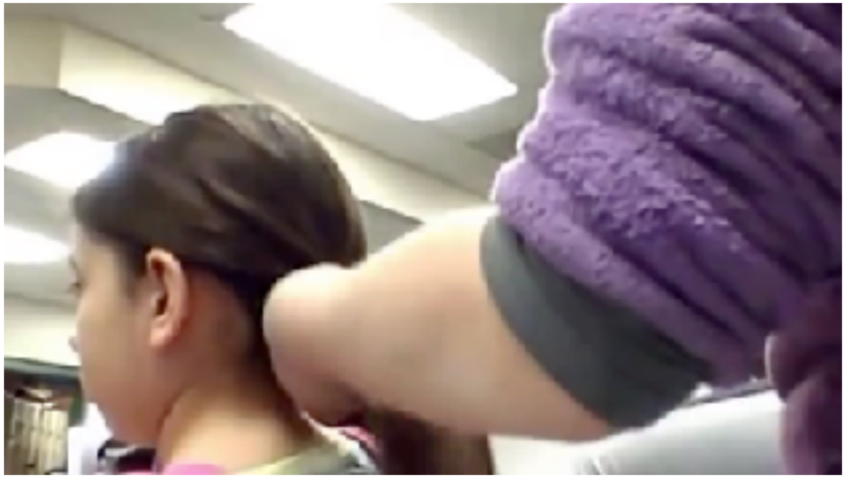




Kinetics Dataset

- ~240K YouTube videos manually annotated with 400 human action classes.
- The clips last around 10s.

Cartwheeling



Braiding Hair

Comparison with Prior Architectures

- I3D can be used to model longer temporal extents with fewer parameters than prior approaches.
- All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet.

Method	#Params	Training		Testing	
		# Input Frames	Temporal Footprint	# Input Frames	Temporal Footprint
ConvNet+LSTM	9M	25 rgb	5s	50 rgb	10s
3D-ConvNet	79M	16 rgb	0.64s	240 rgb	9.6s
Two-Stream	12M	1 rgb, 10 flow	0.4s	25 rgb, 250 flow	10s
3D-Fused	39M	5 rgb, 50 flow	2s	25 rgb, 250 flow	10s
Two-Stream I3D	25M	64 rgb, 64 flow	2.56s	250 rgb, 250 flow	10s

Comparison with Prior Architectures

- I3D can be used to model longer temporal extents with fewer parameters than prior approaches.
- All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet.

Method	#Params	Training		Testing	
		# Input Frames	Temporal Footprint	# Input Frames	Temporal Footprint
ConvNet+LSTM	9M	25 rgb	5s	50 rgb	10s
3D-ConvNet	79M	16 rgb	0.64s	240 rgb	9.6s
Two-Stream	12M	1 rgb, 10 flow	0.4s	25 rgb, 250 flow	10s
3D-Fused	39M	5 rgb, 50 flow	2s	25 rgb, 250 flow	10s
Two-Stream I3D	25M	64 rgb, 64 flow	2.56s	250 rgb, 250 flow	10s

Due to fewer parameters, it's easier to train I3D than C3D.

Comparison with Prior Architectures

- I3D can be used to model longer temporal extents with fewer parameters than prior approaches.
- All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet.

Method	#Params	Training		Testing	
		# Input Frames	Temporal Footprint	# Input Frames	Temporal Footprint
ConvNet+LSTM	9M	25 rgb	5s	50 rgb	10s
3D-ConvNet	79M	16 rgb	0.64s	240 rgb	9.6s
Two-Stream	12M	1 rgb, 10 flow	0.4s	25 rgb, 250 flow	10s
3D-Fused	39M	5 rgb, 50 flow	2s	25 rgb, 250 flow	10s
Two-Stream I3D	25M	64 rgb, 64 flow	2.56s	250 rgb, 250 flow	10s

I3D processes many more video frames than prior state-of-the-art two-stream methods.

Comparison with Prior Architectures

- Evaluation is done on UCF-101, HMDB-51, and Kinetics datasets.
- All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet.

Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

Comparison with Prior Architectures

- Evaluation is done on UCF-101, HMDB-51, and Kinetics datasets.
- All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet.

Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

I3D performs best suggesting that the benefits of ImageNet pre-training extend to 3D CNNs.

Importance of Imagenet Pretraining

Performance training and testing on Kinetics with and without ImageNet pretraining.

	Kinetics			ImageNet then Kinetics		
Architecture	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

Importance of Imagenet Pretraining

Performance training and testing on Kinetics with and without ImageNet pretraining.

Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

Imagenet pretraining is beneficial even when training on large-scale datasets such as Kinetics

Generalizability of Kinetics Features

- Evaluating how well Kinetics features transfer to smaller UCF-101 and HMDB-51 datasets.
- The results are evaluated with / without ImageNet pretrained weights

Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

Generalizability of Kinetics Features

- Evaluating how well Kinetics features transfer to smaller UCF-101 and HMDB-51 datasets.
- The results are evaluated with / without ImageNet pretrained weights

Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

Kinetics pretraining leads to substantial gains on both datasets for all models.

Generalizability of Kinetics Features

- Evaluating how well Kinetics features transfer to smaller UCF-101 and HMDB-51 datasets.
- The results are evaluated with / without ImageNet pretrained weights

Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

Training on fixed Kinetics features leads to much better performance than full training on UCF / HMDB alone.

Generalizability of Kinetics Features

- Evaluating how well Kinetics features transfer to smaller UCF-101 and HMDB-51 datasets.
- The results are evaluated with / without ImageNet pretrained weights

Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

Imagenet pretraining is still beneficial in most cases.

Comparison to the State-of-the-Art

Comparison to all prior action recognition methods on UCF-101 and HMDB-51.

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

Two-stream I3D achieves best performance on both datasets.

Comparison to the State-of-the-Art

Comparison to all prior action recognition methods on UCF-101 and HMDB-51.

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

Kinetics pretraining brings ~5% and ~14% improvement on UCF and HMDB respectively.

Contributions

- Simple yet effective way to adapt pretrained image models to video.
- Very important dataset contribution.
- Great transfer learning performance.
- State-of-the-art action recognition results.
- Good ablation experiments.

Weaknesses

- The proposed model relies heavily on pretrained image-level models.
- The necessity for a two-stream architecture.
- Unclear what the inflated 3D filters actually learn when trained on the video data.
- Kinetics is a spatially biased dataset.

Discussion Questions

- What are some of the constraints imposed by using a pretrained Imagenet model?

Discussion Questions

- What are some of the constraints imposed by using a pretrained Imagenet model?
- In general, does it make sense to start with image-level representation and fine-tune it to video?

Discussion Questions

- What are some of the constraints imposed by using a pretrained Imagenet model?
- In general, does it make sense to start with image-level representation and fine-tune it to video?
- Should you put more effort into data collection or model development ?