Lecture 10: Videos

June 6st, 2021

Tali Dekel

Videos

Videos are all around us

Span an enormous space of spatial and temporal signals

LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA

[redacted]

Challenges in Videos: size of video

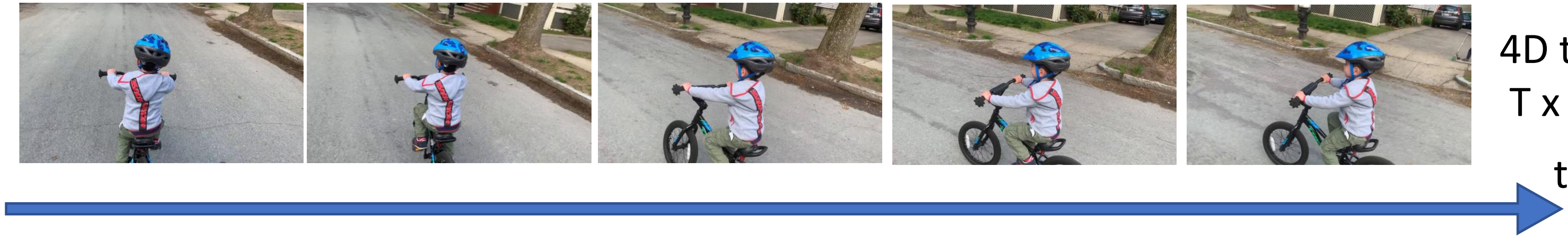
Image

video

Size of video >> size of image

Computational constrains → short, low-res clips

3 x H x W



4D tensor:
T x 3 x H x W

~30 frames per second (fps)

Uncompressed size (3 bytes per pixel):
SD (640 x 480): **~1.5 GB per minute**
HD (1920 x 1080): **~10 GB per minute**

Reduce spatial and temporal resolution

5fps, half the spatial resolution

Slide inspiration: Justin Johnson, EECS 498-007

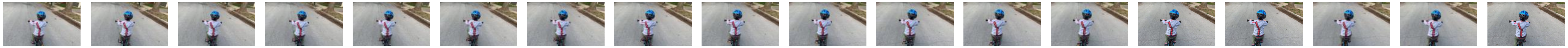
Challenges in Videos: size of video

Computational constrains → short, low-res clips

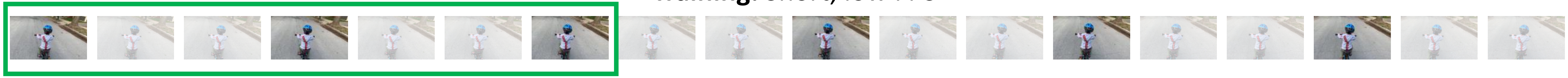
Input video

Walking
Running
Cycling
Jumping
.
.

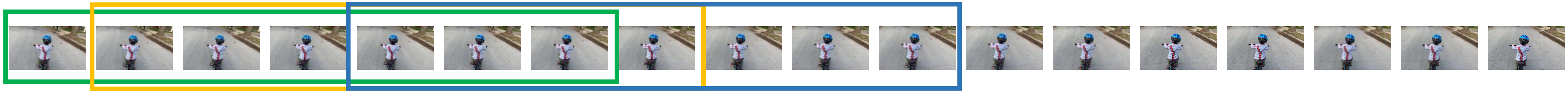
Original video(long, high FPS)



Training: Short, low FPS



Test time: inference on different short clips, average predictions



Slide inspiration: Justin Johnson, EECS 498-007

[redacted]

Challenges in Videos: Videos Datasets

space of video >> space of image → lots of training data

"ImageNet"-equivalent dataset for videos?

Massive human labelling efforts

UCF101

YouTube videos

13320 videos, 101 action categories

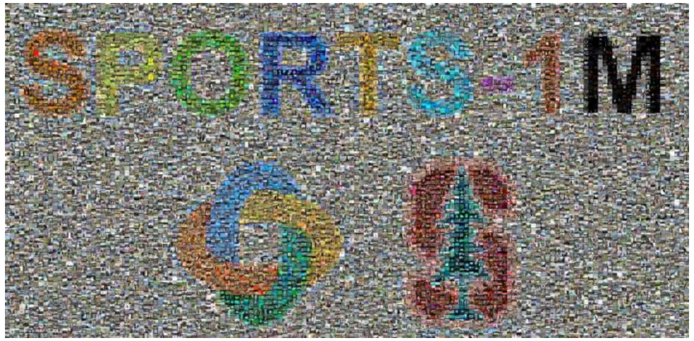
Kinetics

YouTube videos

650,000 video clips, 600 human action classes

YouTube-8M

8M video clips, Machine-generated annotations from 3,862 classes



Sports-1M

YouTube videos

1,133,157 videos, 487 sports labels

[redacted]

Today

Deep Learning-based Models for Videos

- How to reduce computation cost without sacrificing accuracy?
- What architecture to best capture temporal patterns?

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

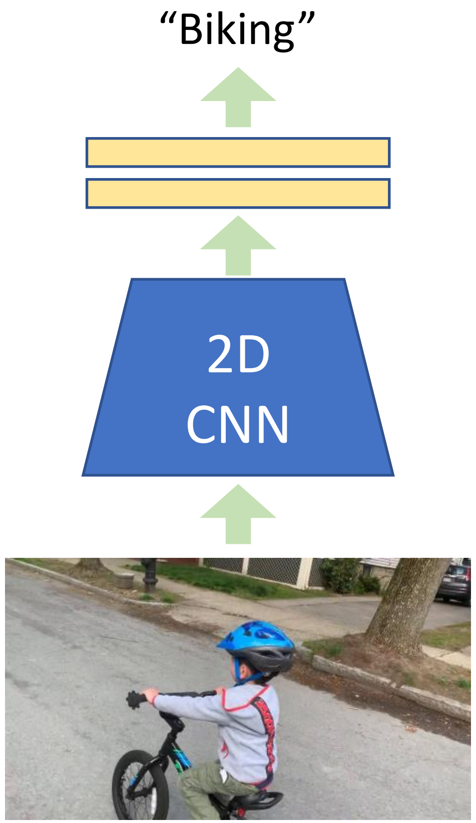
Self-Supervision in Videos

- Which types of pretext tasks can we define to capture temporal information?
- Applications

[redacted]

Models for Videos: Single-Frame Baseline

- Train 2D CNN to classify video frames independently



Input video frame

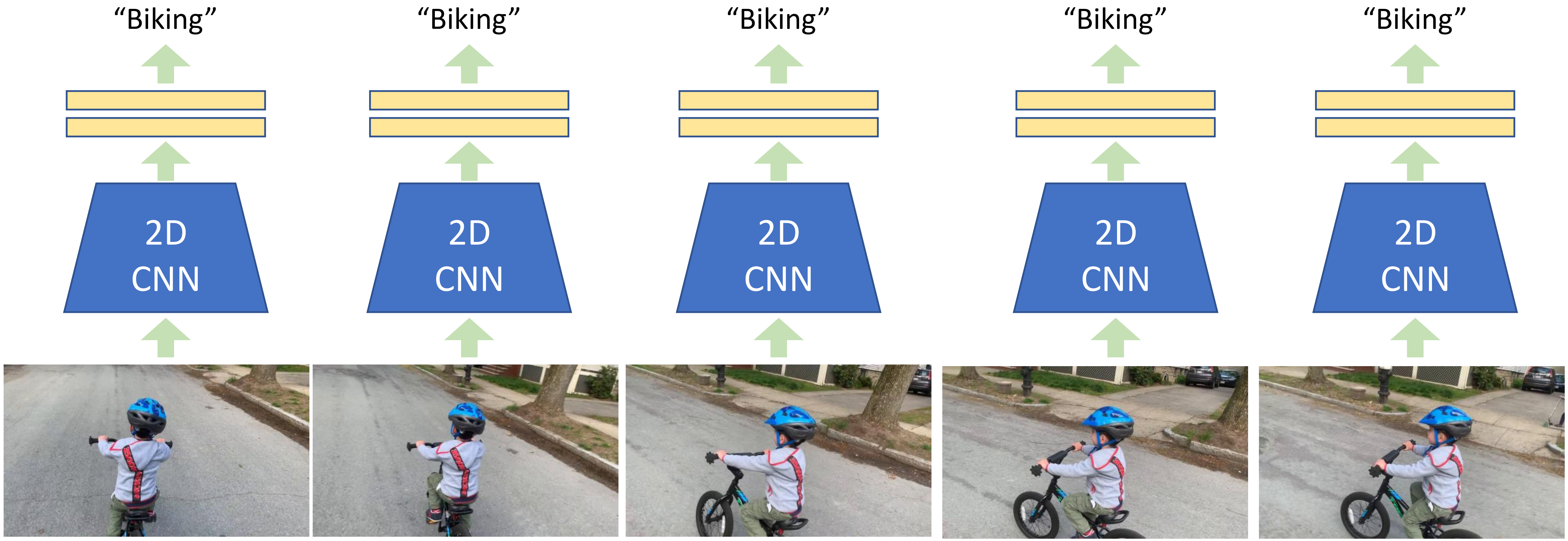
Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014

[redacted]

Models for Videos: Single-Frame Baseline

- Train 2D CNN to classify video frames independently
- Average predicted probs at test-time

Often a surprisingly strong baseline!



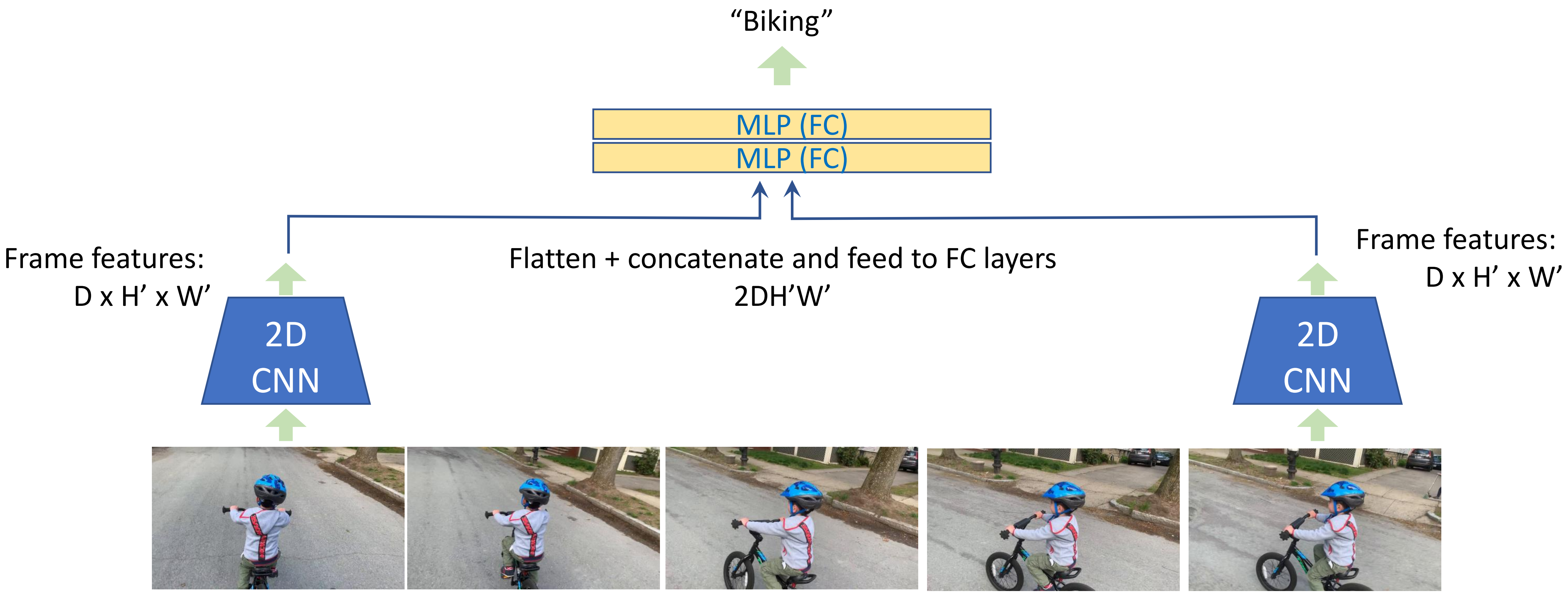
Input video frames

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

[redacted]

Models for Videos: Late Fusion

- Learn features for each frame using a 2D CNN, concatenate feature, and fuse

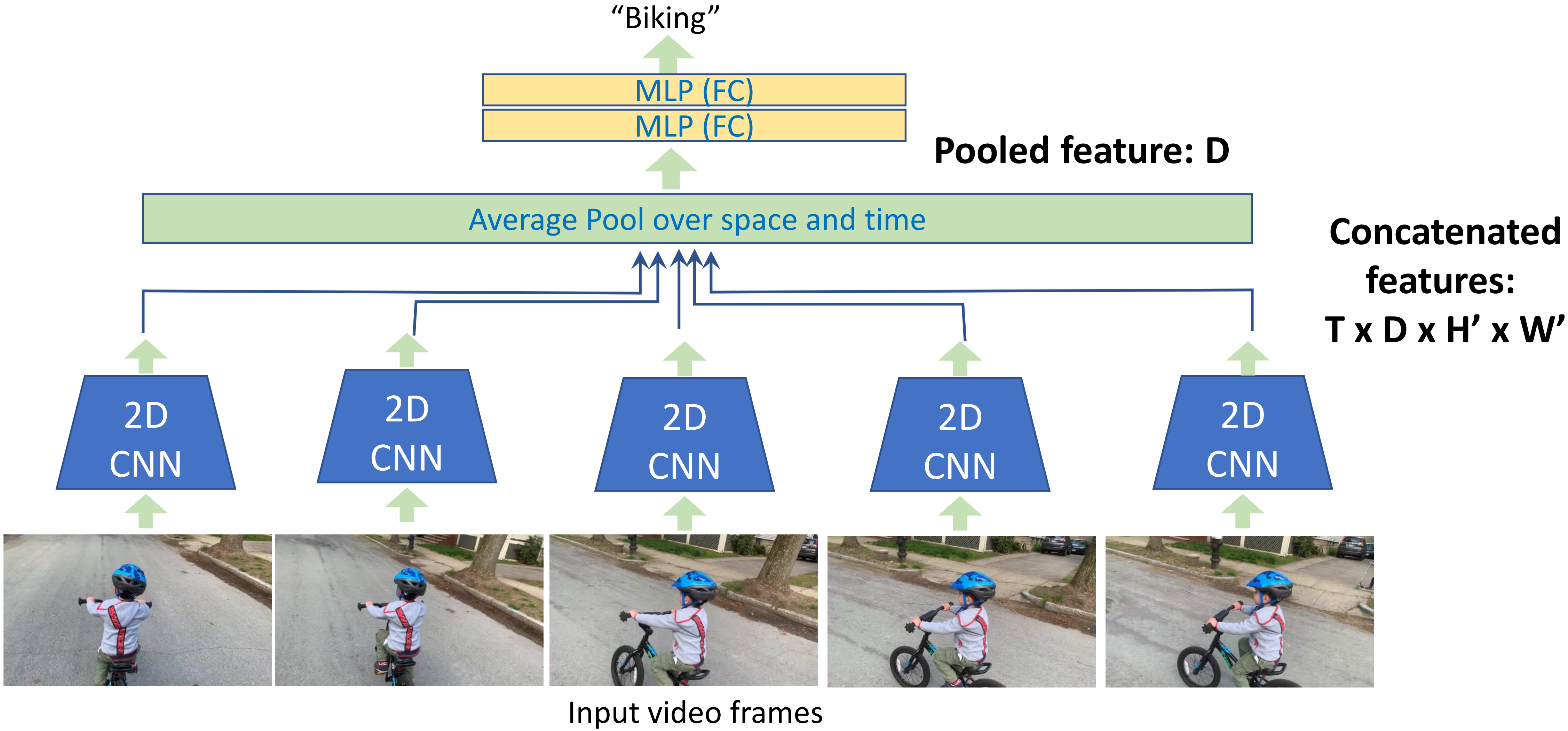


Input video frames

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

Models for Videos: Late Fusion w/ pooling

Learn features for each frame, apply spatial-temporal average pool, and then fuse



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

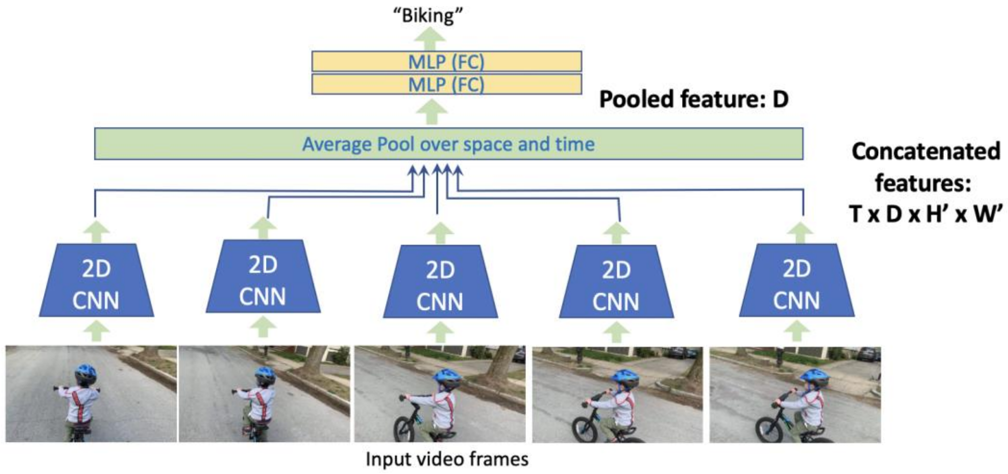
[redacted]

Models for Videos: Late Fusion w/ pooling

Learn features for each frame, apply spatial-temporal average pool, and then fuse

Pros: allow the network to learn global motion characteristics by comparing outputs of both towers

Cons: late fusion is late…
hard to represent low level motion between frames

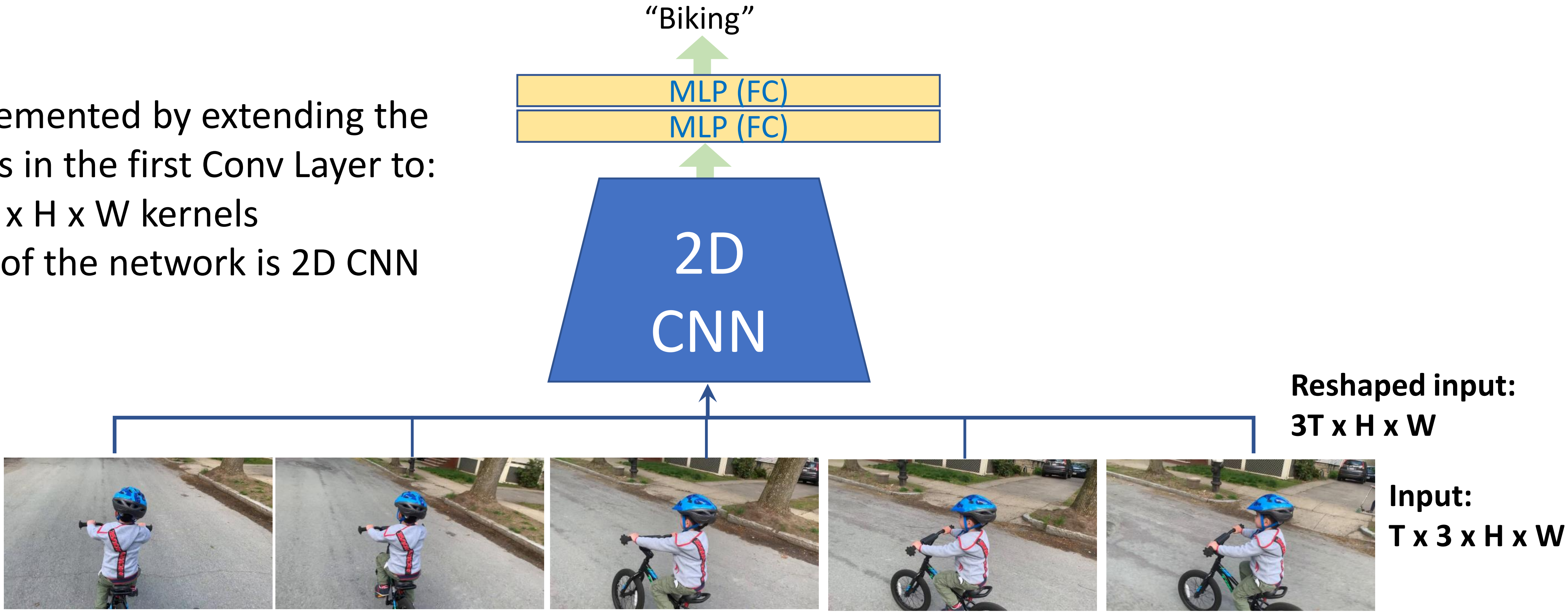


Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014

Models for Videos: Early Fusion

- Combines temporal information immediately on the pixel level
- Treat time as another “channel” dimension

Implemented by extending the filters in the first Conv Layer to:
T x 3 x H x W kernels
Rest of the network is 2D CNN



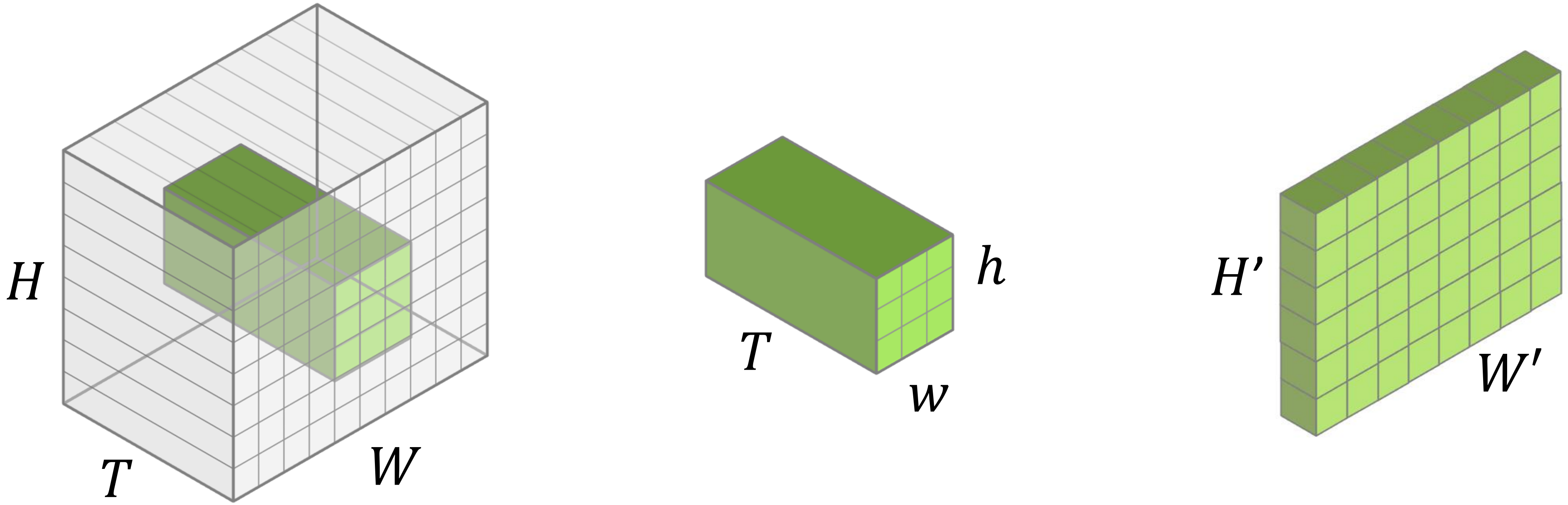
Input video frames

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

[redacted]

Models for Videos: Early Fusion

Extending the filters in the first Conv Layer to: T x 3 x H x W kernel



Input: T x 3 x H x W

Weights: C x T x 3 x h x w

Output: C x H’ x W’

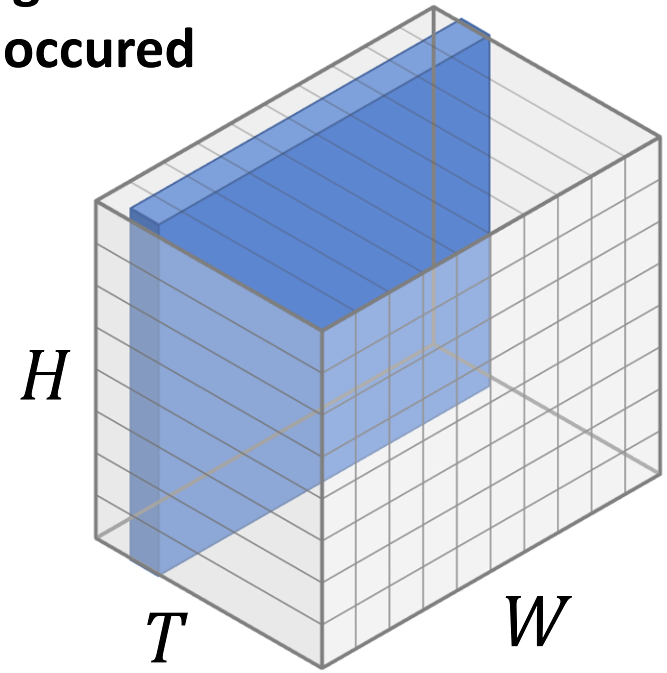
[redacted]

Models for Videos: Early Fusion

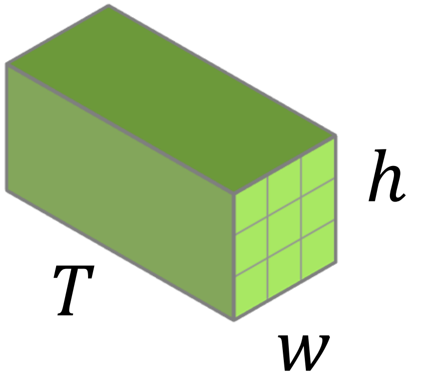
Extending the filters in the first Conv Layer to: T x 3 x H x W kernel

- Not temporal shift invariance; specific filter is learned to each time step

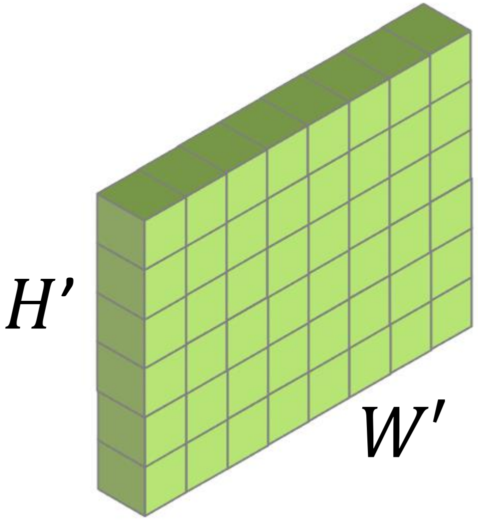
Large motion occured



Input: T x 3 x H x W



Weights: C x T x 3 x h x w



Output: C x H’ x W’

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

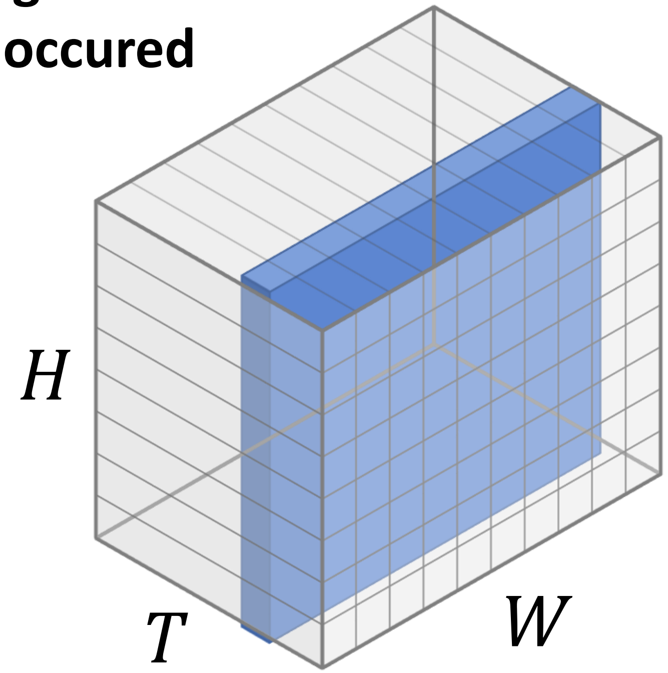
[redacted]

Models for Videos: Early Fusion

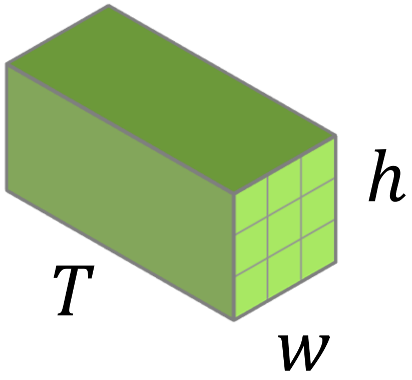
Extending the filters in the first Conv Layer to: T x 3 x H x W kernel

- Not temporal shift invariance; specific filter is learned to each time step

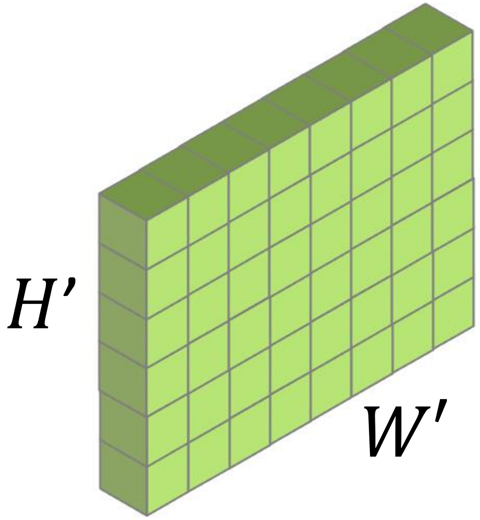
Large motion occured



Input: T x 3 x H x W



Weights: C x T x 3 x h x w



Output: C x H’ x W’

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

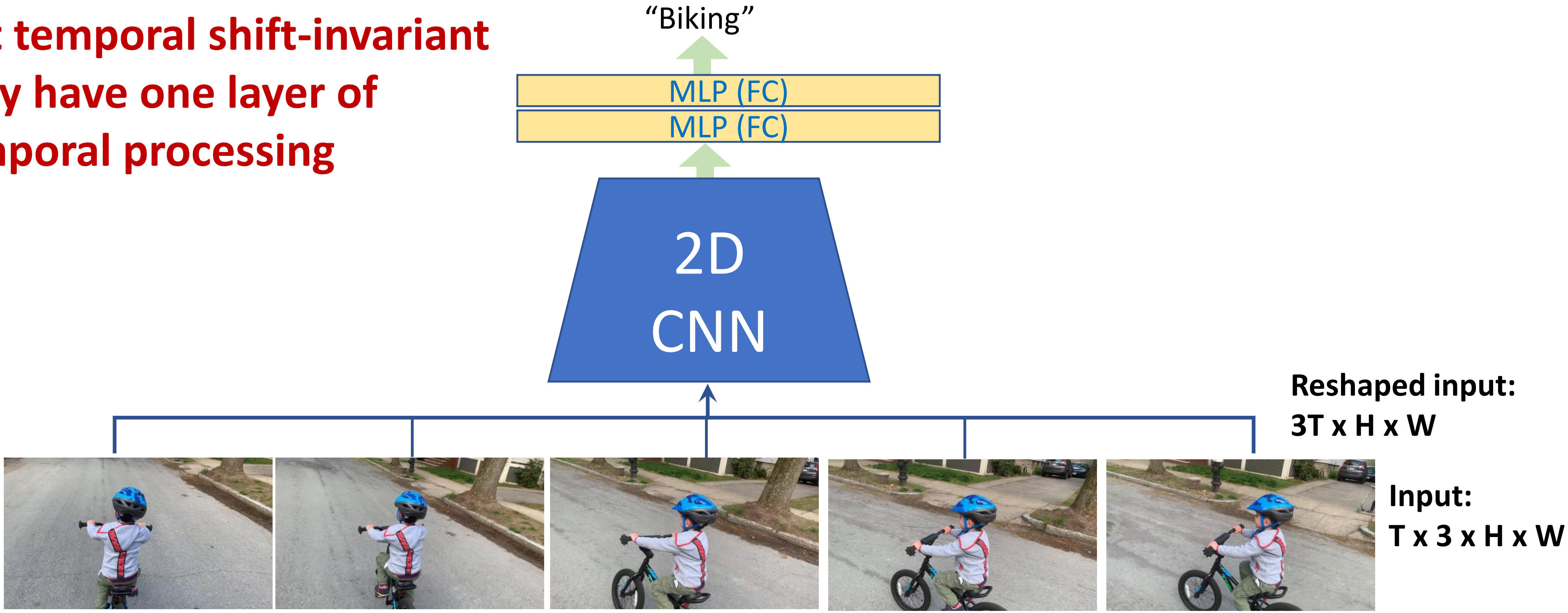
[redacted]

Models for Videos: Early Fusion

Pros: Allow the network to learn local motion characteristics

Cons:

- **Not temporal shift-invariant**
- **Only have one layer of temporal processing**



Input video frames

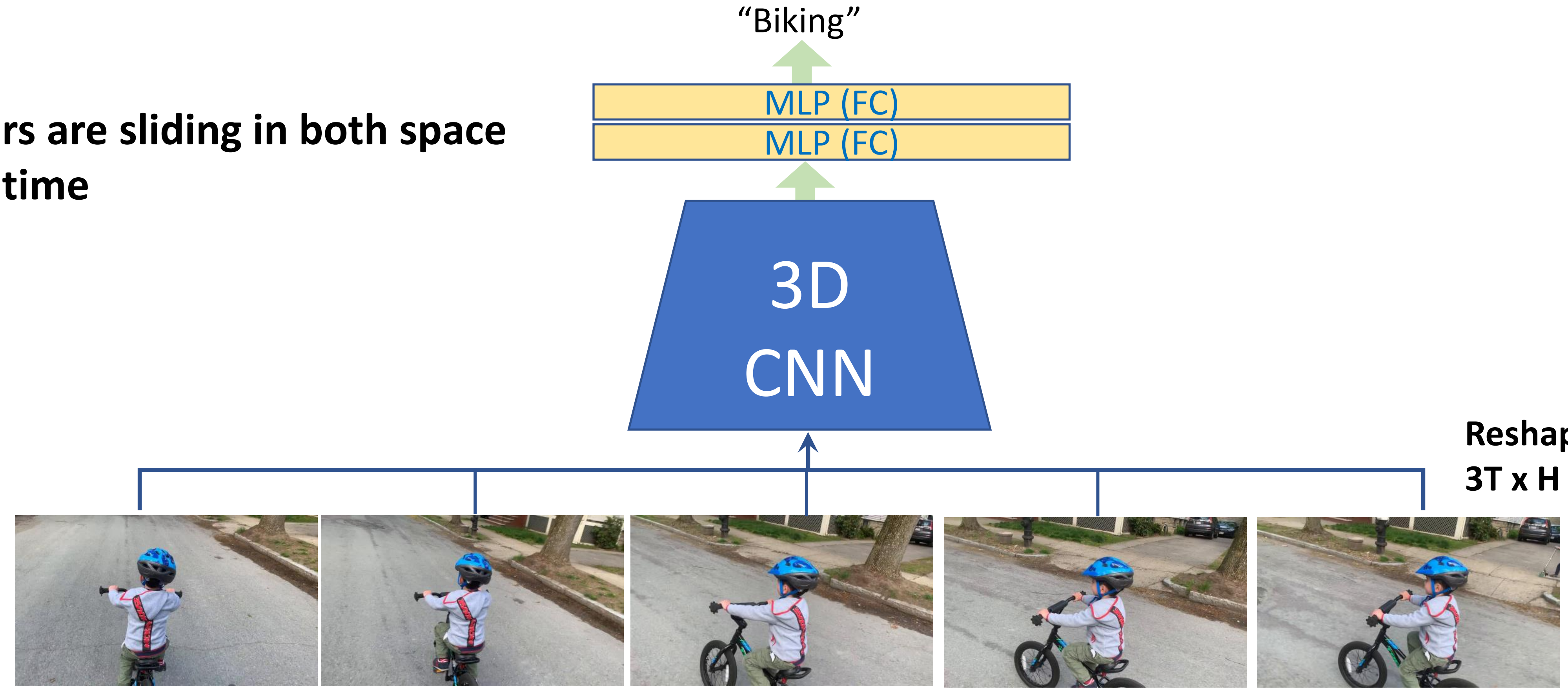
Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014

[redacted]

Models for Videos: Slow Fusion a.k.a 3D Convs

- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model

Filters are sliding in both space and time



Reshaped input:
3T x H x W

Input:
T x 3 x H x W

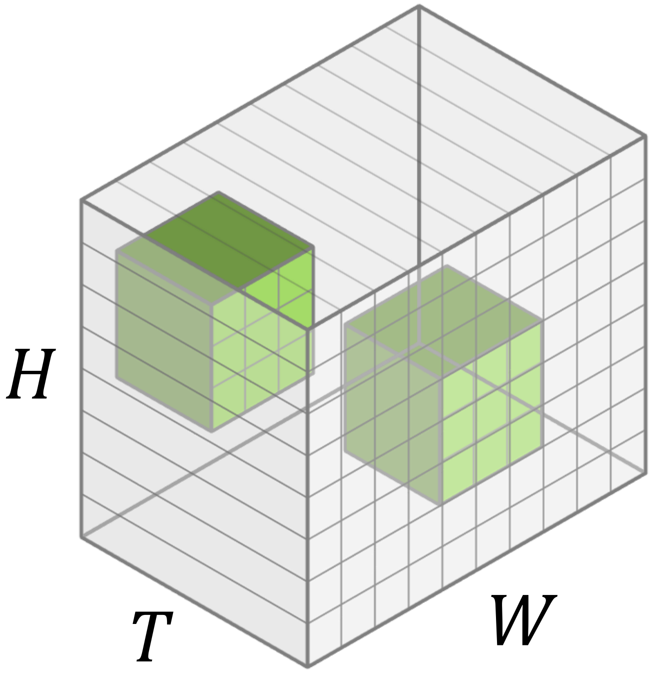
Input video frames

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

[redacted]

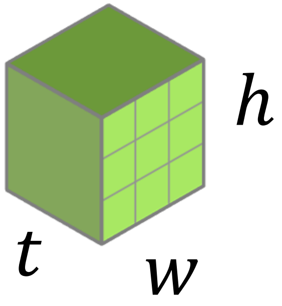
Models for Videos: Slow Fusion a.k.a 3D Convs

- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model
- **Slide the kernels in both space and time**

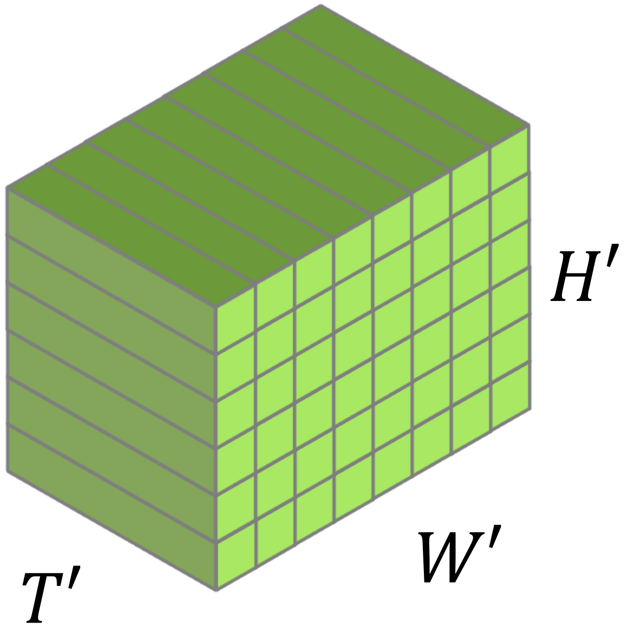


Input: T x 3 x H x W

- Temporal shift-invariant!



Weights: C x t x 3 x h x w

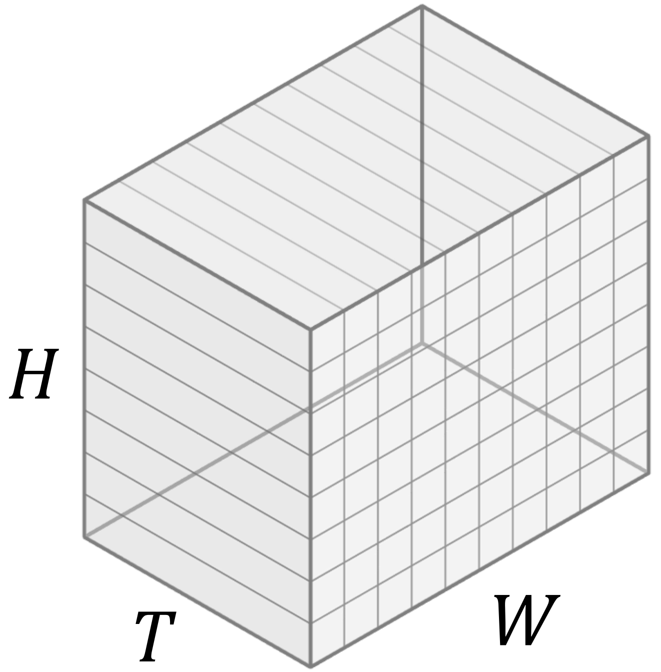


Output: C x T’ x H’ x W’

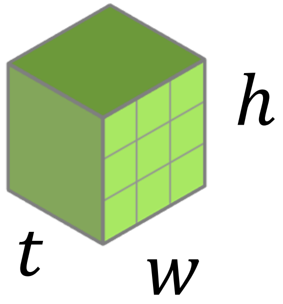
[redacted]

Models for Videos: Slow Fusion a.k.a 3D Convs

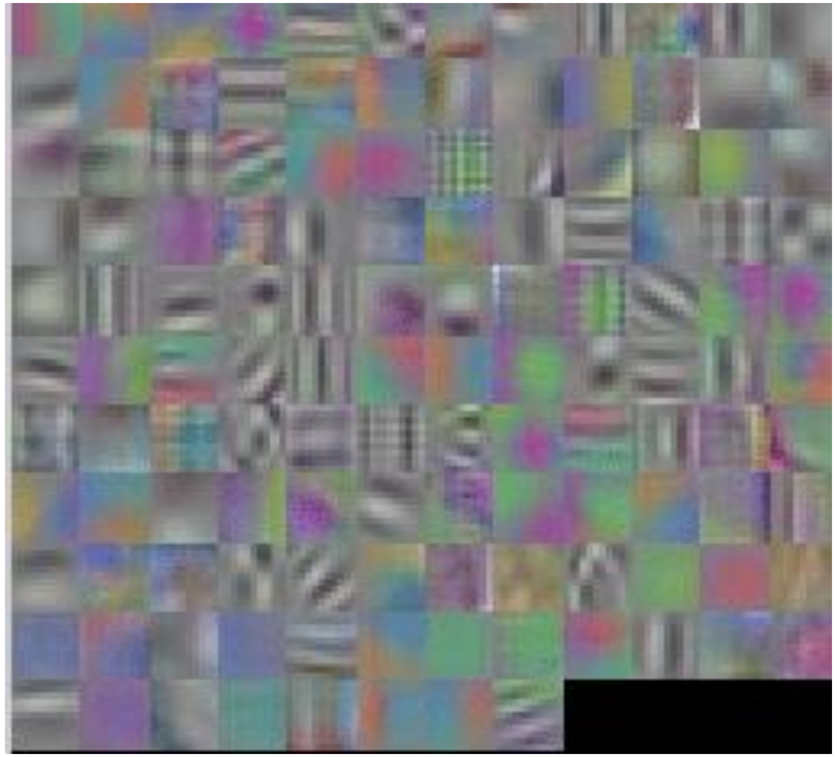
- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model
- **Slide the kernels in both space and time**



Input: T x 3 x H x W



Weights: C x t x 3 x h x w



First layer filters
3(rgb) x 4 (t) x 5 (h) x 5 (w)

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

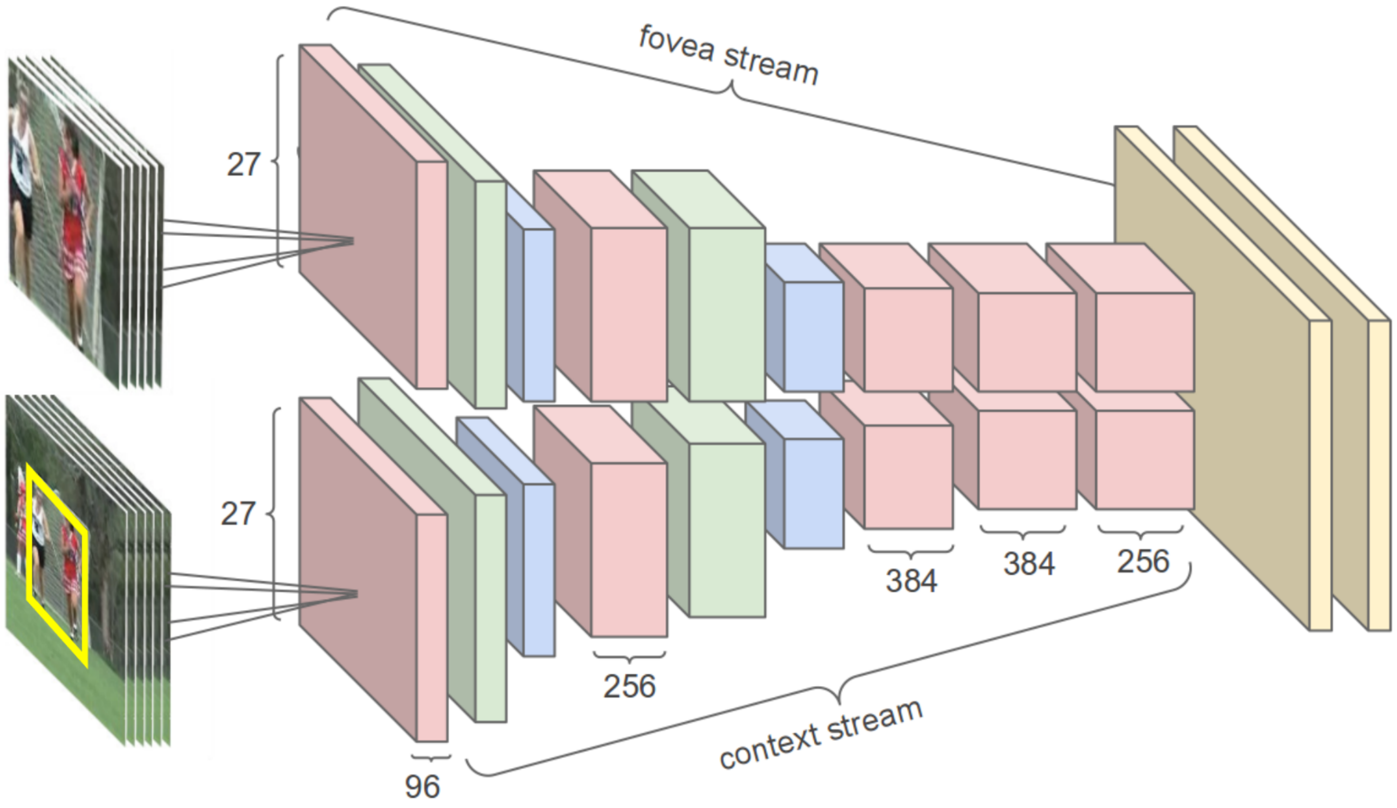
[redacted]

Models for Videos: Multi-scale

How can we reduce computational cost while maintaining accuracy?

Reduce video resolution → lower performance

Reduce network’s capacity → lower performance



- Context stream ***(low res)***:
  process low res video frames (H/2, W/2)

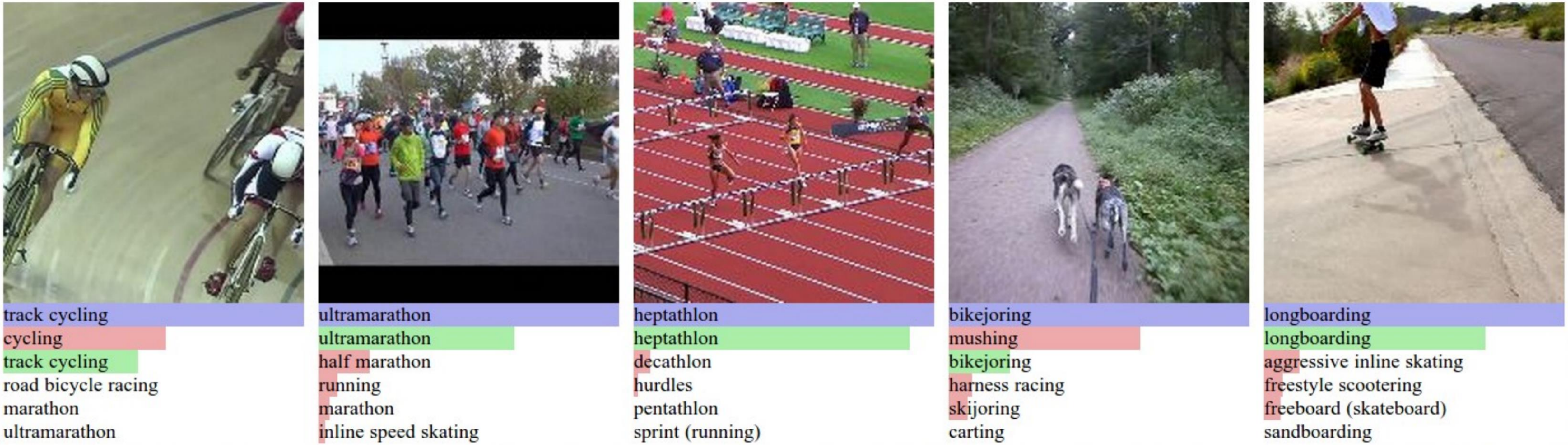
- Fovea sterm ***(high res)***:
  process a (H/2, W/2) crop from the original resolution

Reduce the input dimentionalty by half

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

[redacted]

Action classification -- Sports-1M



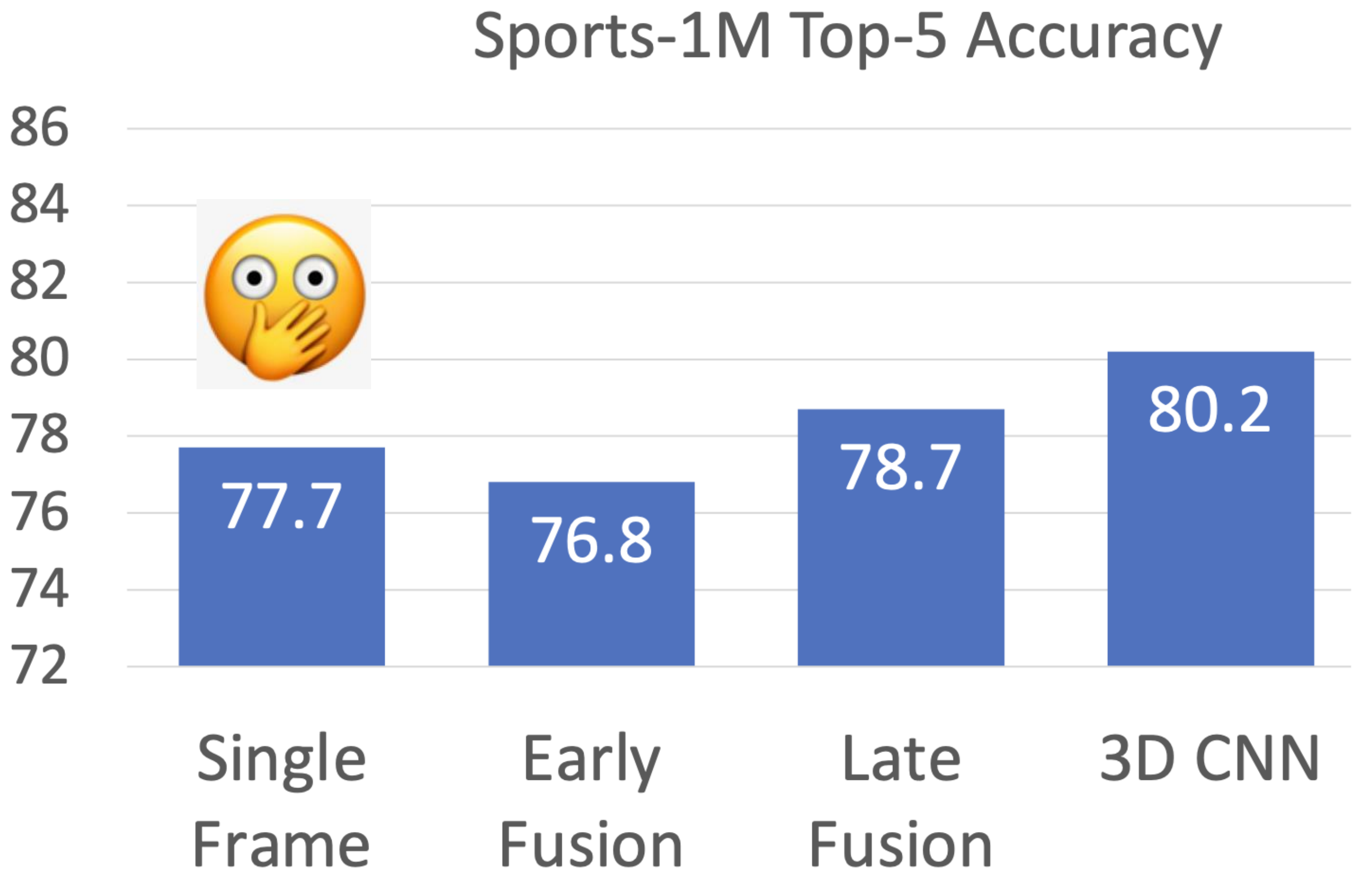
- **1 million YouTube videos**
- **Fine grained labels for 487 different types of sports**

- **Ground truth**
- **Correct prediction**
- **Incorrect prediction**

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

[redacted]

Action classification -- Sports-1M



Single frame: a shockly powerful baseline

This is from 2014…

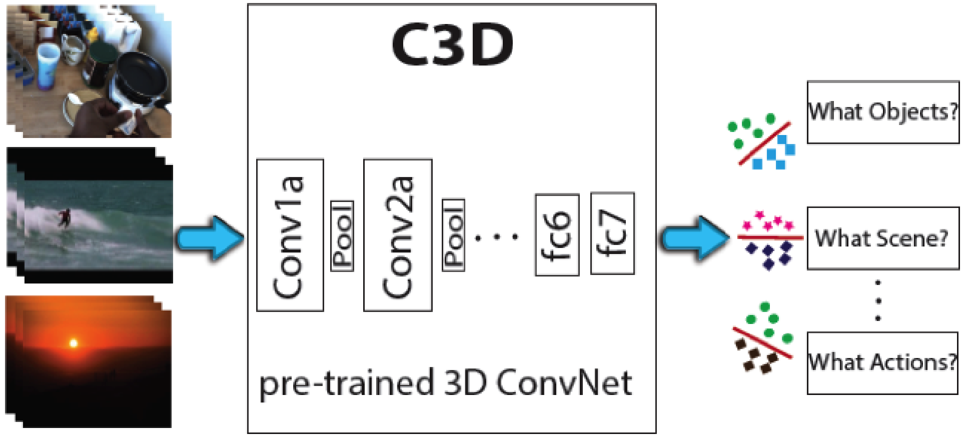
Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014 Slide credit: Justin Johnson, EECS 498-007

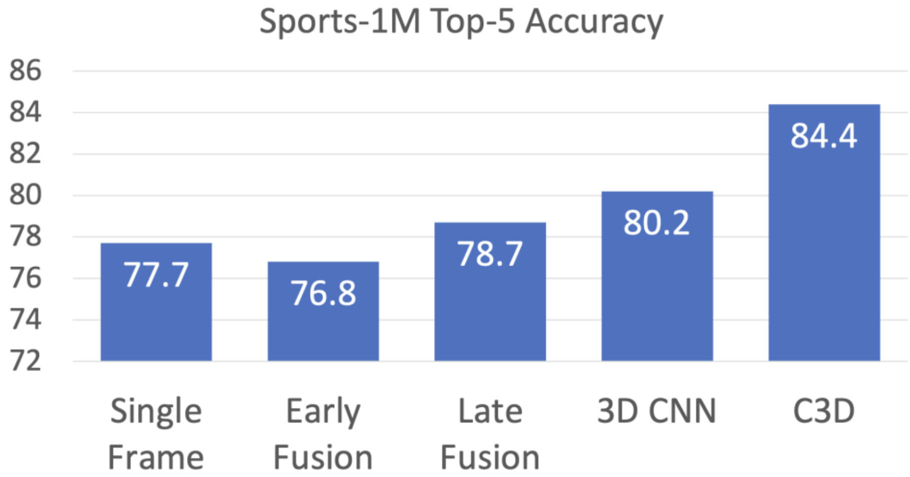
[redacted]

Models for Videos: C3D (Convolutional 3D)

- 3D CNN that uses all 3x3x3 Convs and 2x2x2 poolings
- The “VGG” of 3D CNNs

- Transfer learning: extract learned video features, train a simple linear classifier for various tasks





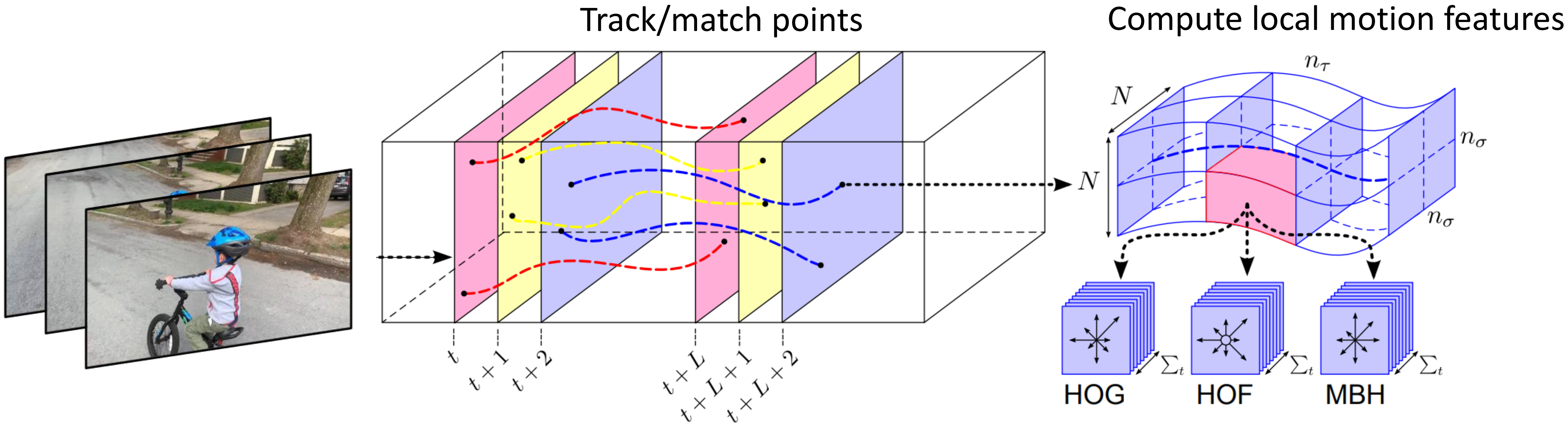
- Problem: 3D convs are VERY expensive!
  C3D on small inputs takes 3x VGG and 56x AlexNet FLOPs

Tran et al, “Learning Spatiotemporal Features with 3D Convolutional Networks”, ICCV 2015

[redacted]

Non-deep learning video classification

Motion is the most informative cue for action recognition → design hand crafted motion features:



Track/match points

Compute local motion features

Aggregate local motion features to compute a global representation of the video → linear SVM for action recognition

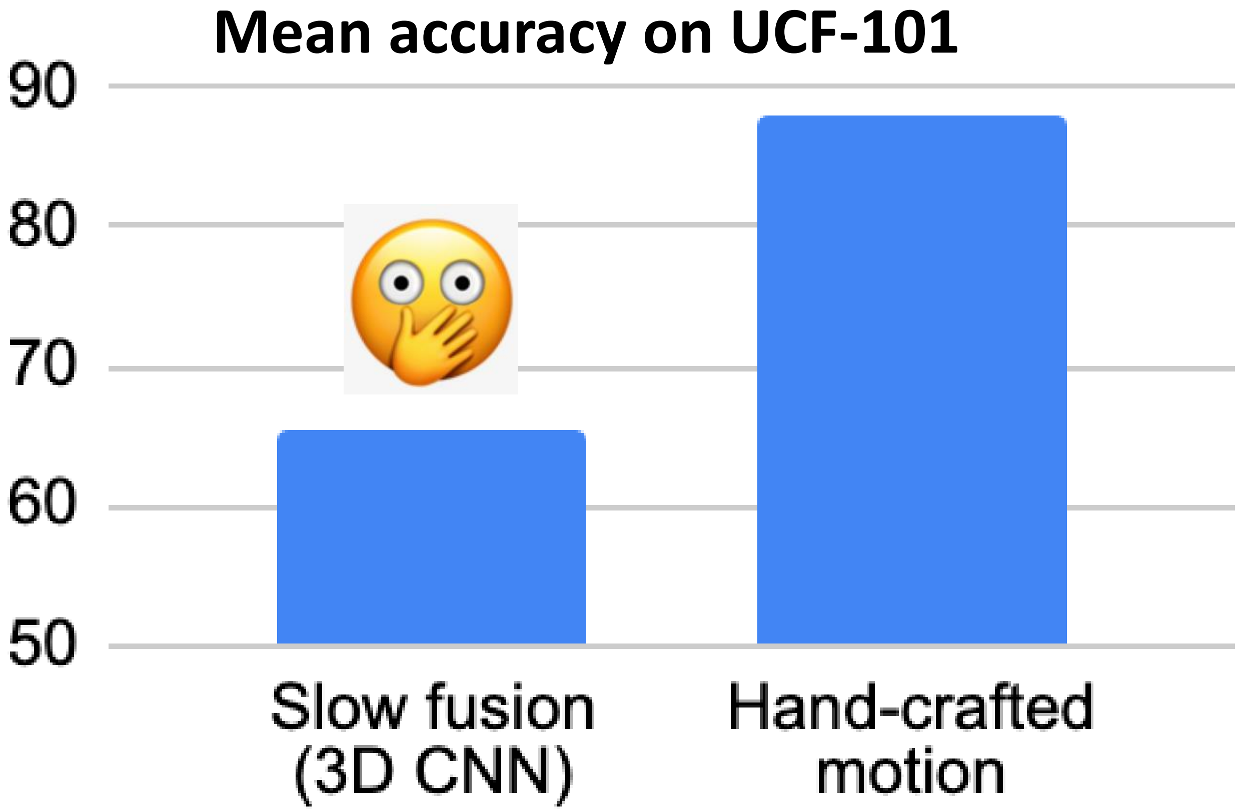
MODEL MOTION EXPLICITLY

Wang et. al., Dense trajectories and motion boundary descriptors for action recognition, 2013
Peng et. al., Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice, 2014

[redacted]

Non-deep learning video classification

Motion is the most informative cue for action recognition → hand crafted motion features:

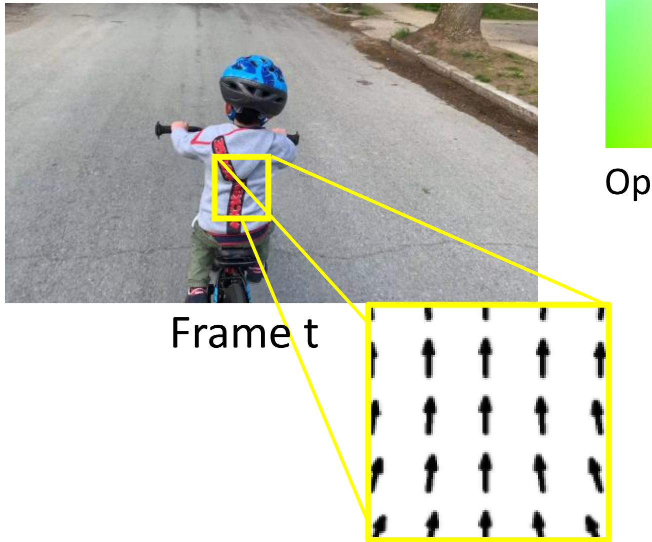


[redacted]

Explicitly modeling motion in deep-based models

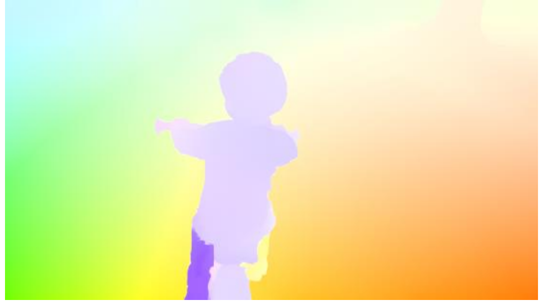
Optical flow: For each pixel in frame t, determines its corresponding pixel in frame t+1

Frame t+1

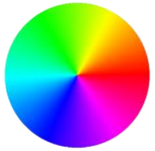


Frame t

Optical flow provides **local motion cues**



Optical flow between two frames

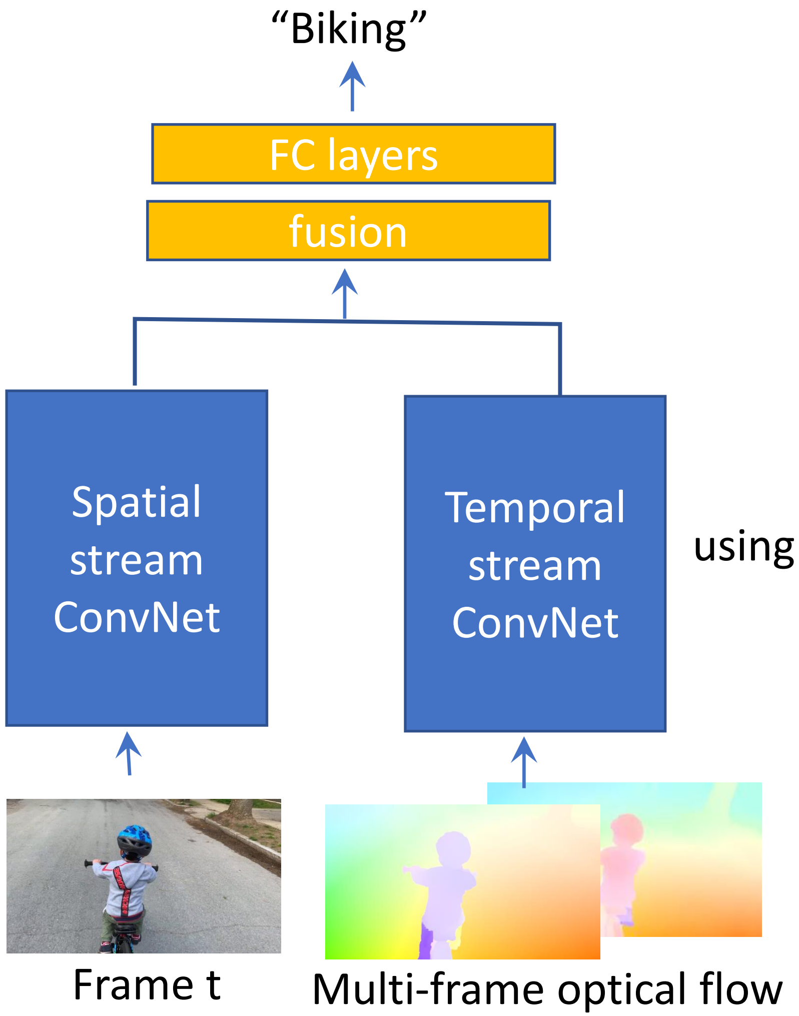


Color wheel
Saturation = mag.
Color = angle

[redacted]

Two Stream Networks: modeling motion explicitly

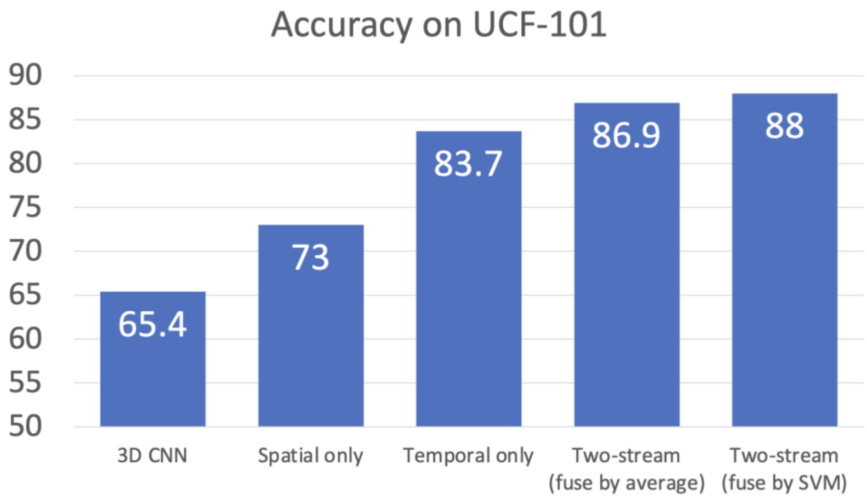
Idea: separate motion (multi-frame) from static appearance (single frame)



”Single Frame” baseline

using "Early Fusion" baseline

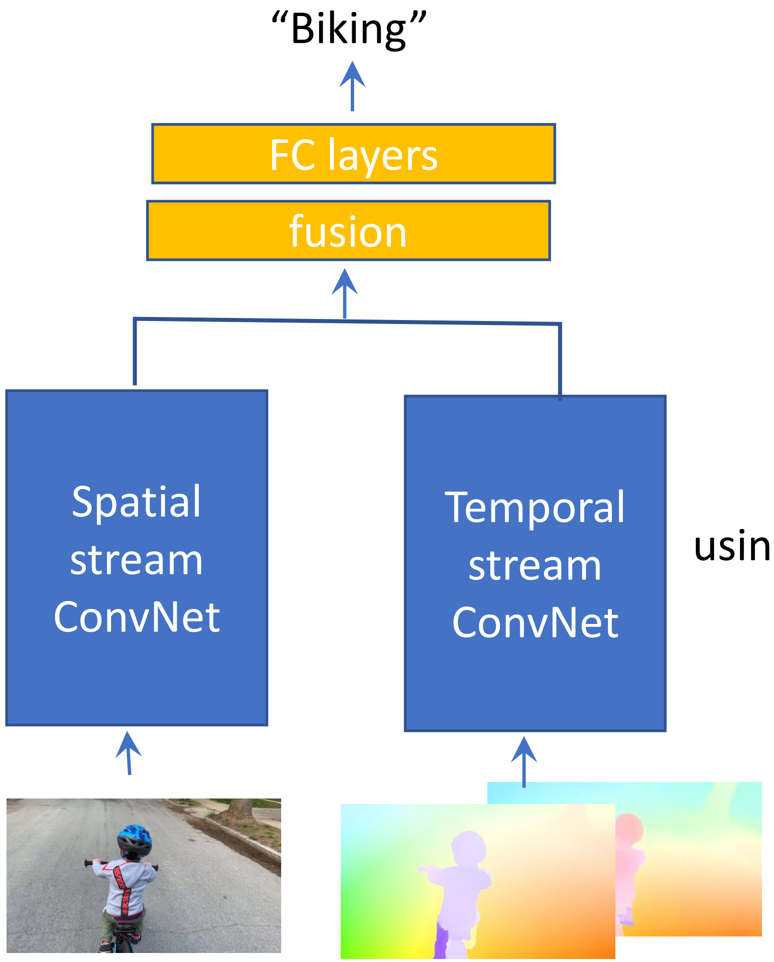
Precomputed flow fields between concuective frames



Simonyan and Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NIPS 2014

Two Stream Networks: modeling motion explicitly

Idea: separate motion (multi-frame) from static appearance (single frame)



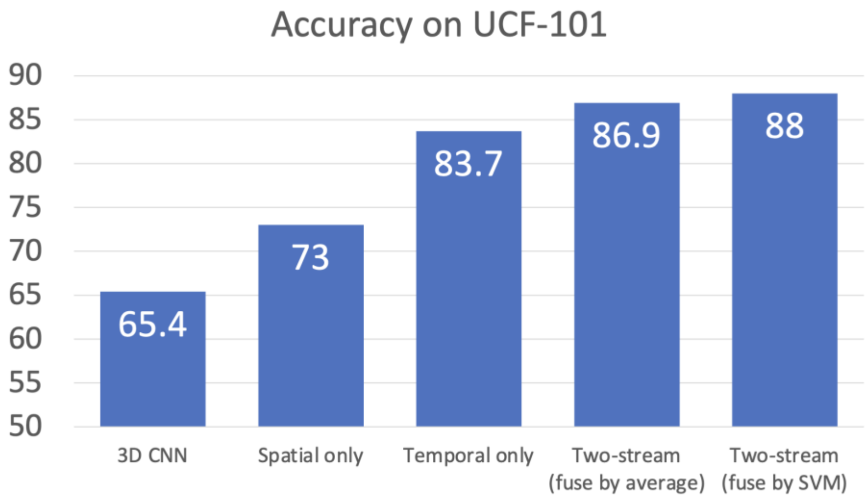
"Single Frame" baseline

using "Early Fusion" baseline

Frame t

Multi-frame optical flow

Precomputed flow fields between concuective frames



Simonyan and Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NIPS 2014

[redacted]

Additional models

Inflating 2D networks to 3D (I3D)

Take an existing 2D CNN model → convert it to a 3D CNN model

Transfer the weights from 2D and 3D

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

Long range temporal processing

Use LSTMs and RNNs to model long range temporal information

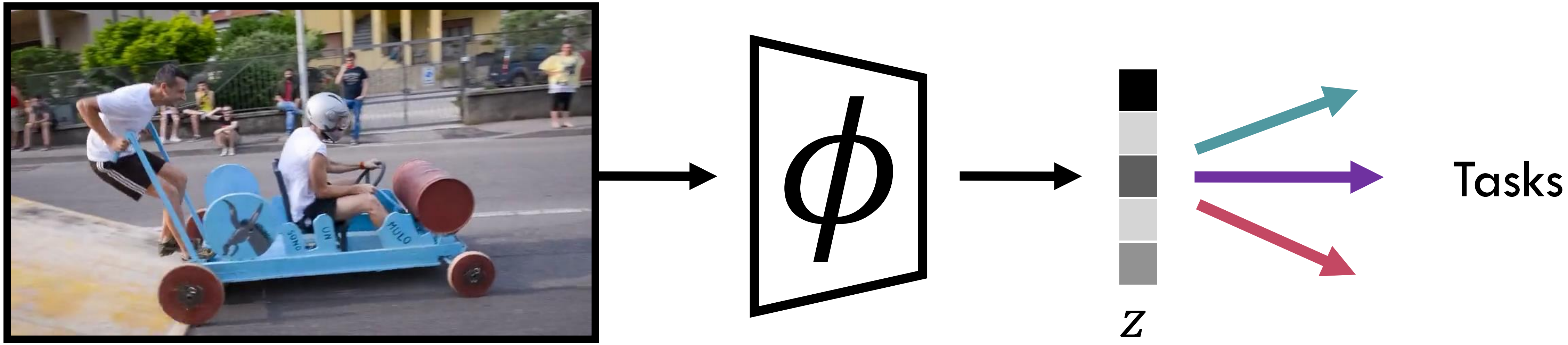
Baccouche et al, "Sequential Deep Learning for Human Action Recognition”, 2011 Donahue et al, “Long-term recurrent convolutional networks for visual recognition and description”, CVPR 2015

Long range temporal processing

Self attention, non-local networks, Transformers

Self-Supervision in Videos

[redacted]



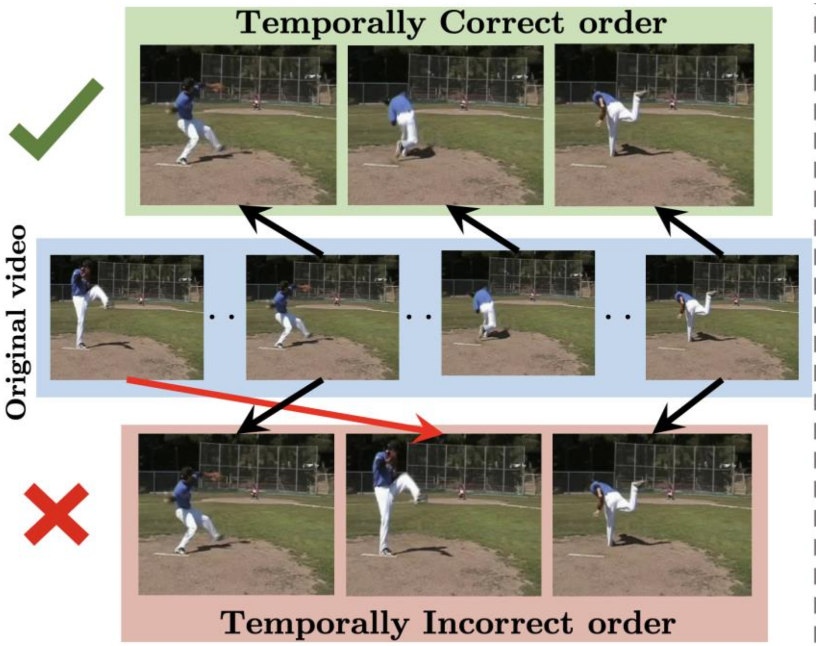
- Temporal order
- Cycle consistency
- Video Speedup
- Video colorization

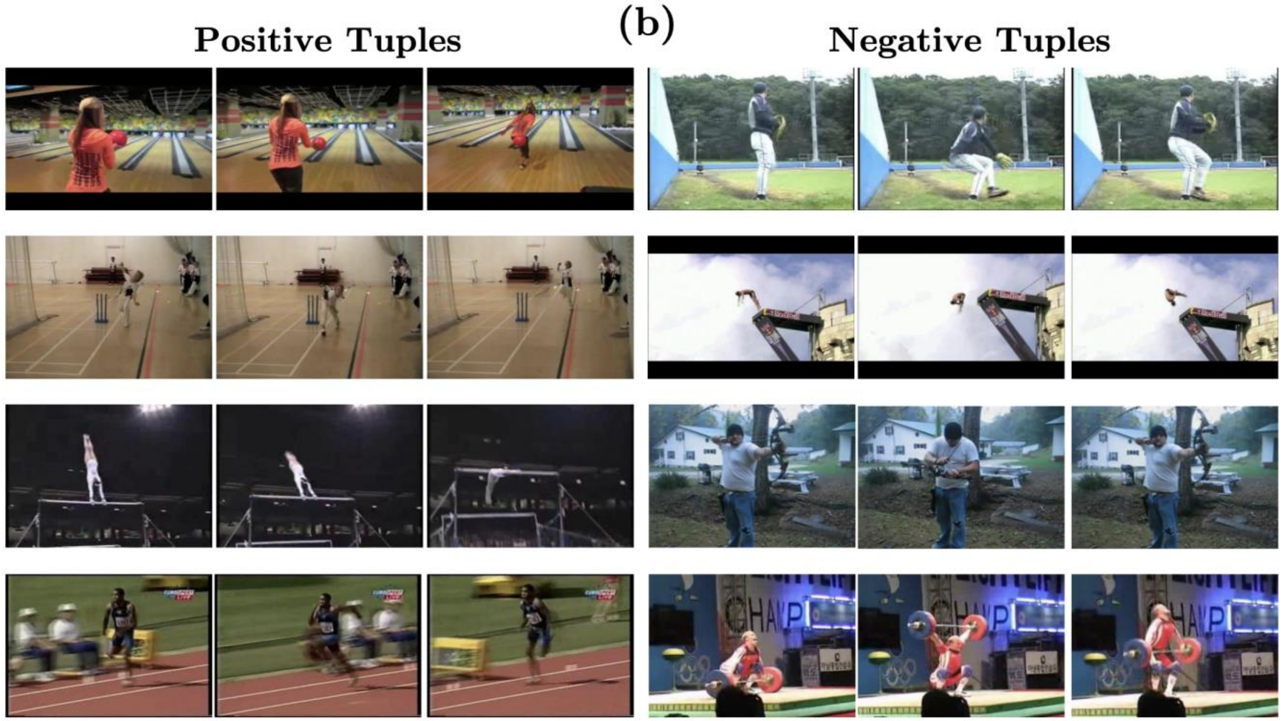
Video: https://ajabri.github.io/timecycle/

[redacted]

Self-Supervision in Videos: frame ordering

Training data: shuffled video frames, original video frames
Pretext task: predict if the frames are in the correct temporal order (binary classification task)

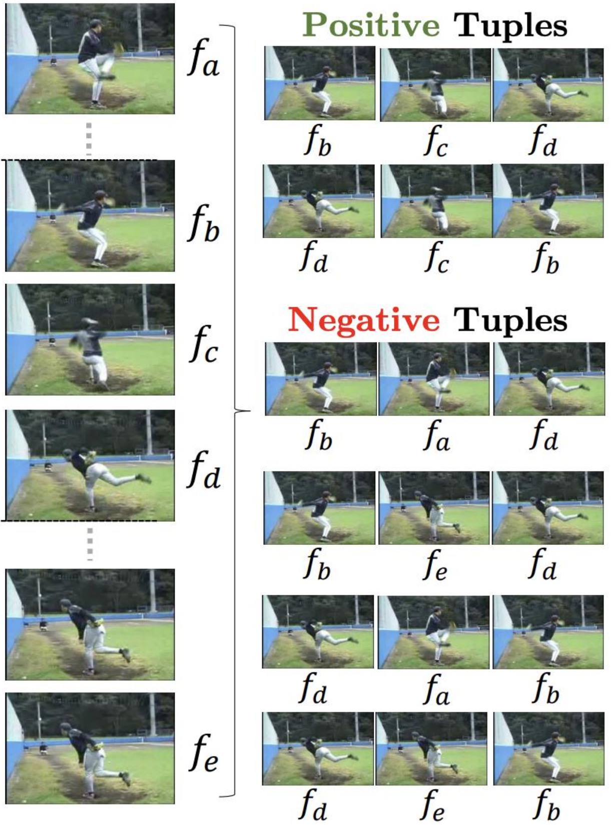




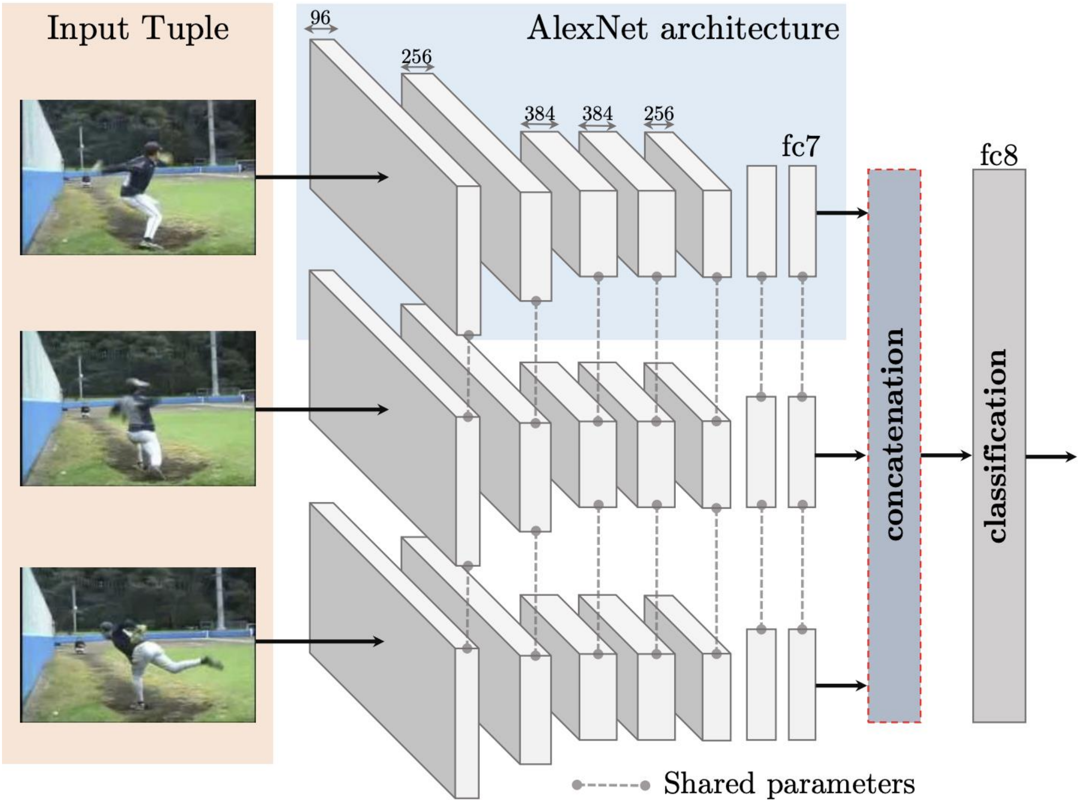
Misra et. al., Shuffle and Learn: Unsupervised Learning using Temporal Order Verification, ECCV 2016

[redacted]

Self-Supervision in Videos: frame ordering



Generating positive and negative examples



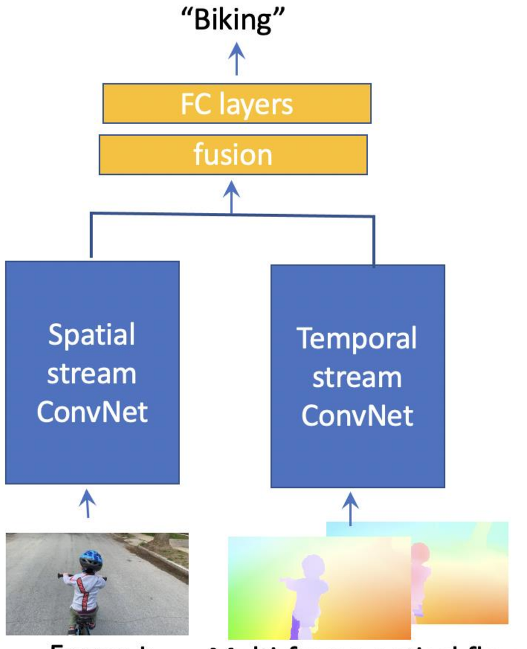
Triplet Siamese network for sequence verification

Misra et. al., Shuffle and Learn: Unsupervised Learning using Temporal Order Verification, ECCV 2016

[redacted]

Self-Supervision in Videos: frame ordering

Transfer learning: fine-tune spatial stream for video classification



Frame t

Multi-frame optical flow

Dataset	Initialization	Mean Accuracy
UCF101	Random	38.6
	(Ours) Tuple verification	**50.2**
HMDB51	Random	13.3
	UCF Supervised	15.2
	(Ours) Tuple verification	**18.1**

Misra et. al., Shuffle and Learn: Unsupervised Learning using Temporal Order Verification, ECCV 2016

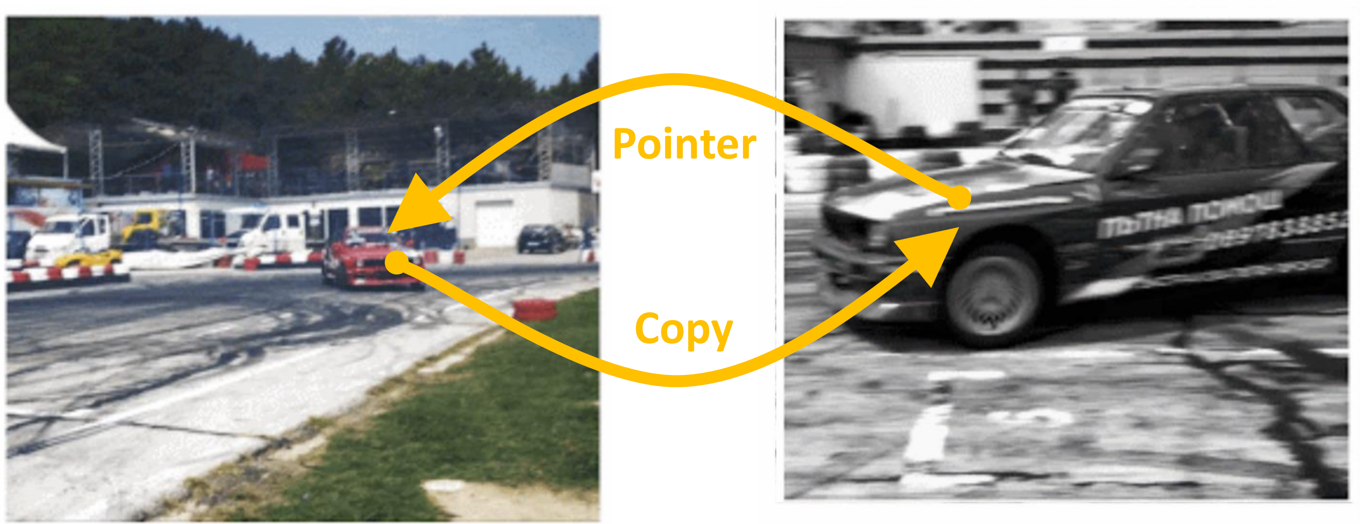
[redacted]

Self-Supervision in Videos: Colorization of Moving Objects

Ultimate goal: Tracking

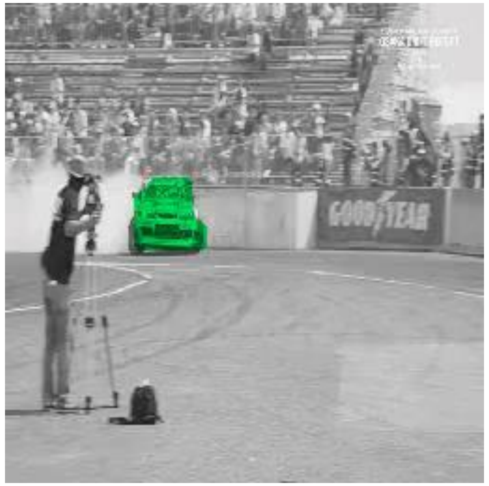
Pretext task: video colorization by learning to copy color from a reference frame

Training data: grayscale videos + original color videos



Reference frame

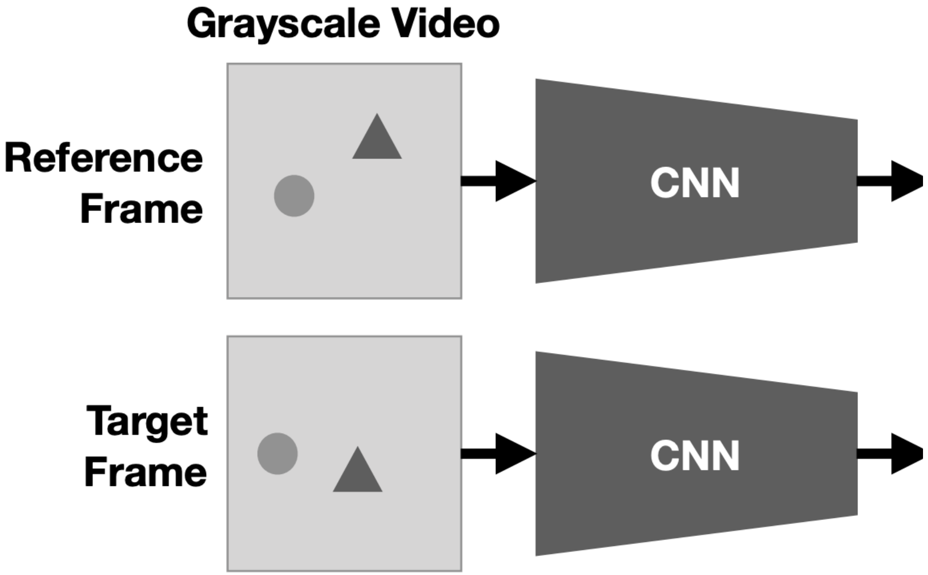
Grayscale video



Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018

Self-Supervision in Videos: Colorization of Moving Objects

Video colorization by learning to copy color from a reference frame



Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018

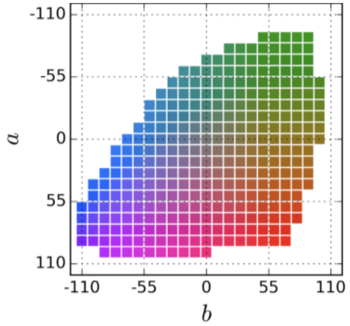
[redacted]

Self-Supervision in Videos: Colorization of Moving Objects

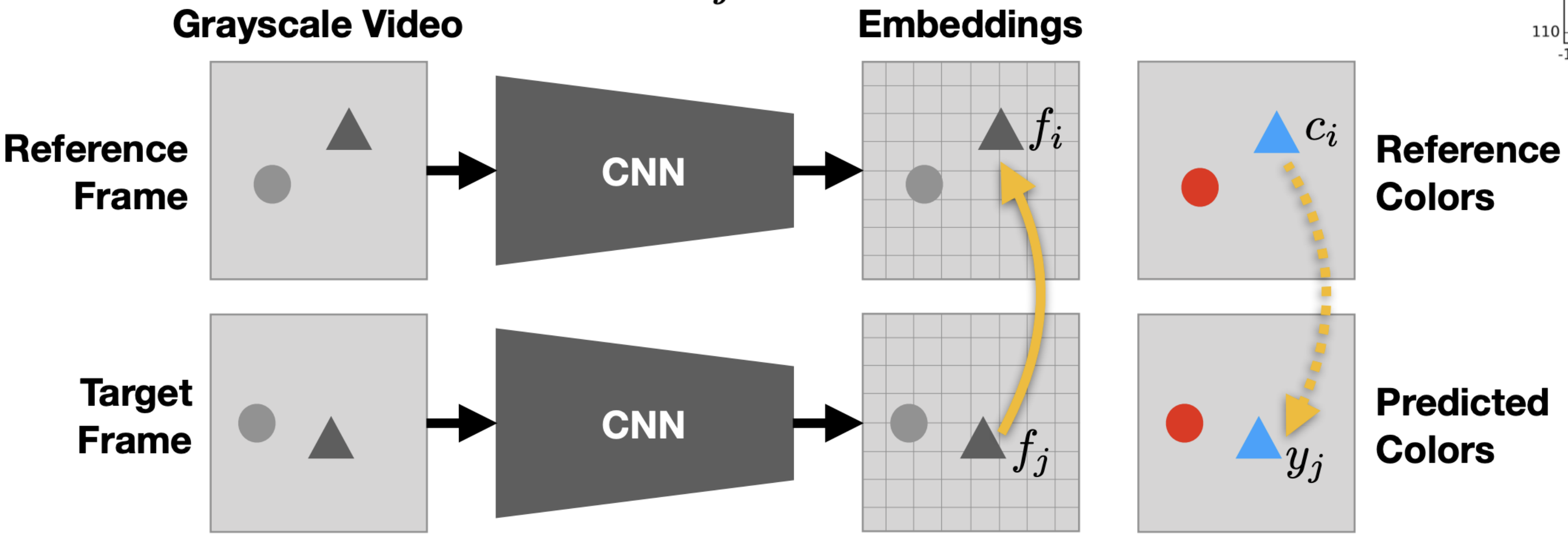
Video colorization by learning to copy color from a reference frame

$$\min_{\theta} \sum_{j} \mathcal{L}\left(y_j, c_j\right)$$

cross-entropy over color distribution



Colors in *ab* space (discrete)



affinity matrix between reference and target (rows sum to one)

$$y_j = \sum_i A_{ij} c_i$$

linear combination of the reference colors

Similarity in the embedding space

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

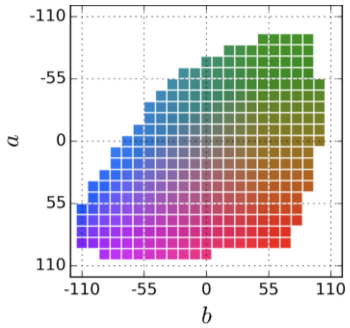
Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018

Self-Supervision in Videos: Colorization of Moving Objects

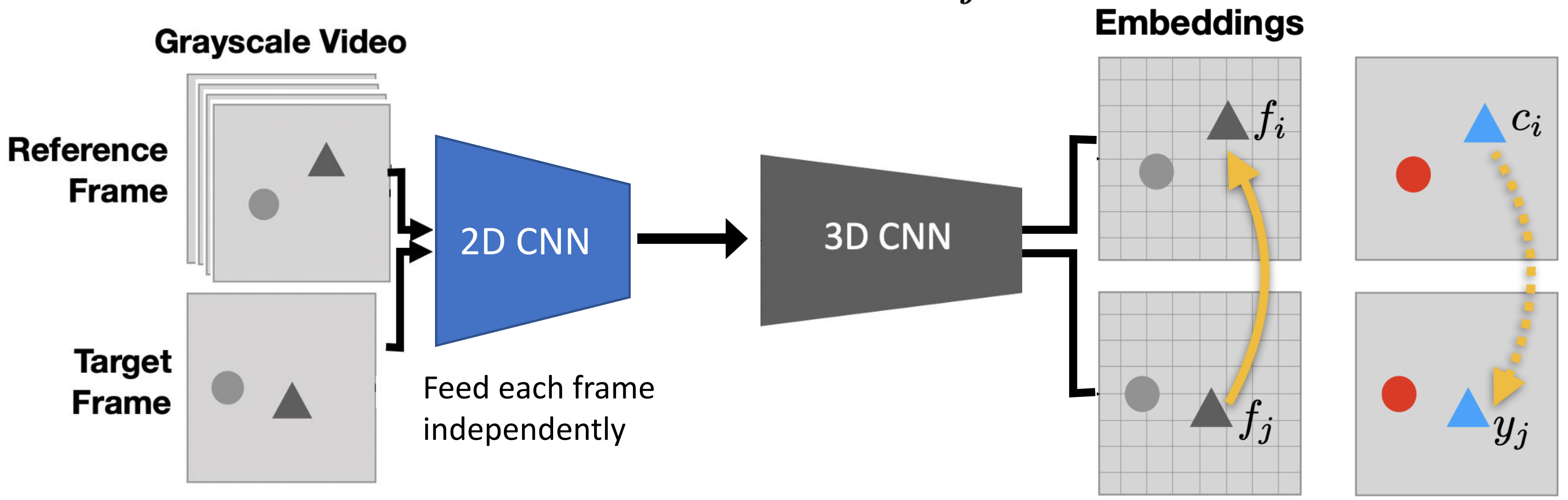
Video colorization by learning to copy color from a reference frame

$$\min_{\theta} \sum_{j} \mathcal{L}\left(y_j, c_j\right)$$

cross-entropy over color distribution



Colors in *ab* space (discrete)



Feed each frame independently

Reference Colors

Predicted Colors

A -- a similarity matrix between reference and target (rows sum to one)

$$y_j = \sum_i A_{ij} c_i$$

linear combination of the reference colors

Similarity in the embedding space

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018

[redacted]

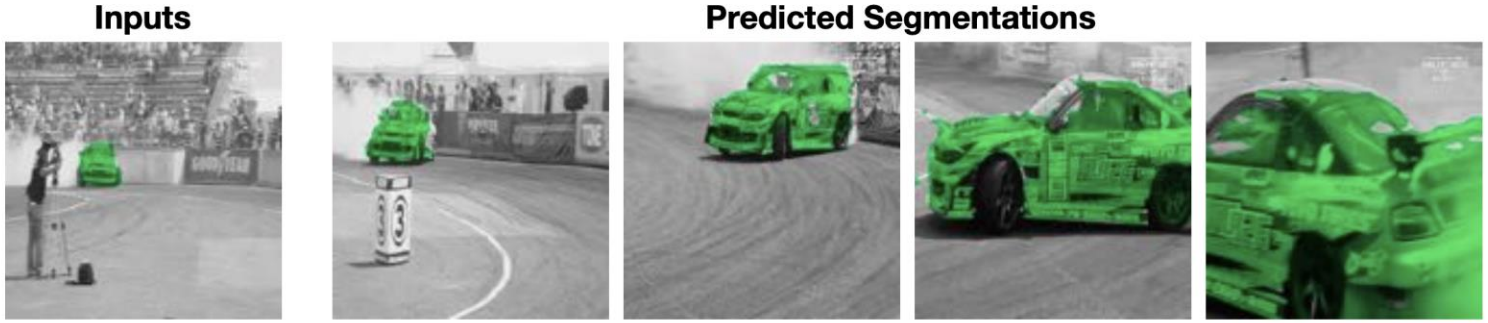
Self-Supervision in Videos: Colorization of Moving Objects

Video colorization by learning to copy color from a reference frame

linear combination of the reference colors

$$y_j = \sum_i A_{ij} c_i$$

a similarity matrix between reference and target (rows sum to one)



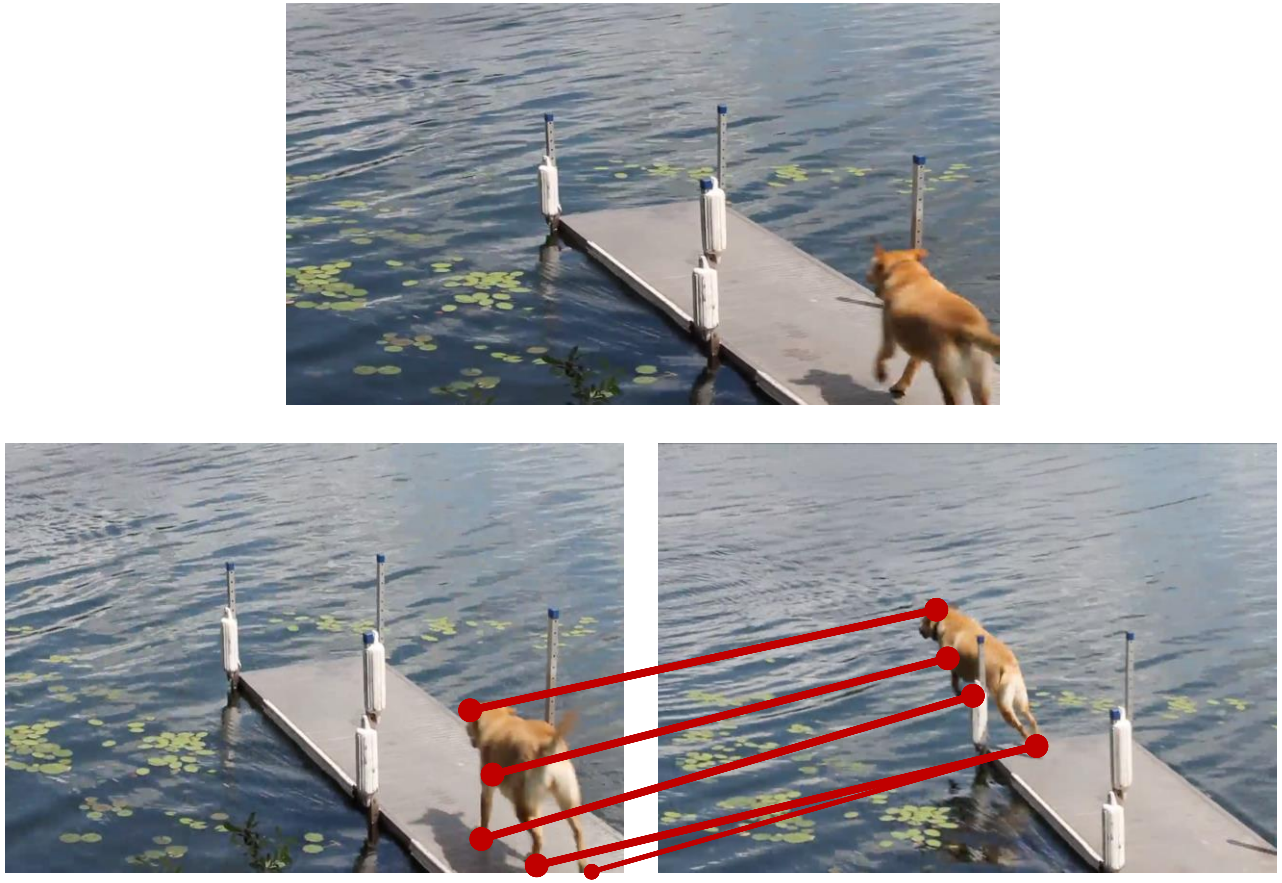
Reference

Held-out video

Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018

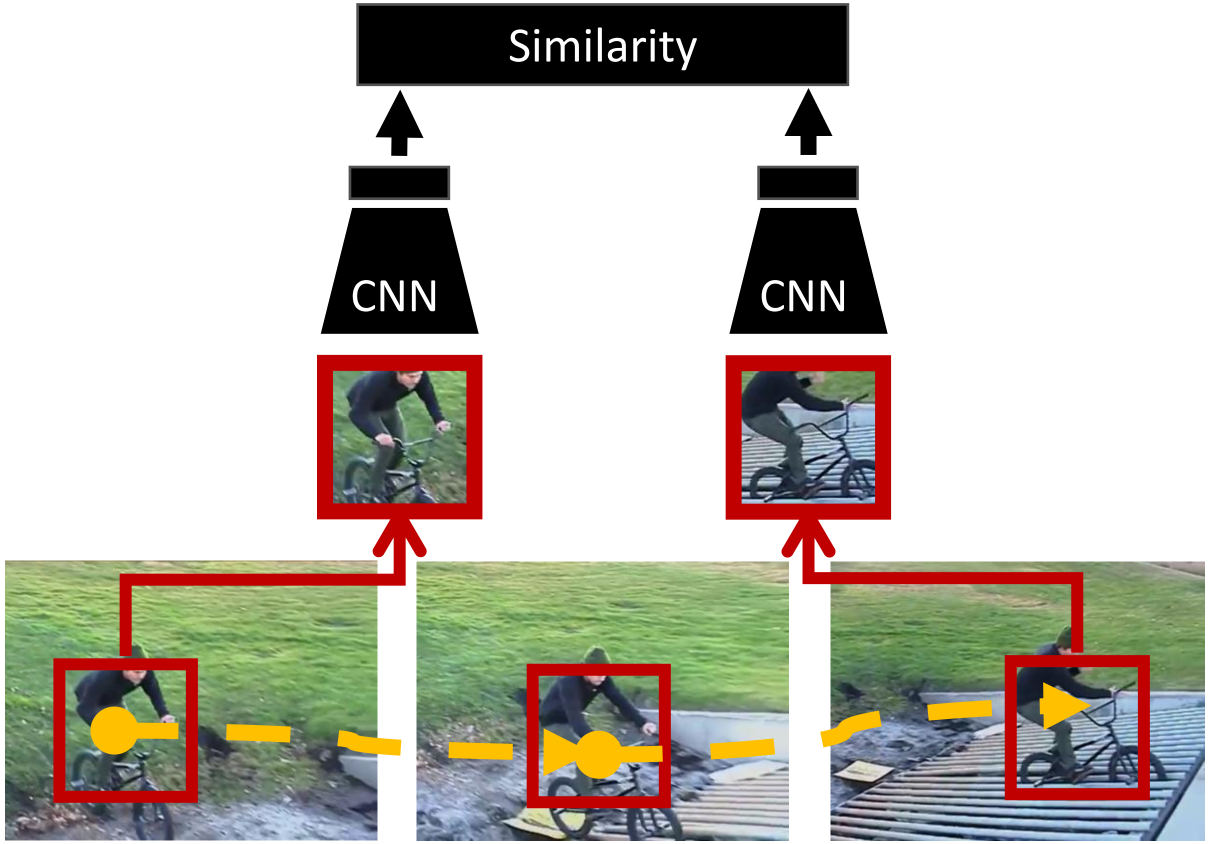
Self-Supervision in Videos: Learning correspondence

Ultimate goal: Correspondence



Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

Learning Similarity from Tracking



Tracking → Similarity

[Wang et al, 2015; Pathak et al, 2017]

Slide credit: https://ajabri.github.io/timecycle/

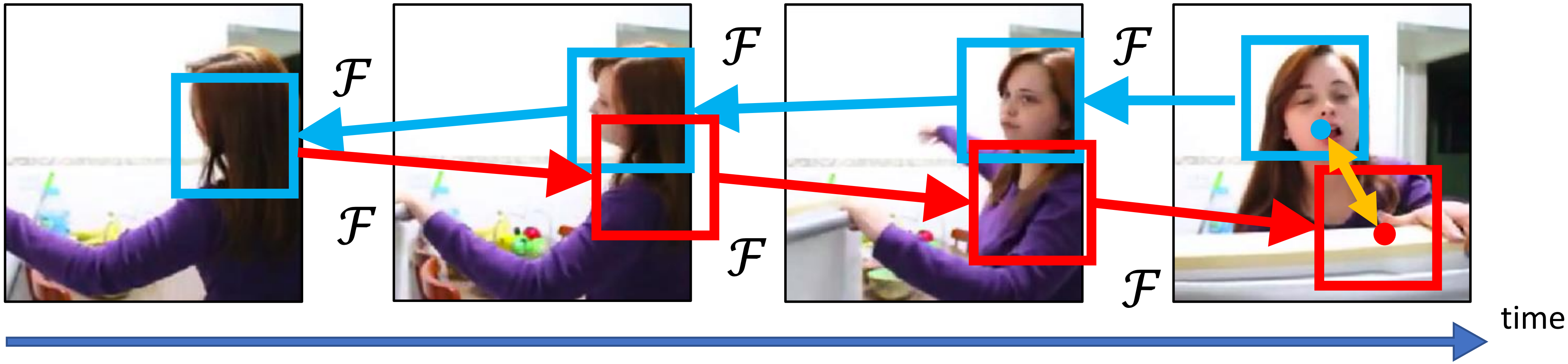
Self-Supervision in Videos: Learning correspondence

[redacted]

Ultimate goal: Correspondence, without using off-the-shelf tracking methods

How to obtain supervision?

Supervision: Cycle-Consistency in Time



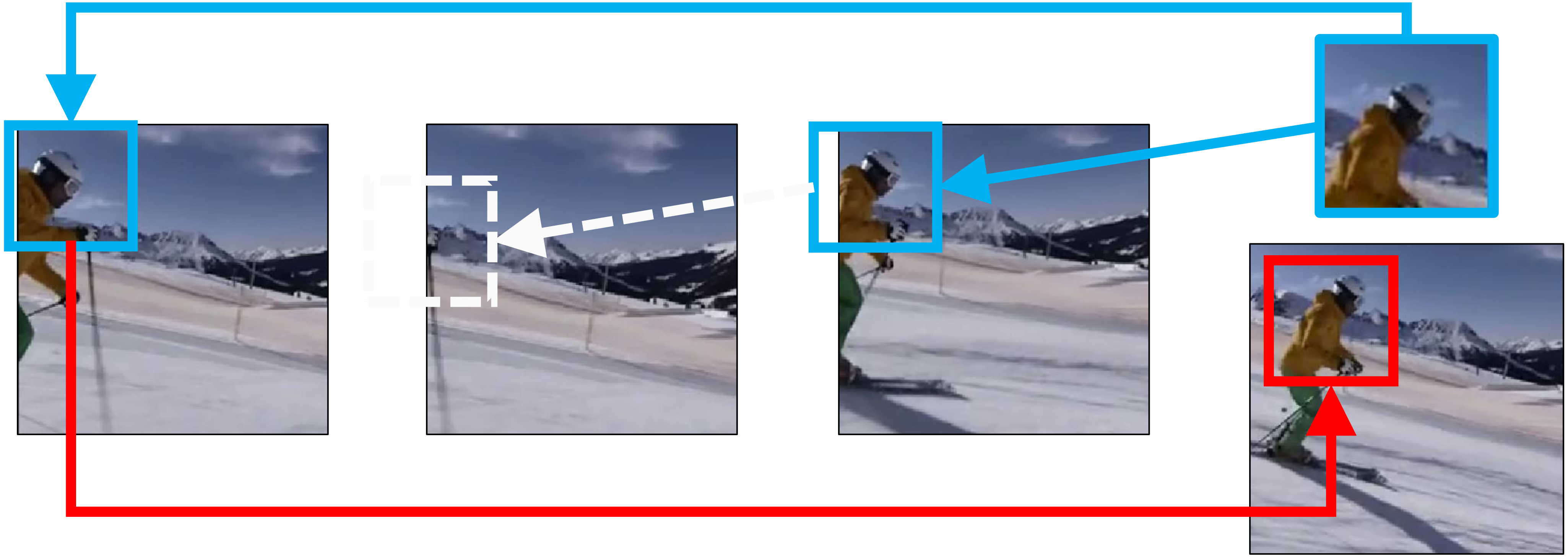
Track backwards in time

Track forwards, back to the future

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

Self-Supervision in Videos: Learning correspondence

Supervision: Cycle-Consistency in Time
Challenge: Occlusions



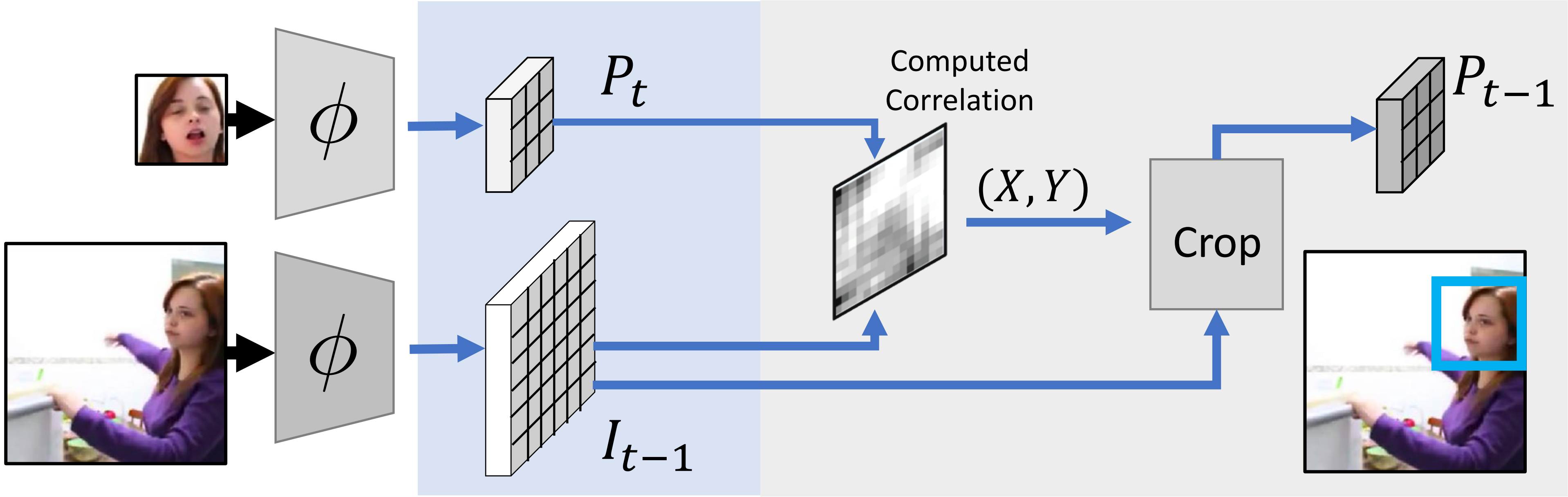
Skip-cycles: skipping occlusions

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

Self-Supervision in Videos: Learning correspondence

Differentiable tracker: densely match features in learned feature space

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

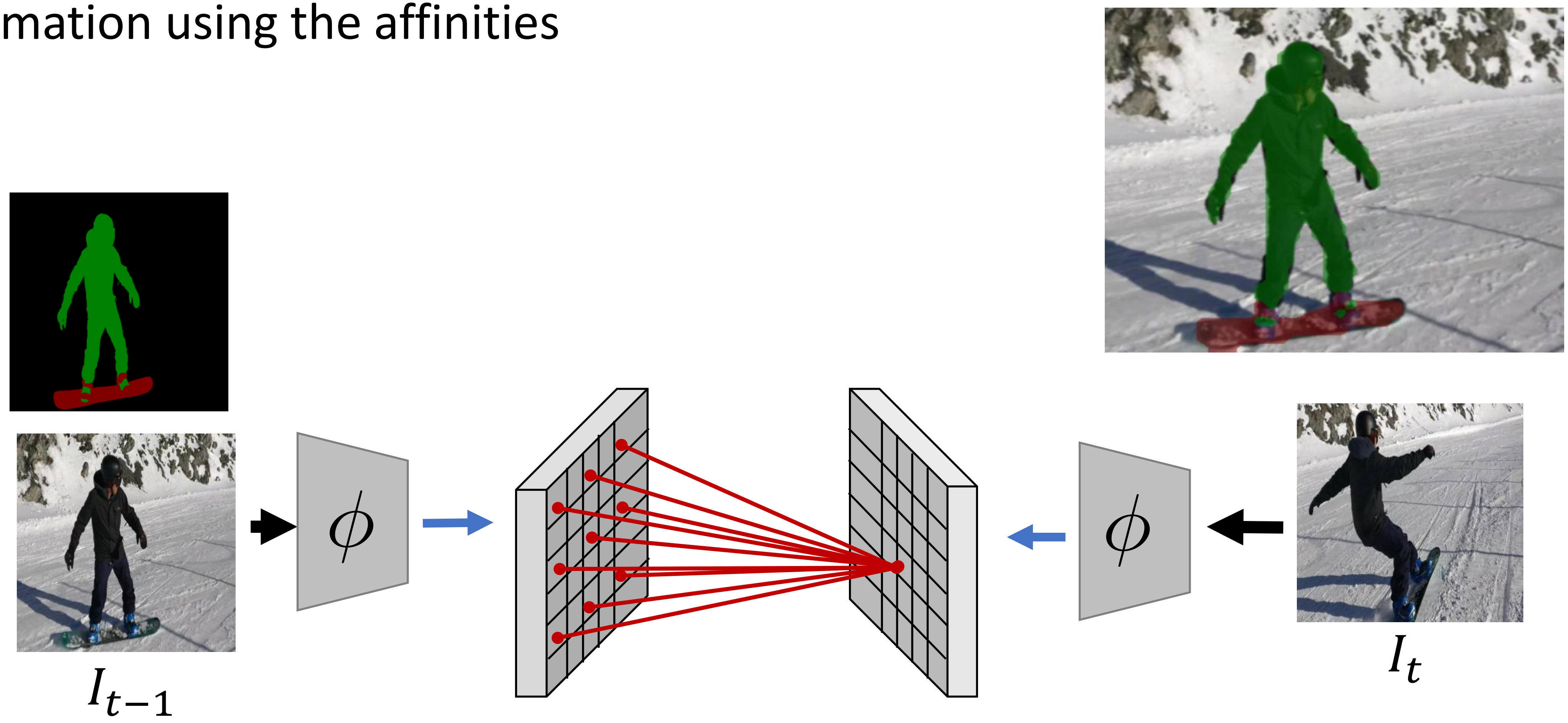


Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

[redacted]

Self-Supervision in Videos: Learning correspondence

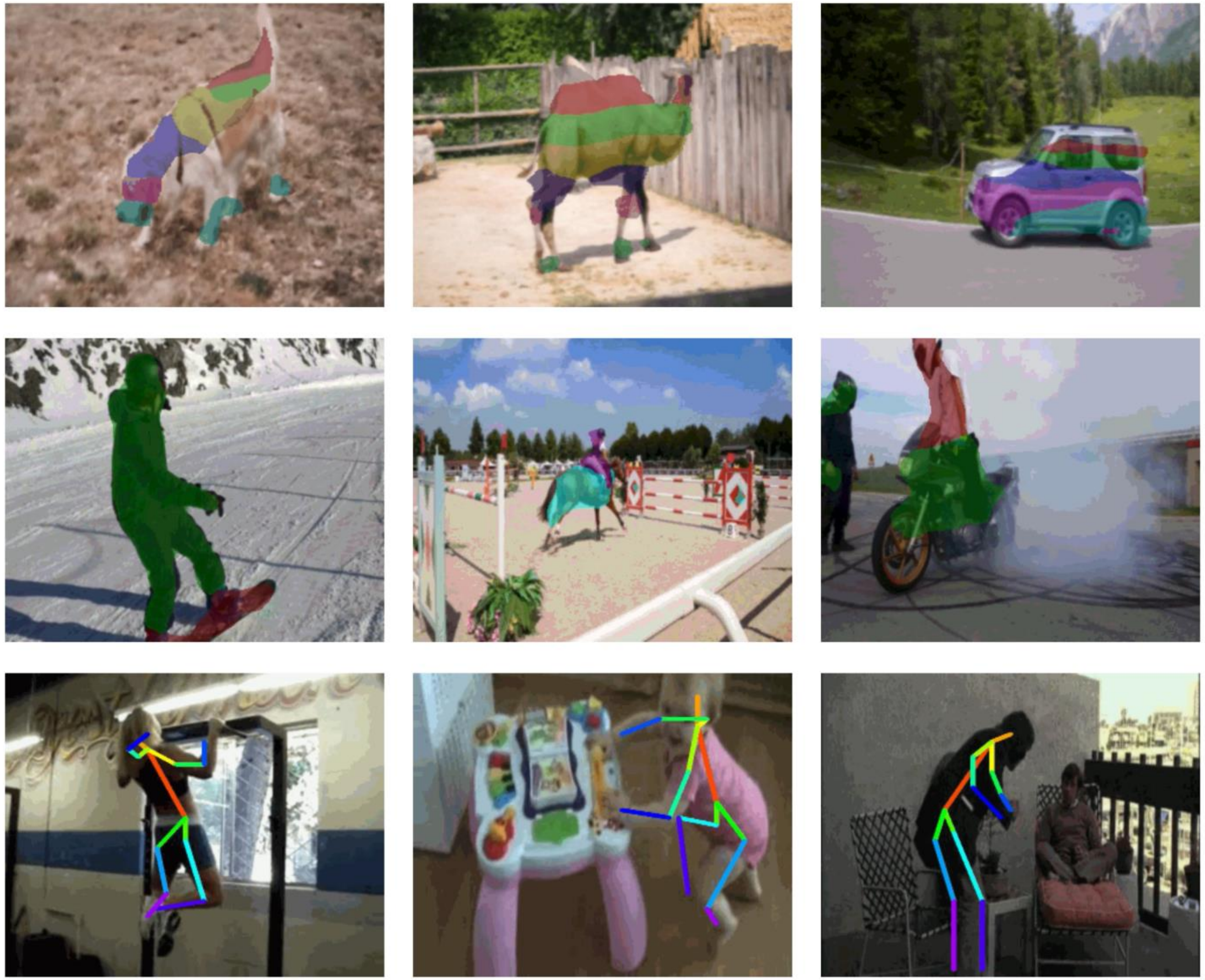
Test time: compute features to each frame, compute features affinity, propagate information using the affinities



Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

[redacted]

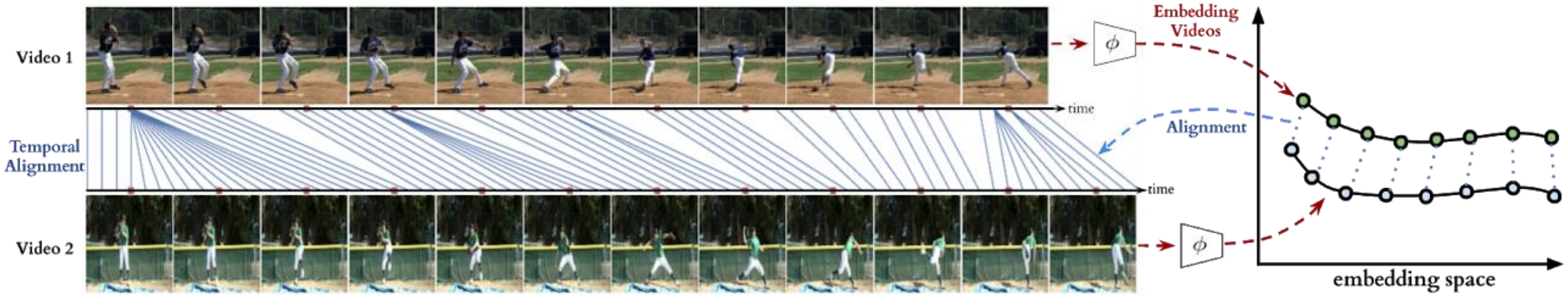
Self-Supervision in Videos: Learning correspondence



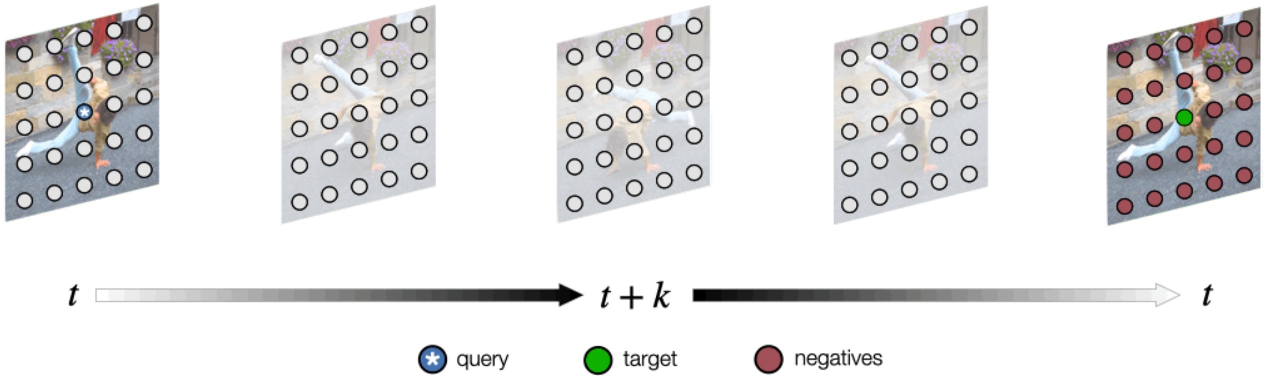
Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019

[redacted]

Self-Supervision in Videos: Temporal cycle consistency



Dwibed et. al. Temporal Cycle-Consistency Learning, CVPR’19

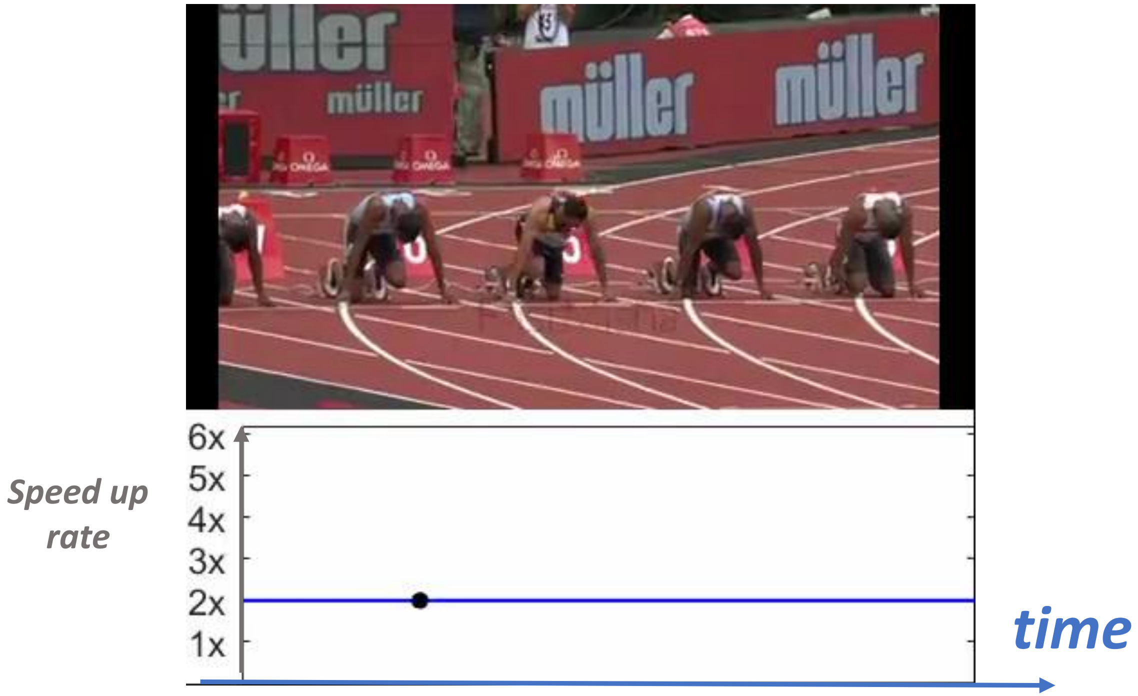


Jabri et. al, Space time correspondence as Contrastive Random Walk, NeurIPS 2020

Self-Supervision in Videos: Learning the Speediness in Videos

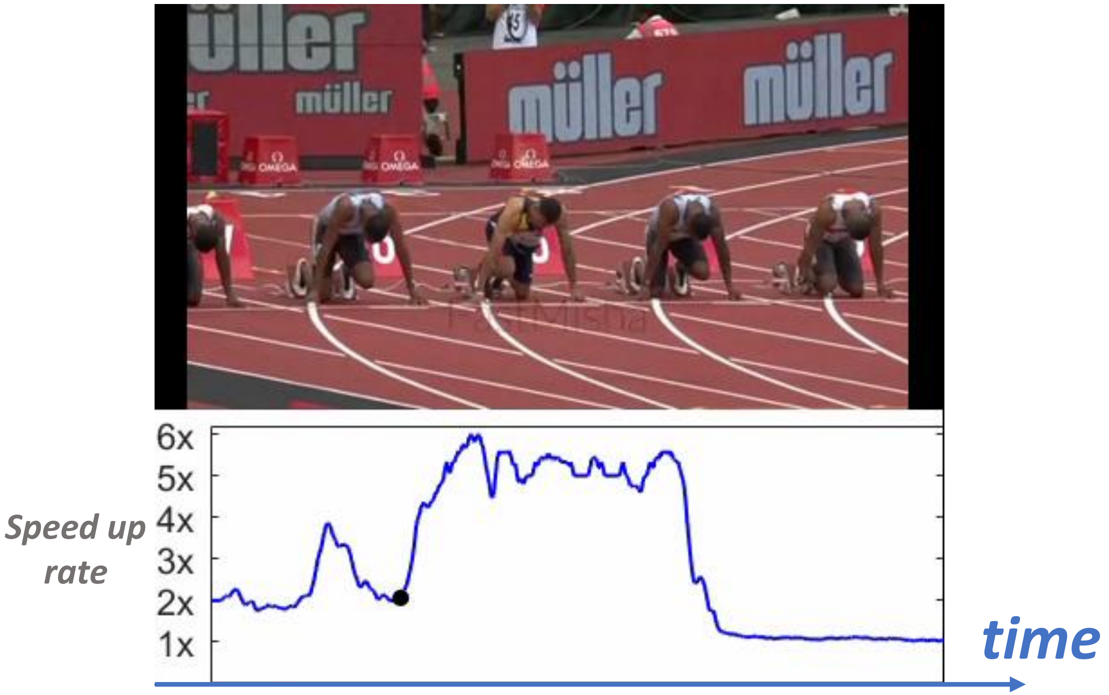
Ultimate goal: Watch video content faster by adaptively speeding up the video

Uniform Speed Up (2x)



Jittery, unnatural motions

Adaptive Speed Up (2x)



Same duration, more natural

Joint work with: Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, Bill Freeman, Miki Rubinstein and Michal Irani, CVPR 2020

[redacted]

“Speediness” in Videos

Slower

Normal speed

Faster

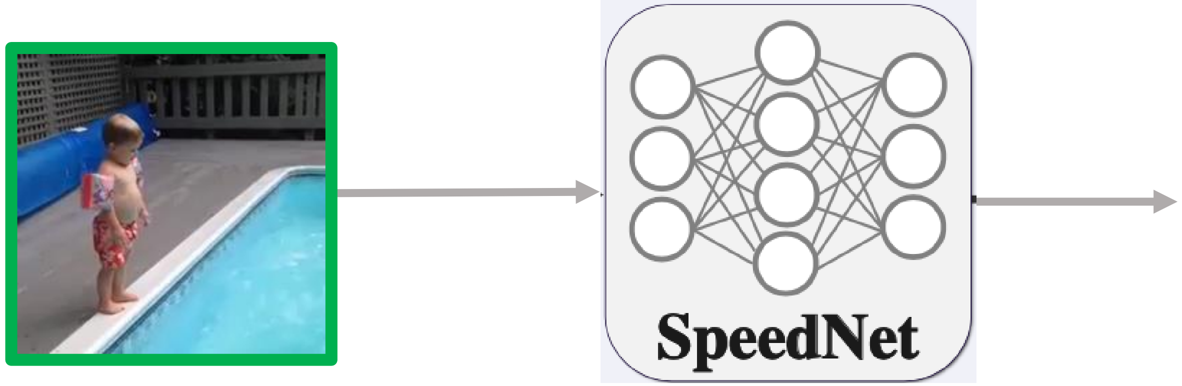
Joint work with: Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, Bill Freeman, Miki Rubinstein and Michal Irani, CVPR 2020

[redacted]

Self-Supervision in Videos: Learning the Speediness in Videos

Pretext task: Predict if a given video segment is sped up or not
Training data: sped up video segments + original video segments

Self supervised
training on Kinetics



Input segment
(30 frames)

Normal speed
or
Sped Up

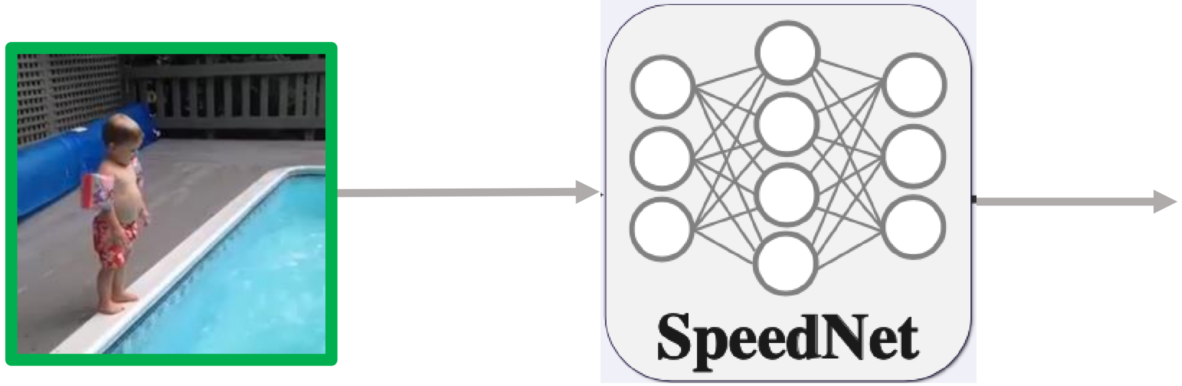
“Learning and Using the Arrow of Time”, Wei at. al, CVPR 2018

Self-Supervision in Videos: Learning the Speediness in Videos

Pretext task: Predict if a given video segment is sped up or not

Training data: sped up video segments + original video segments

Self supervised training on Kinetics



Input segment (30 frames)

Normal speed

or

Sped Up

Networks are lazy!

Learning properties of natural motion, avoid “easy cheats” → **very challenging!**

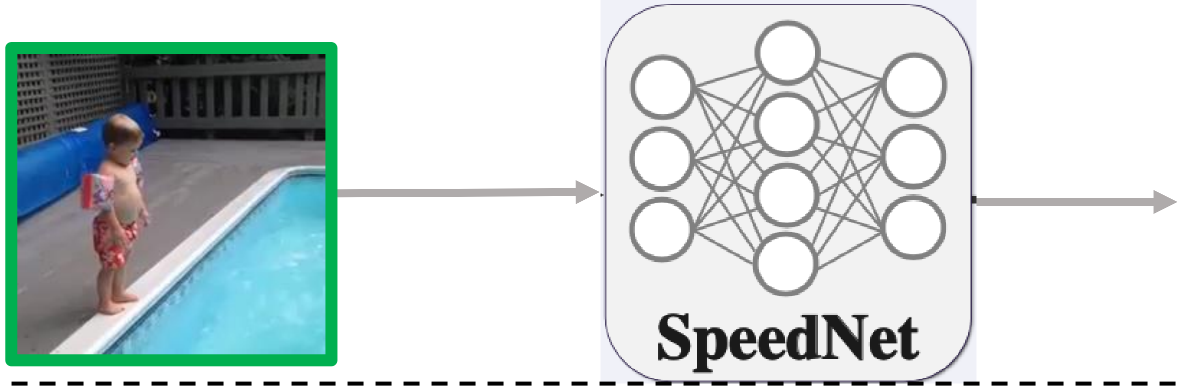
[redacted]

Self-Supervision in Videos: Learning the Speediness in Videos

Pretext task: Predict if a given video segment is sped up or not

Training data: sped up video segments + original video segments

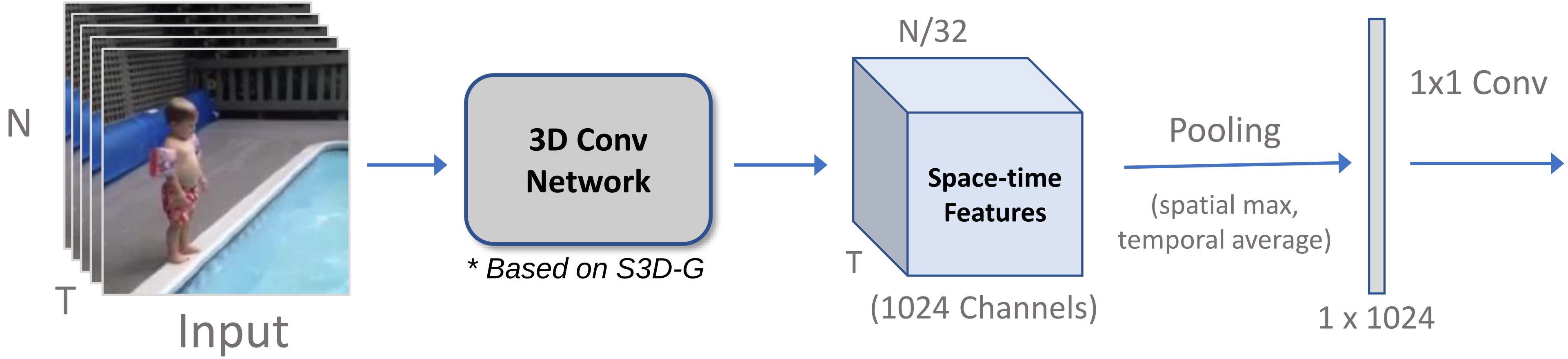
Self supervised training on Kinetics



Normal speed

or

Sped Up



Normal Speed

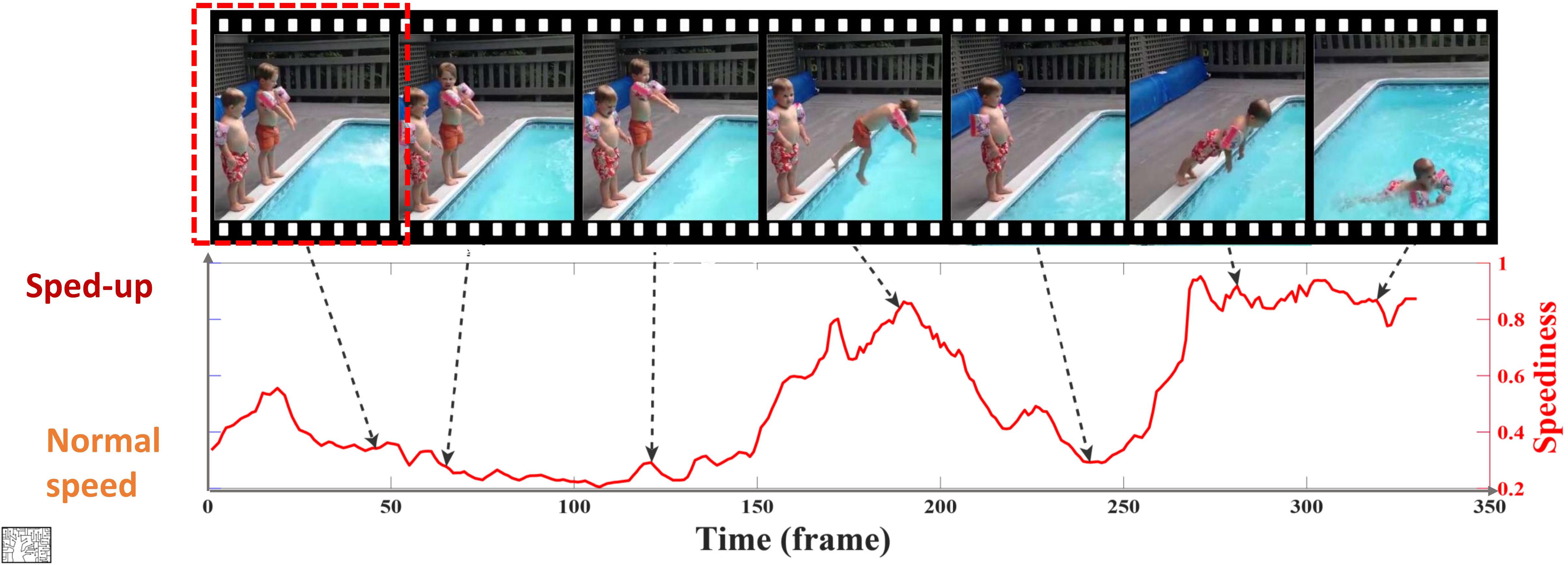
or

Sped Up

* *"Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification"*, Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy, ECCV'18.

Self-Supervision in Videos: Learning the Speediness in Videos

Inference: sliding window → prediction for every frame

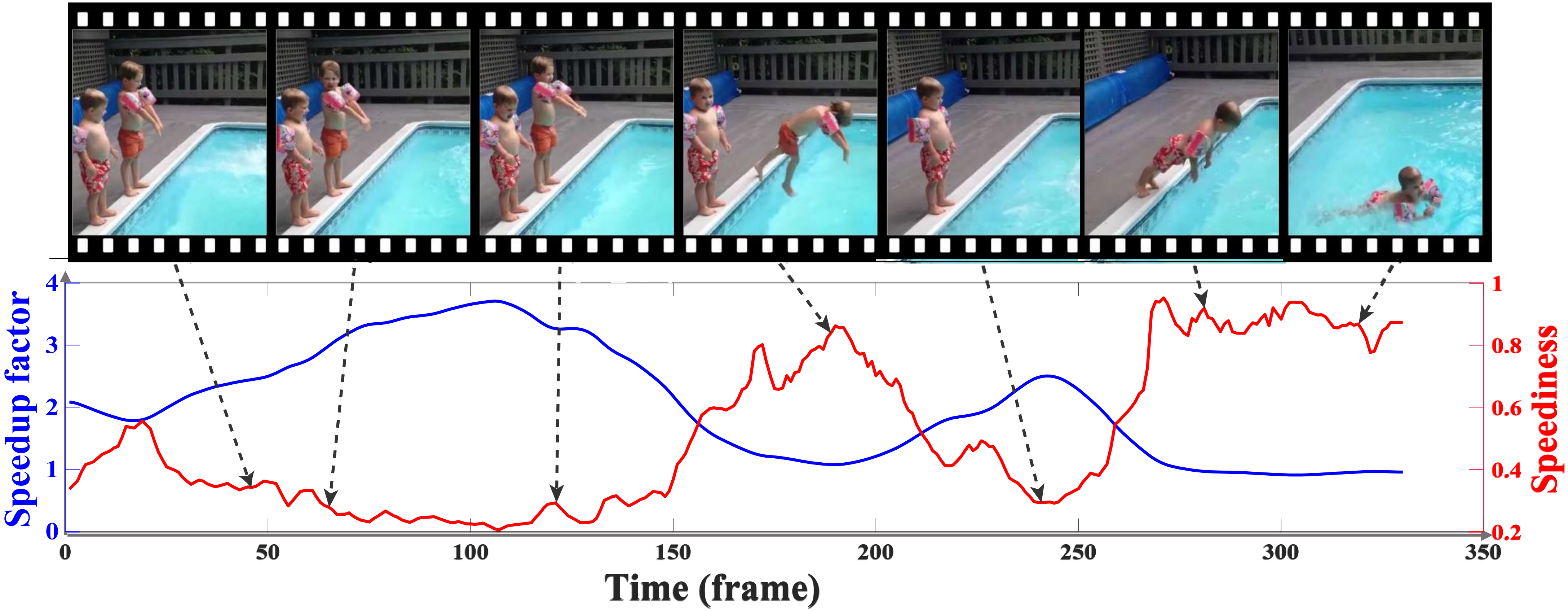


Self-Supervision in Videos: Learning the Speediness in Videos

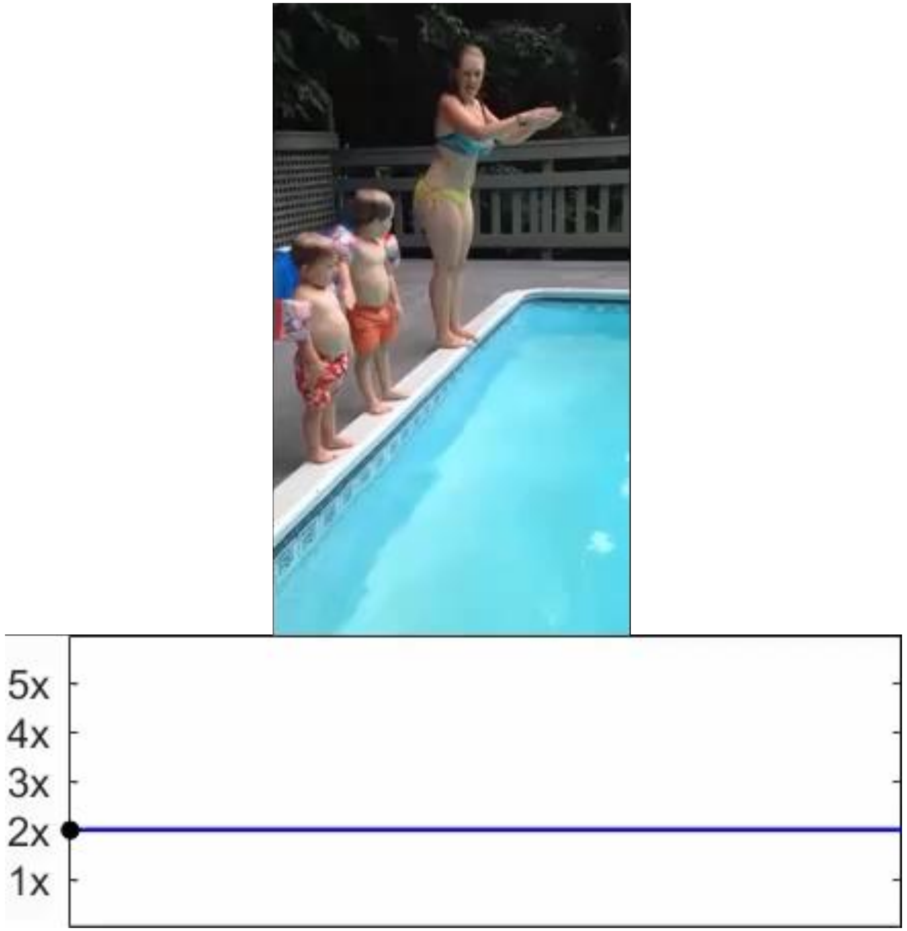
From "Speediness" to Speedup factor:

Low speediness → speedup more

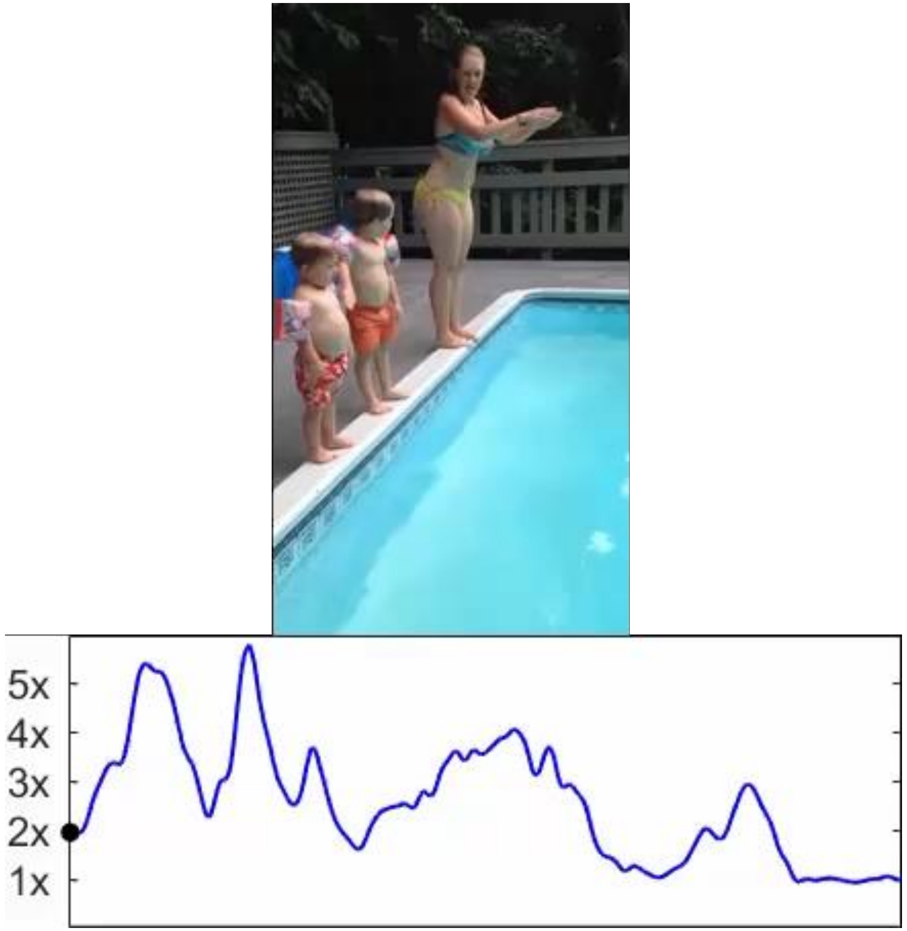
High speediness→ speedup less



Learning the Speediness in Videos: Adaptive Video Speedup



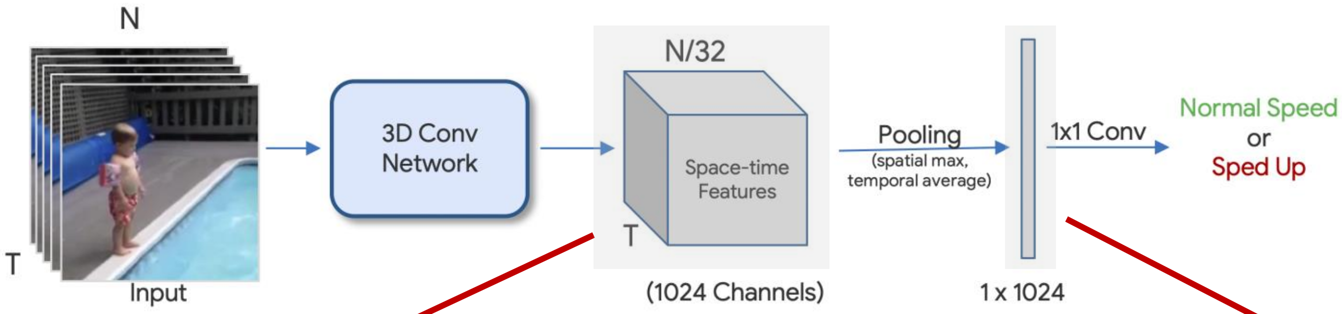
Uniform Speedup 2x



Adaptive Speedup 2x (ours)

Learning the Speediness in Videos: Transfer Learning

Pre-trained SpeedNet



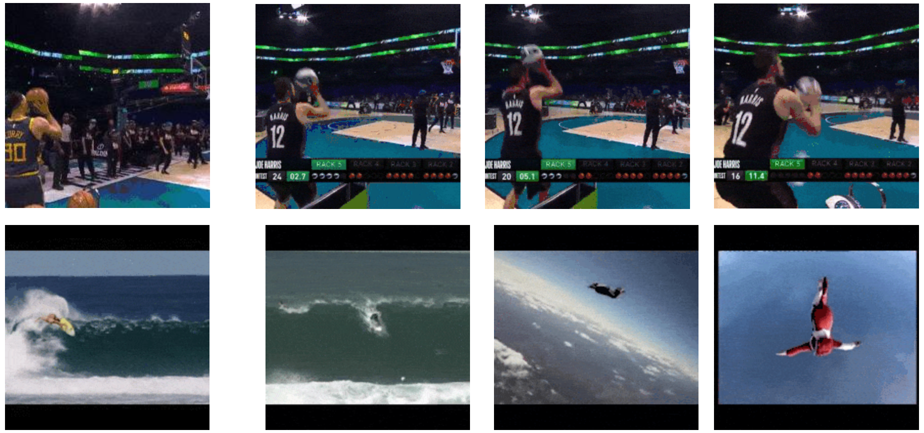
Self Supervised Action Recognition

Initialization		Supervised accuracy	
Method	**Architecture**	**UCF101**	**HMDB51**
Random init	S3D-G	73.8	46.4
ImageNet inflated	S3D-G	86.6	57.7
Kinetics supervised	S3D-G	96.8	74.5
CubicPuzzle [19]	3D-ResNet18	65.8	33.7
Order [40]	R(2+1)D	72.4	30.9
DPC [13]	3D-ResNet34	75.7	35.7
AoT [38]	T-CAM	79.4	-
SpeedNet (Ours)	S3D-G	**81.1**	**48.8**

Video Retrieval

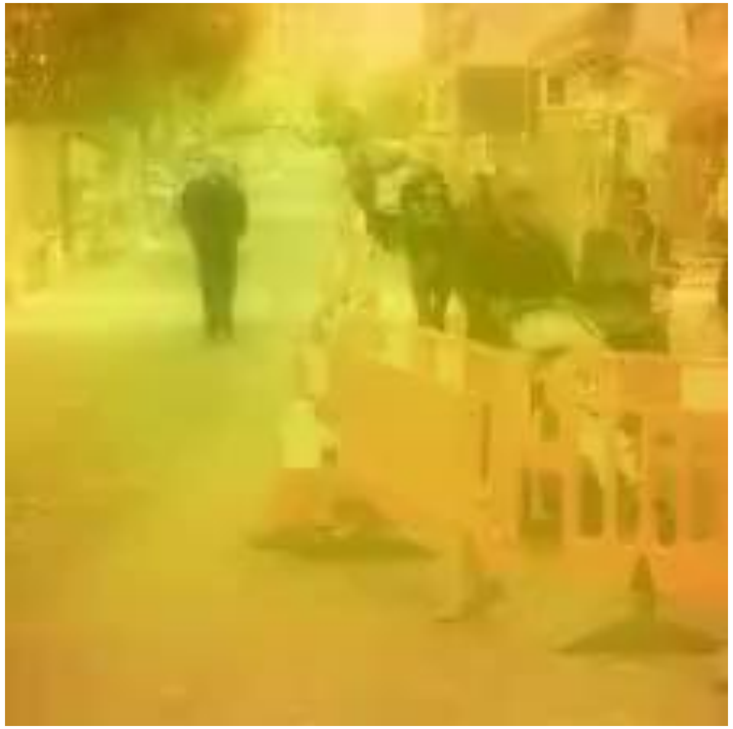
Query

Retrieved top-3 results



Learning the Speediness in Videos: CAM visualizations

"Memory Eleven"
artistic video by Bill Newsinge



Our space-time
speediness visualization

blue/green =
normal speed

yellow/orange =
slowed down

https://www.youtube.com/watch?v=djylS0Wi_Io

Next tutorial:

“GPU Fundamentals”

Next class:

”Neural Rendering”