Lecture 10:
Video Understanding

Recall: (2D) Image classification

(assume given a set of possible labels)
{dog, cat, truck, plane, ...}

cat

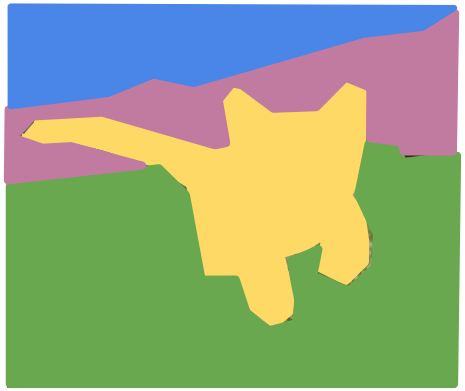
Last Lecture: (2D) Detection and Segmentation

Classification

CAT

No spatial extent

Semantic Segmentation



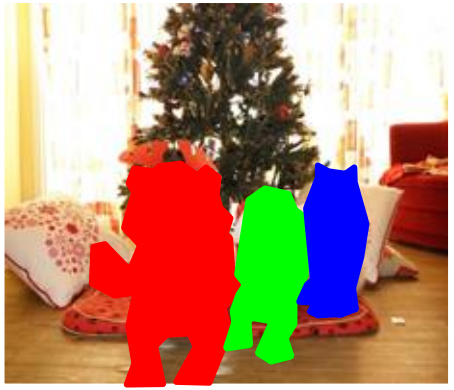
GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection

DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Multiple Objects


Living room
Dog
Baby

Today: Video = 2D + Time

A video is a sequence of images
4D tensor: T x 3 x H x W

Example task: Video Classification

Input video:
T x 3 x H x W

Swimming
Running
Jumping
Eating
Standing

Example task: Video Classification

Images: Recognize objects

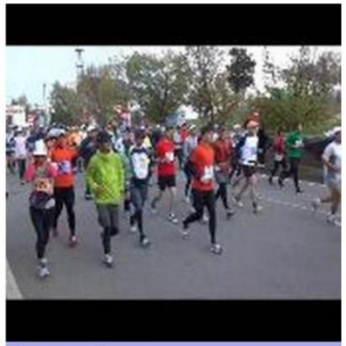
Dog
Cat
Fish
Truck

Videos: Recognize actions

Swimming
Running
Jumping
Eating
Standing

Example Video Dataset: Sports-1M

track cycling
cycling
track cycling
road bicycle racing
marathon
ultramarathon



ultramarathon
ultramarathon
half marathon
running
marathon
inline speed skating

heptathlon
heptathlon
decathlon
hurdles
pentathlon
sprint (running)

bikejoring
mushing
bikejoring
harness racing
skijoring
carting

longboarding
longboarding
aggressive inline skating
freestyle scootering
freeboard (skateboard)
sandboarding

1 million YouTube videos annotated with 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

Problem: Videos are big!

Videos are often ~30 frames per second (fps)

Input video:
T x 3 x H x W

Size of uncompressed video
(3 bytes per pixel):

SD (640 x 480): ~1.5 GB per minute
HD (1920 x 1080): ~10 GB per minute

Problem: Videos are big!

Input video:
T x 3 x H x W

Videos are often ~30 frames per second (fps)

Size of uncompressed video
(3 bytes per pixel):

SD (640 x 480): ~1.5 GB per minute
HD (1920 x 1080): ~10 GB per minute

One Solution: Train on short clips:
low fps and low spatial resolution
e.g. T = 16, H=W=112
(3.2 seconds at 5 fps, 588 KB)

Training vs Testing on Short Video Clips

Raw video: Long, high FPS

Training vs Testing on Video Clips

Raw video: Long, high FPS

Training: Train model to classify short clips with low FPS

Training vs Testing on Video Clips

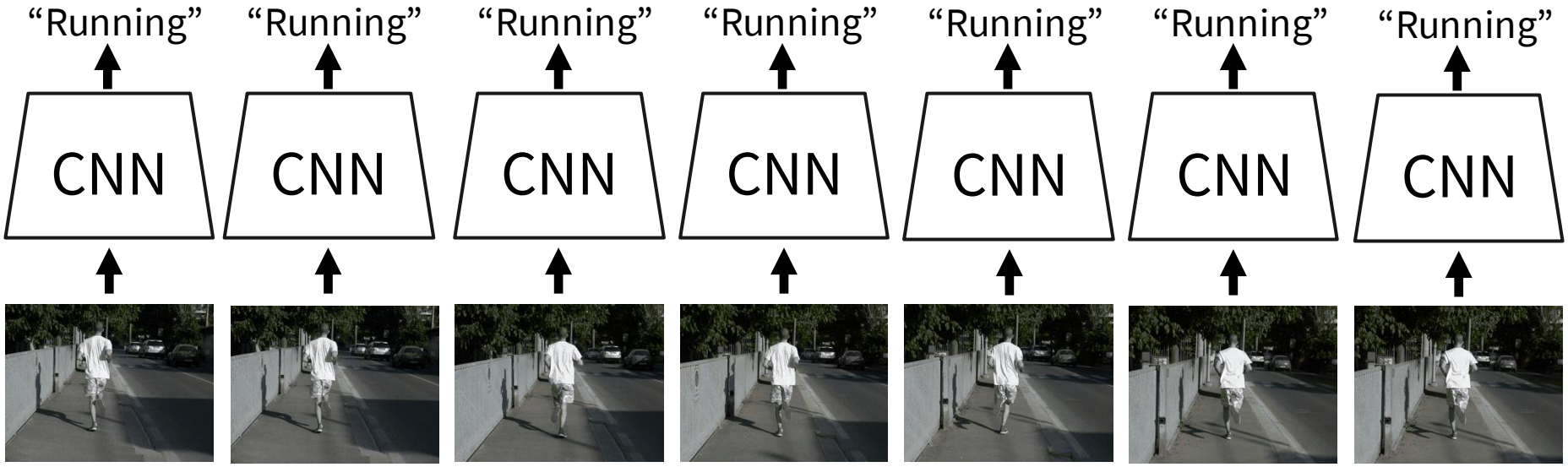
Raw video: Long, high FPS

Training: Train model to classify short clips with low FPS

Testing: Run model on different clips, average predictions

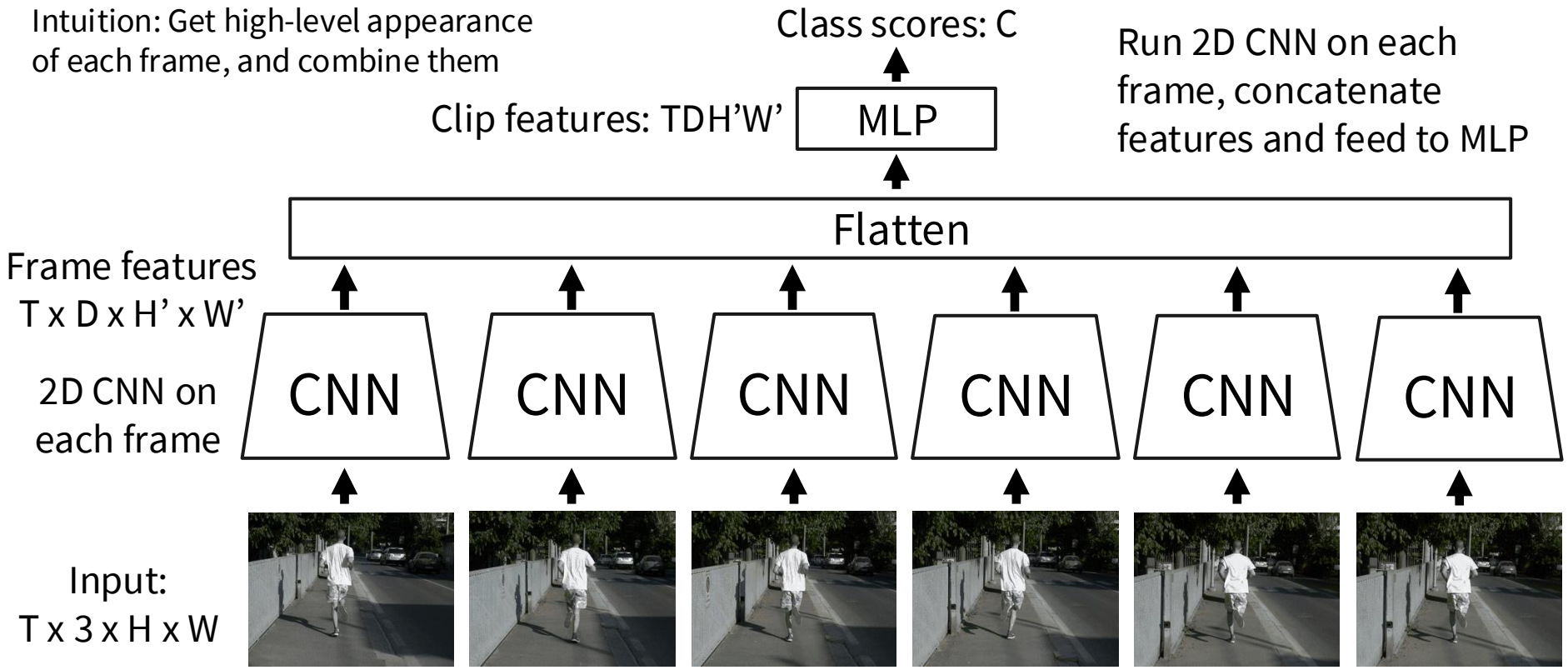
Video Classification: Single-Frame CNN

Simple idea: train normal 2D CNN to classify video frames independently!
(Average predicted probs at test-time)
Often a very strong baseline for video classification



Video Classification: Late Fusion (with FC layers)

Intuition: Get high-level appearance of each frame, and combine them



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

Video Classification: Late Fusion (with FC layers)

Intuition: Get high-level appearance of each frame, and combine them

Class scores: C

Clip features: TDH'W'

MLP

Run 2D CNN on each frame, concatenate features and feed to MLP

Q: How to handle arbitrary length?

Flatten

Frame features
T x D x H' x W'

2D CNN on each frame

CNN CNN CNN CNN CNN CNN

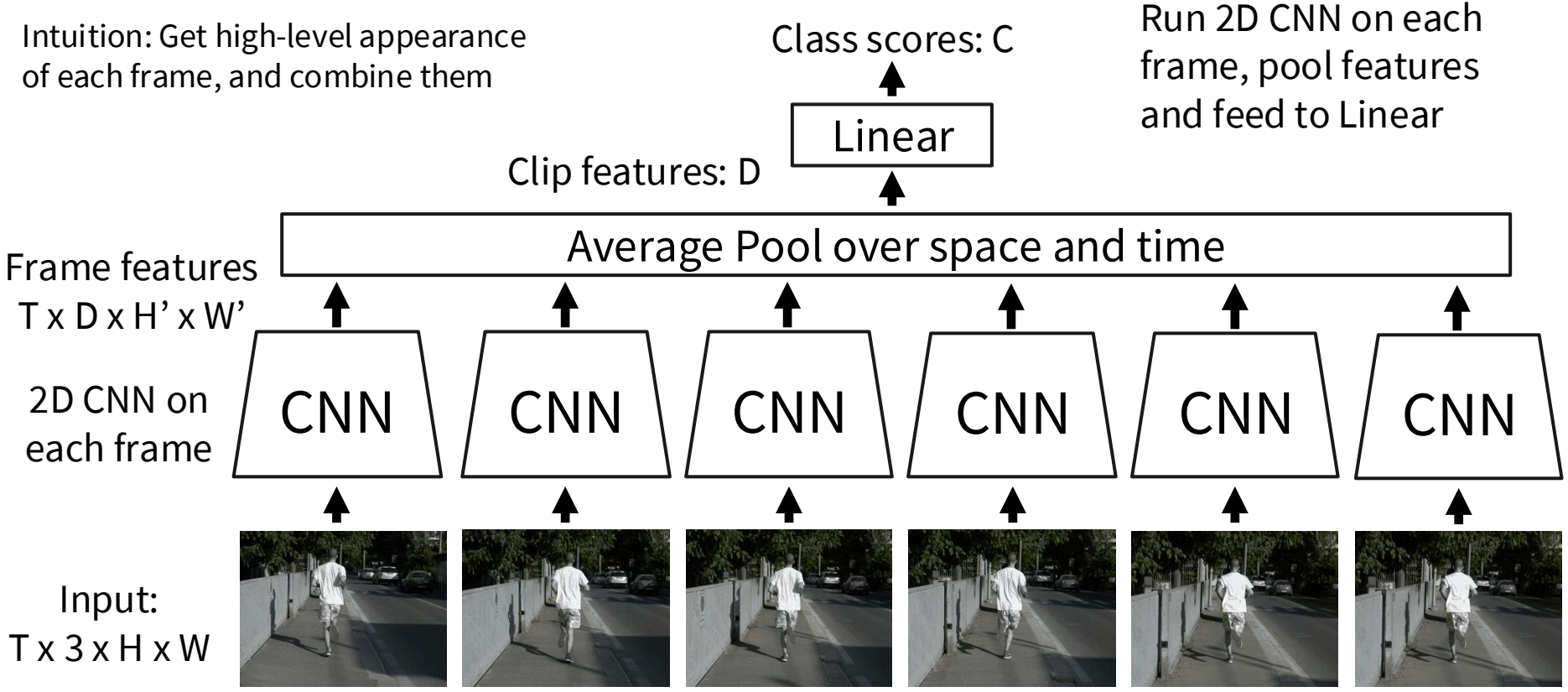
Input:
T x 3 x H x W

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

Video Classification: Late Fusion (with pooling)

Intuition: Get high-level appearance of each frame, and combine them

Run 2D CNN on each frame, pool features and feed to Linear



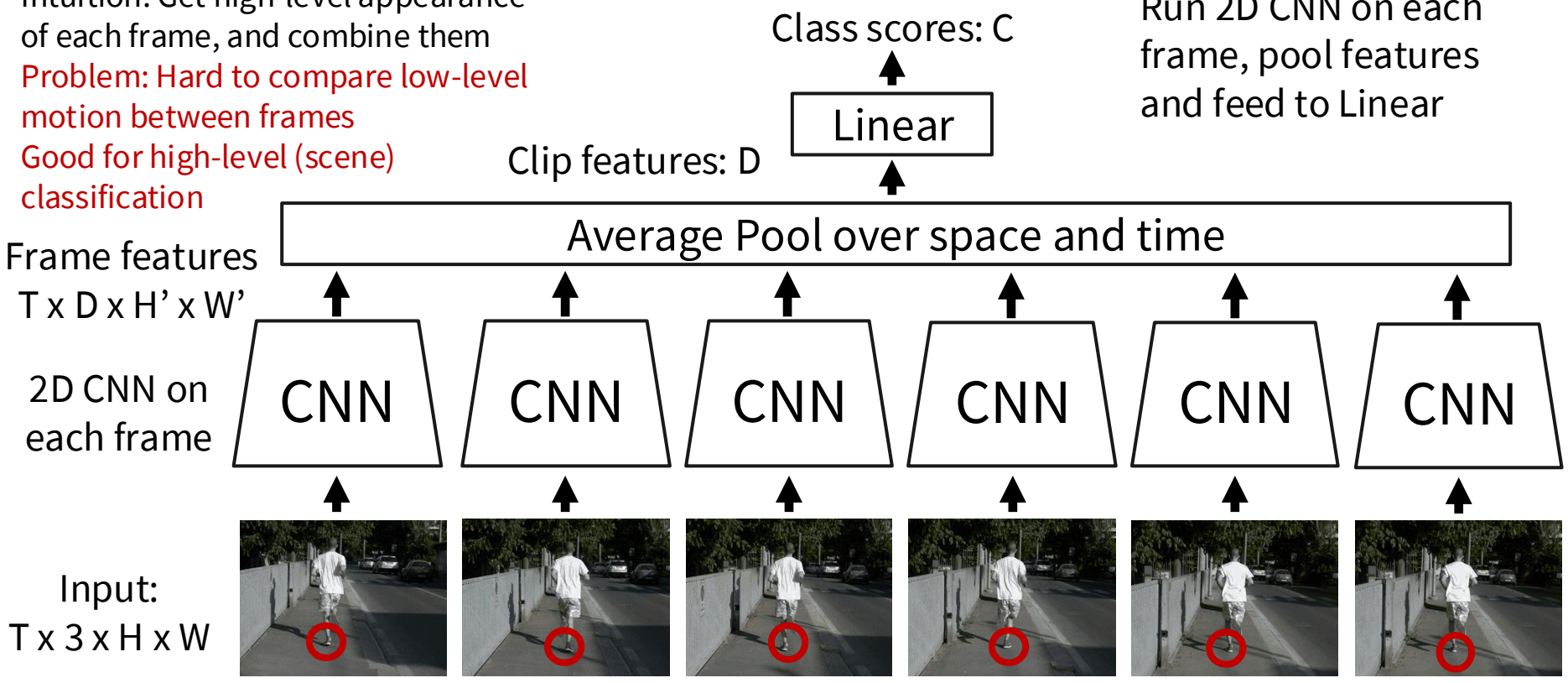
Video Classification: Late Fusion (with pooling)

Intuition: Get high-level appearance of each frame, and combine them

Problem: Hard to compare low-level motion between frames

Good for high-level (scene) classification

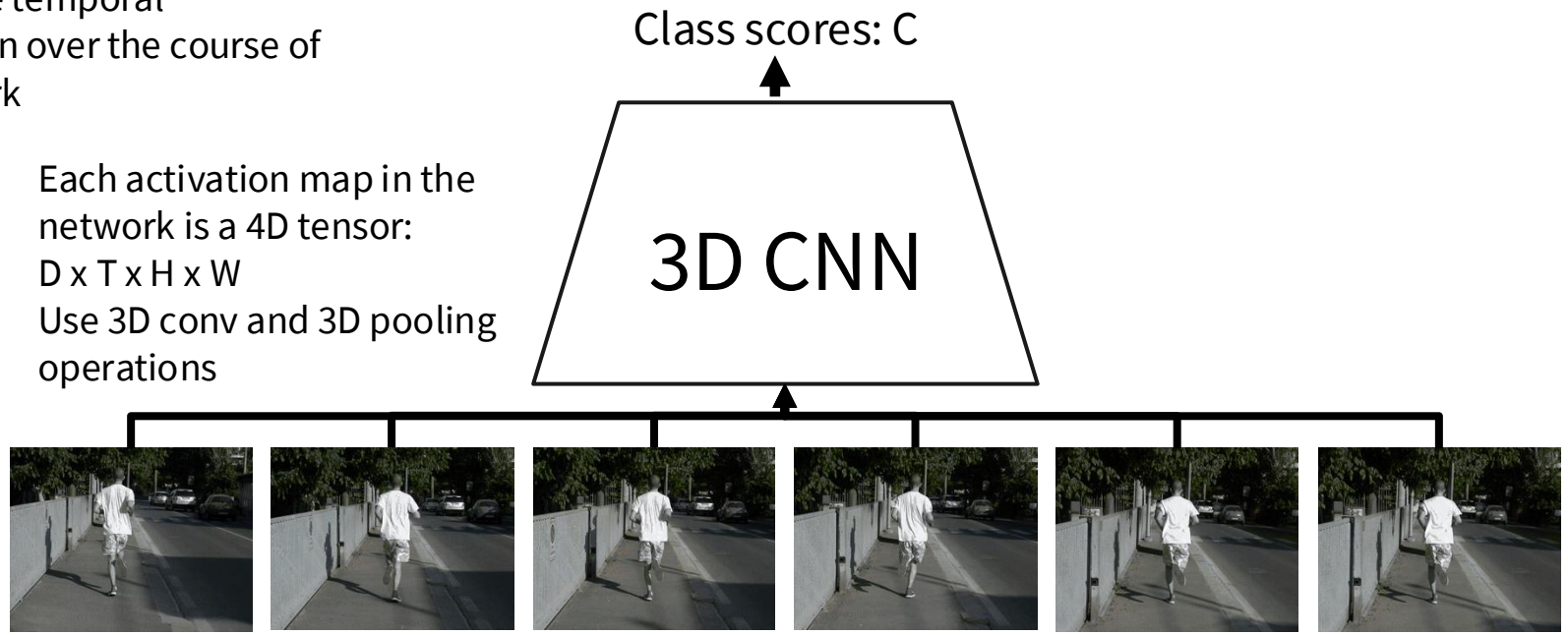
Run 2D CNN on each frame, pool features and feed to Linear



Video Classification: Early Fusion with 3D CNNs

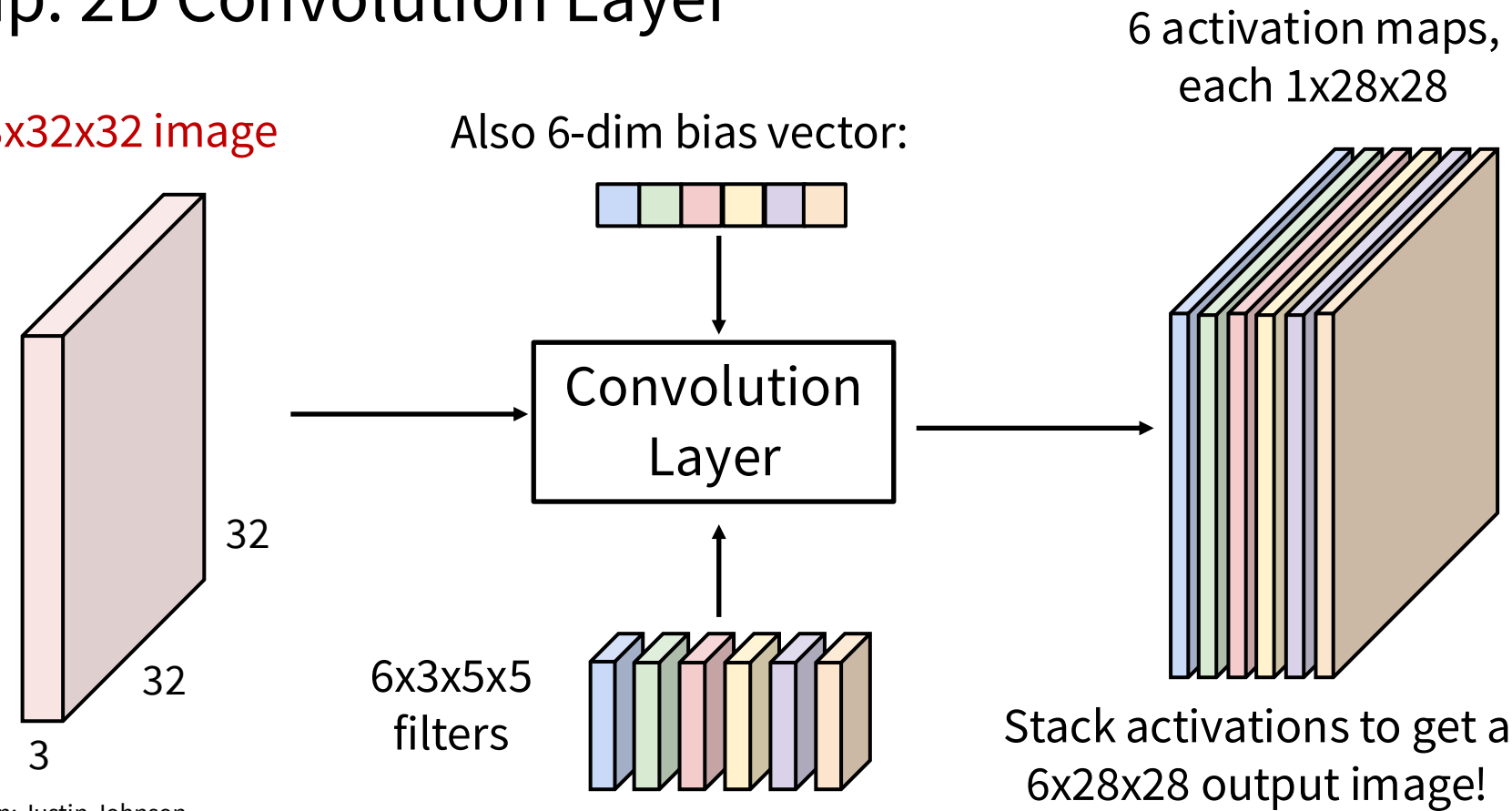
Intuition: Use 3D versions of convolution and pooling to slowly fuse temporal information over the course of the network

Each activation map in the network is a 4D tensor:
D x T x H x W
Use 3D conv and 3D pooling operations



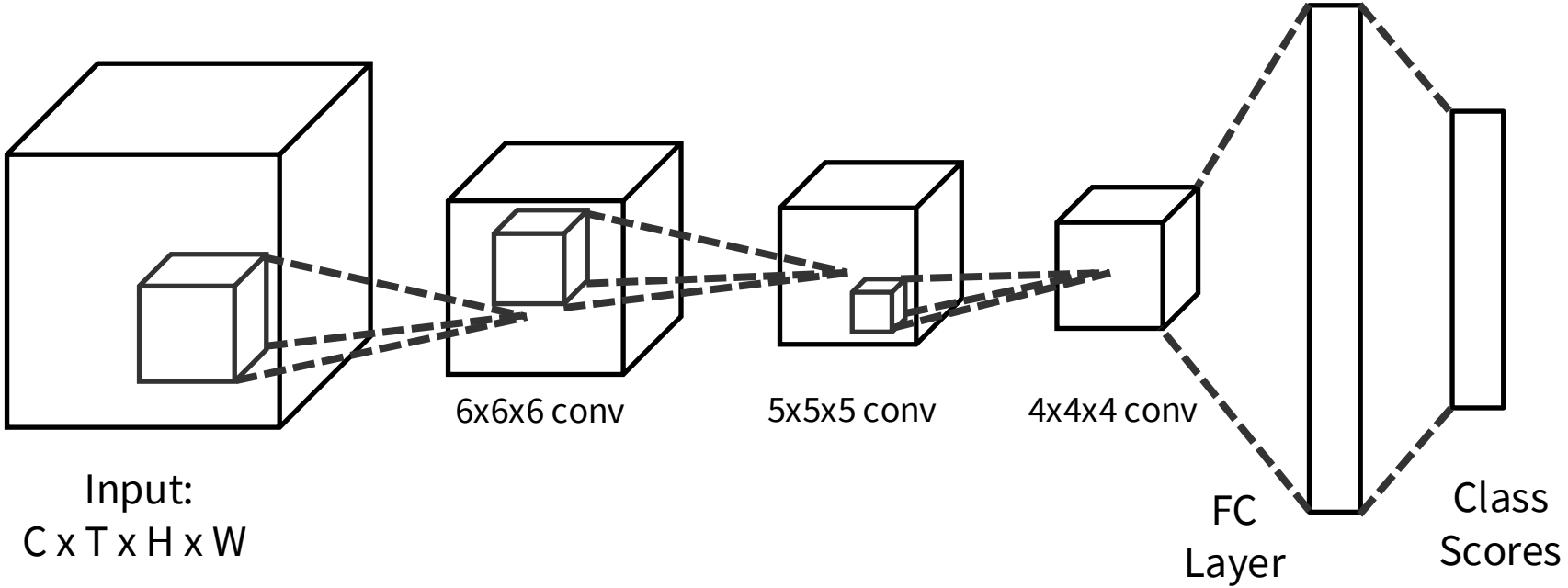
Ji et al, "3D Convolutional Neural Networks for Human Action Recognition", TPAMI 2010; Karpathy et al, "Large-scale Video Classification with Convolutional Neural Networks", CVPR 2014

Recap: 2D Convolution Layer



Slide inspiration: Justin Johnson

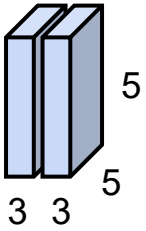
3D Convolution: High Level Idea



How is 3D Different? Filters

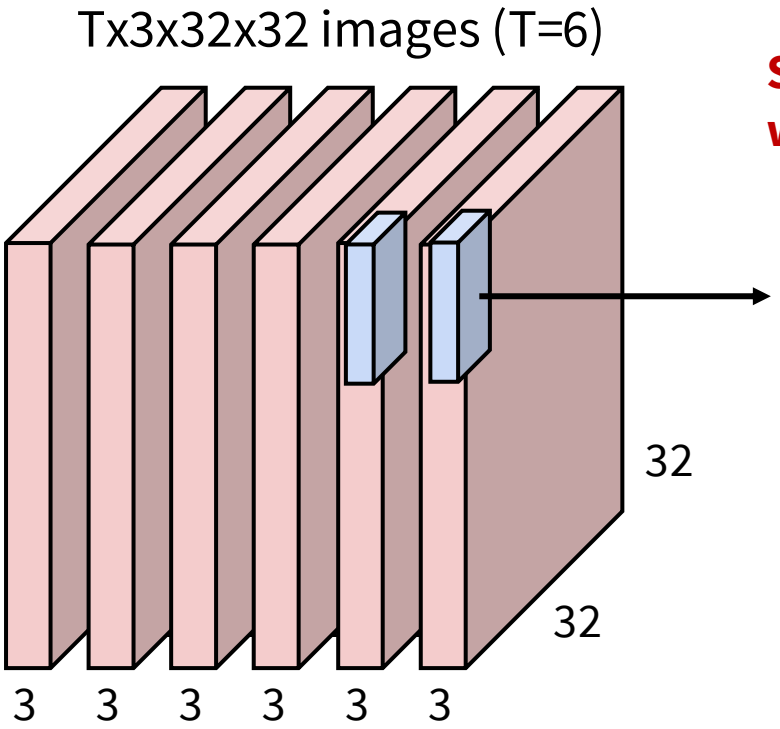
What does one filter look like?

One 2x3x5x5 filter



Now has time dimension!

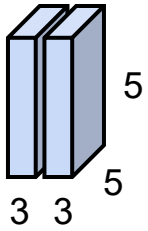
How is 3D Different? Filters



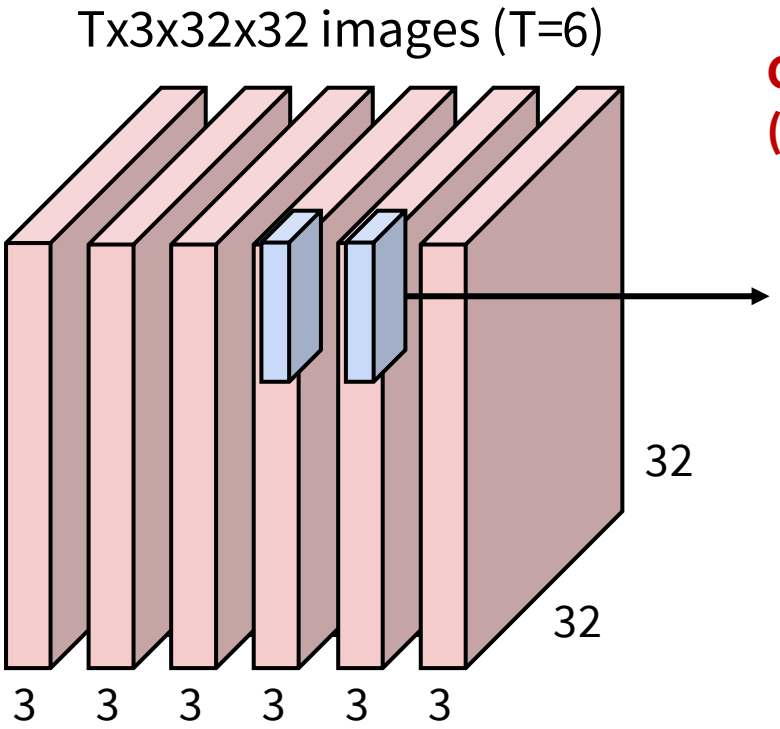
Sliding window mechanism is the same, with an extra dimension

Dot product to produce value at each location (+ bias)

One 2x3x5x5 filter



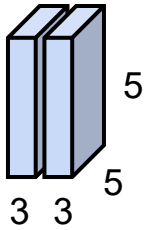
How is 3D Different? Stride



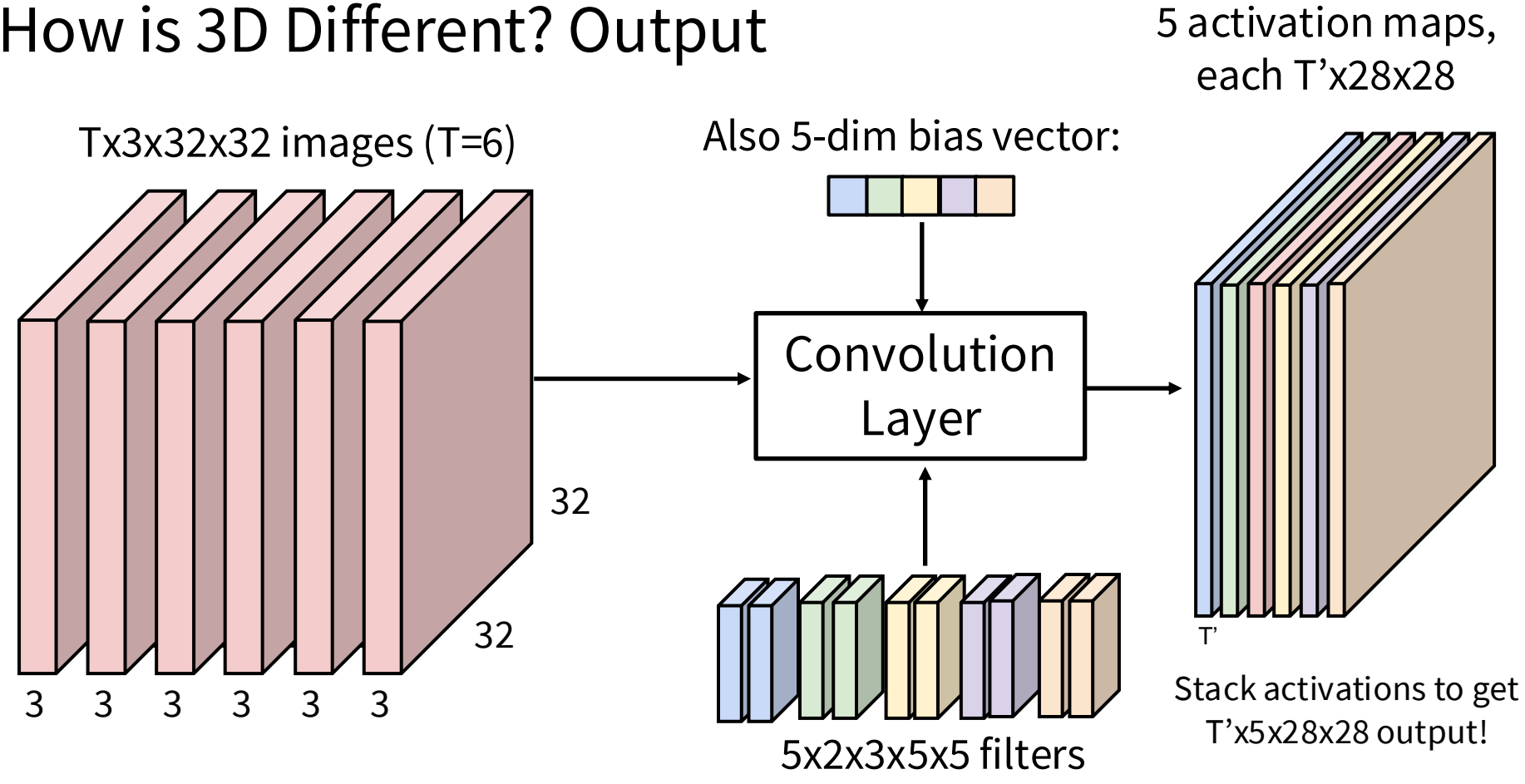
Can move forward/backward in time (stride rules apply)

Dot product to produce value at each location (+ bias)

One 2x3x5x5 filter



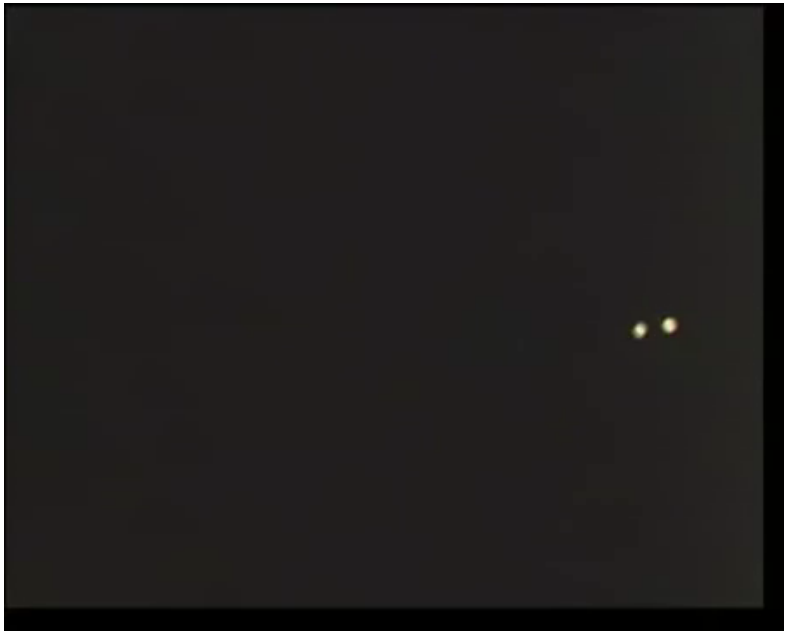
How is 3D Different? Output



Next: Some Tricks To Improve Performance

Trick #1: Recognizing Actions from Motion

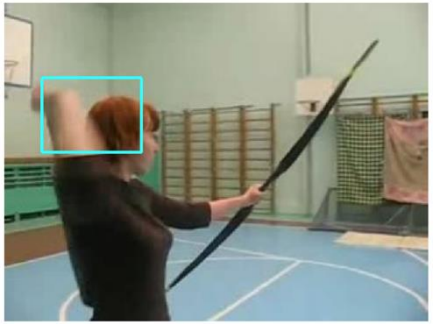
Insight: We can easily recognize actions using only motion information



Johansson, "Visual perception of biological motion and a model for its analysis." Perception & Psychophysics. 14(2):201-211. 1973.

Measuring Motion: Optical Flow

Image at frame t



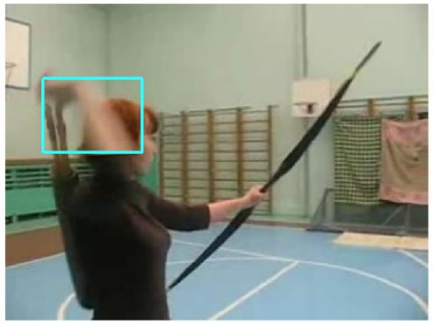
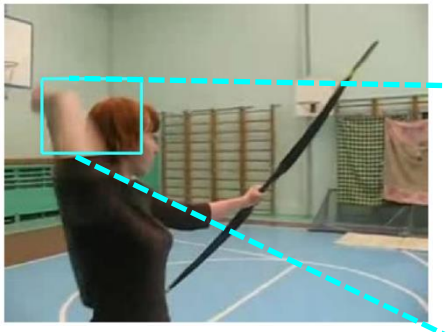


Image at frame t+1

Simonyan and Zisserman, “Two-stream convolutional networks for action recognition in videos”, NeurIPS 2014

Measuring Motion: Optical Flow

Image at frame t



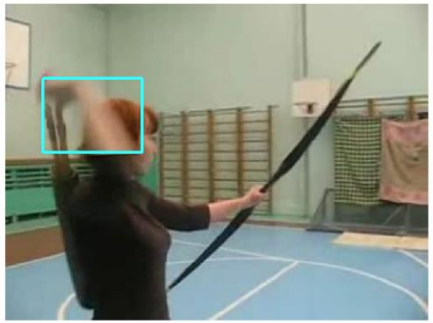
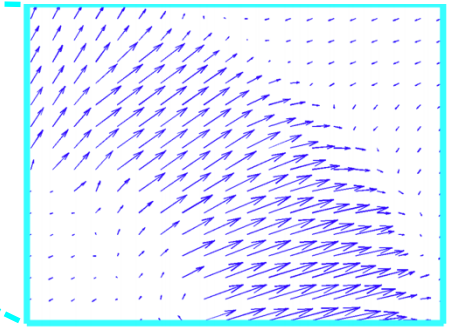


Image at frame t+1

Optical flow gives a displacement field F between images I_t and I_{t+1}



Tells where each pixel will move in the next frame:

$F(x, y) = (dx, dy)$

$I_{t+1}(x+dx, y+dy) = I_t(x, y)$

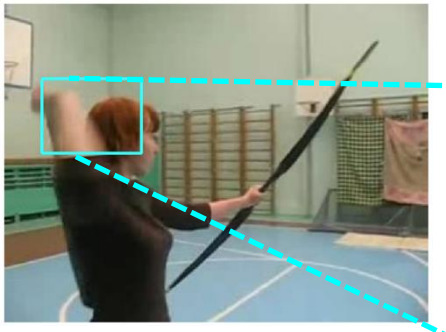
Computed via algorithm or faster NN approximation

Simonyan and Zisserman, "Two-stream convolutional networks for action recognition in videos", NeurIPS 2014

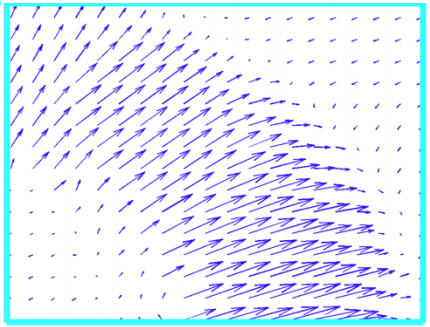
Measuring Motion: Optical Flow

Optical Flow highlights local motion

Image at frame t



Optical flow gives a displacement field F between images I_t and I_{t+1}



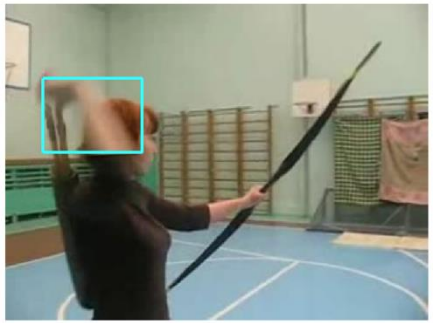


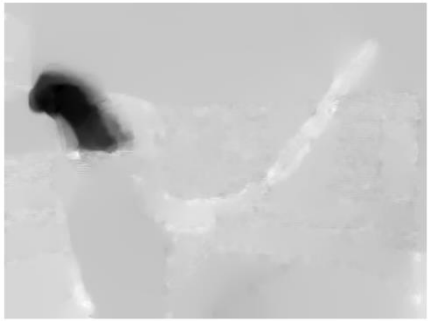
Image at frame t+1

Tells where each pixel will move in the next frame:

$F(x, y) = (dx, dy)$

$I_{t+1}(x+dx, y+dy) = I_t(x, y)$

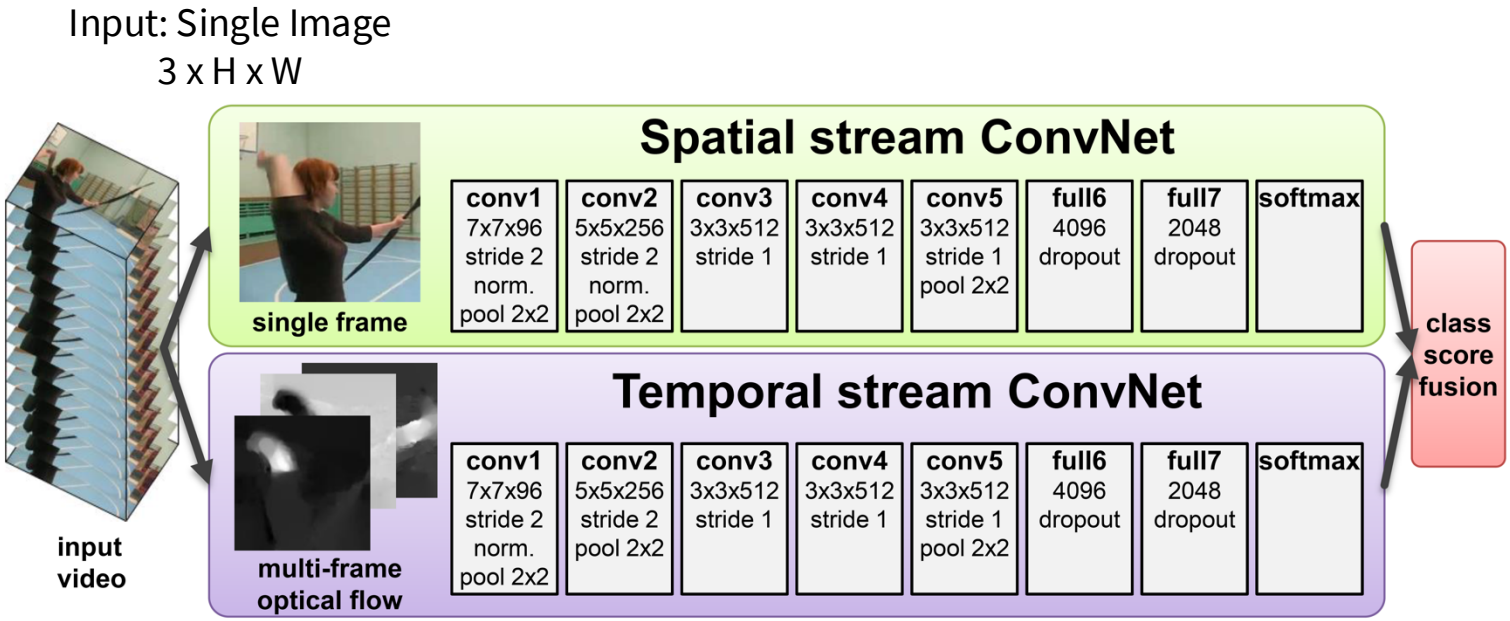
Horizontal flow dx



Vertical Flow dy

Simonyan and Zisserman, "Two-stream convolutional networks for action recognition in videos", NeurIPS 2014

Separating Motion and Appearance: Two-Stream Networks

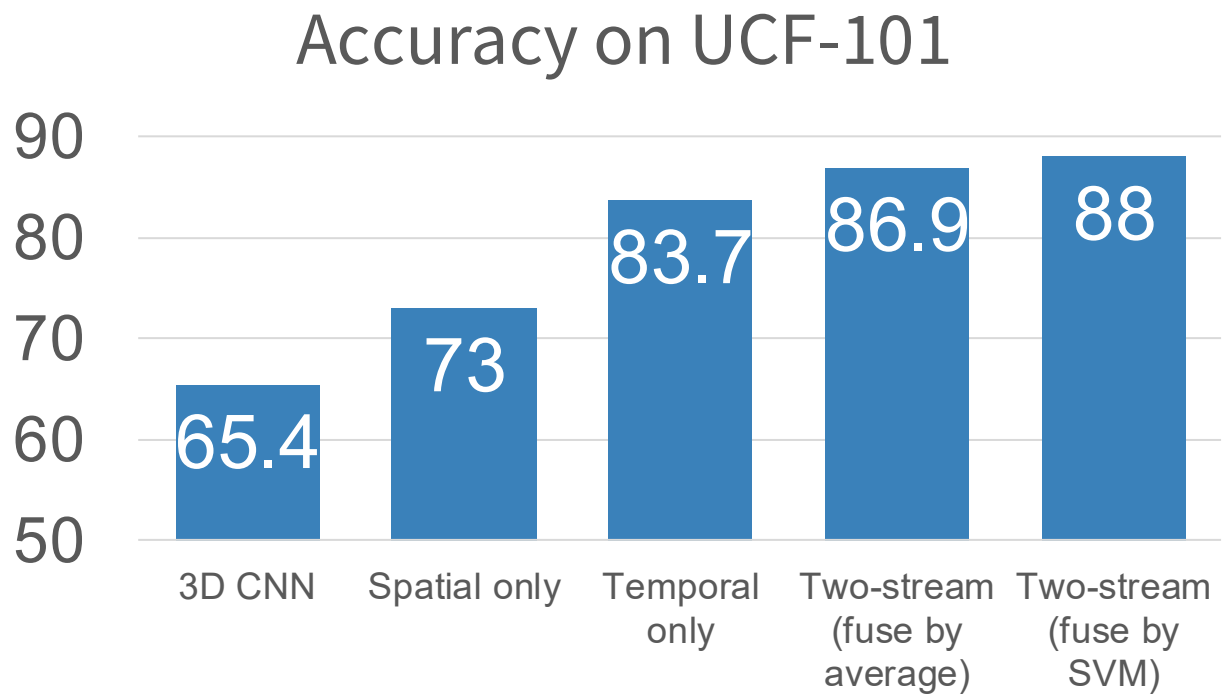


Input: Stack of optical flow:
[2*(T-1)] x H x W

Early fusion: First 2D conv processes all flow images

Simonyan and Zisserman, "Two-stream convolutional networks for action recognition in videos", NeurIPS 2014

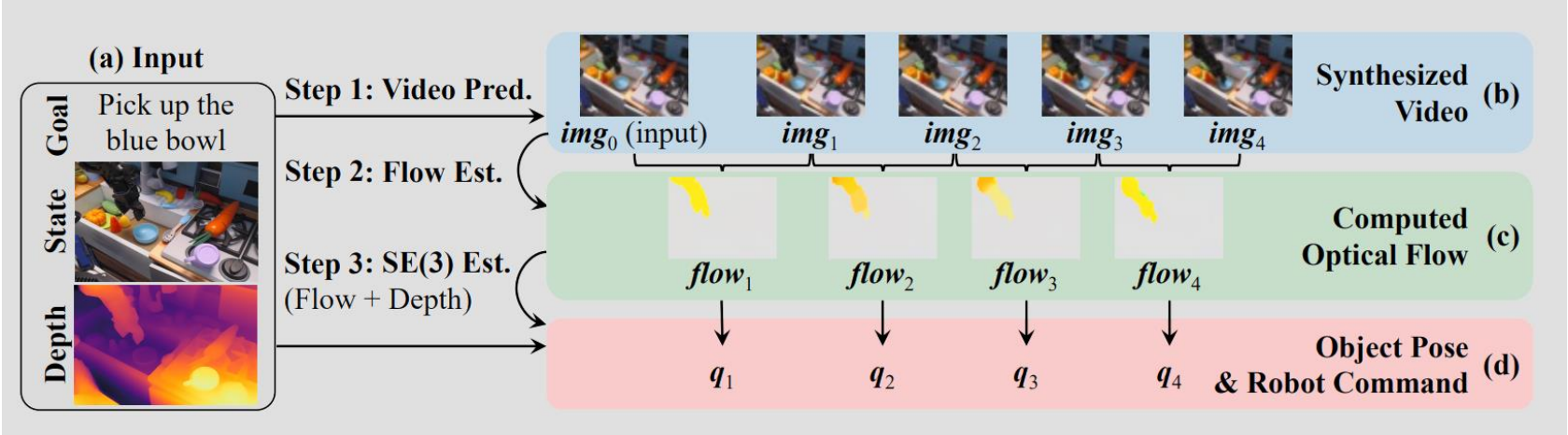
Separating Motion and Appearance: Two-Stream Networks



Simonyan and Zisserman, "Two-stream convolutional networks for action recognition in videos", NeurIPS 2014

Recently Optical Flow Rarely Used for Video Understanding

Used mainly as an intermediate representation for other tasks
(for example, robotic control)



Ko, Po-Chen, et al. "Learning to Act from Actionless Videos through Dense Correspondences." *ICLR 2024*.

Trick #2: Inflating 2D Networks to 3D (I3D)

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

Idea: take a 2D CNN architecture.

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

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

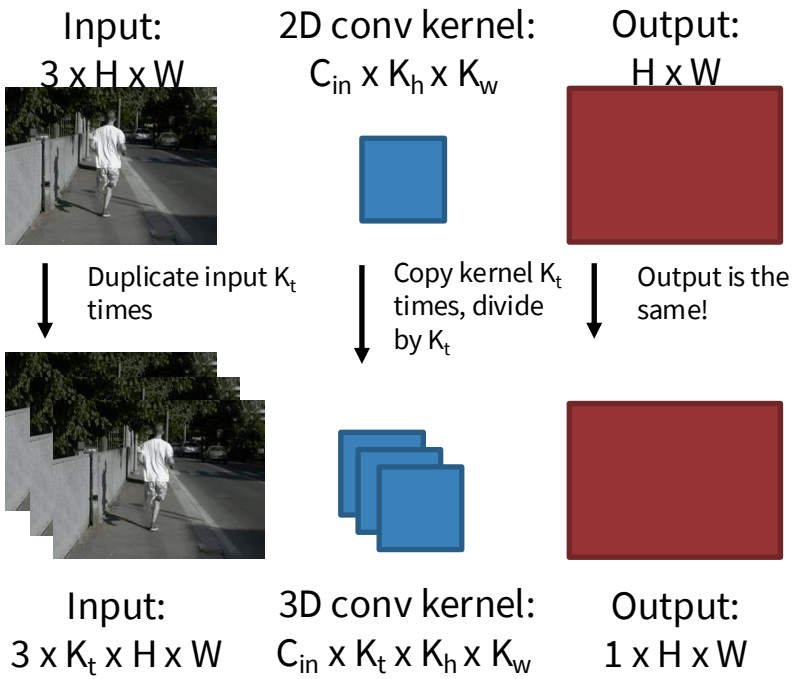
Inflating 2D Networks to 3D (I3D)

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

Idea: take a 2D CNN architecture.

Replace each 2D $K_h \times K_w$ conv/pool layer with a 3D $K_t \times K_h \times K_w$ version

Can use weights of 2D conv to initialize 3D conv: copy K_t times in space and divide by K_t
This gives the same result as 2D conv given "constant" video input



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

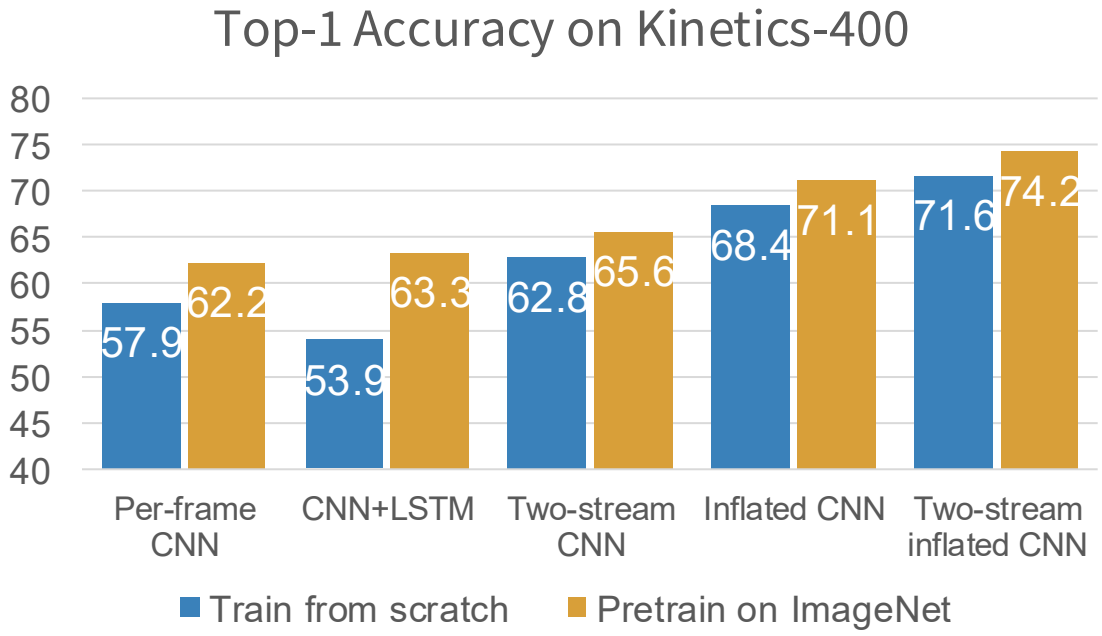
Inflating 2D Networks to 3D (I3D)

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

Idea: take a 2D CNN architecture.

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

Can use weights of 2D conv to initialize 3D conv: copy K_t times in space and divide by K_t
This gives the same result as 2D conv given "constant" video input



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

All using Inception CNN

Video Understanding: Evolution of the Field

2014 – 2021 Era of 3D CNNs + RNNs

2021 – 2026 Transformers

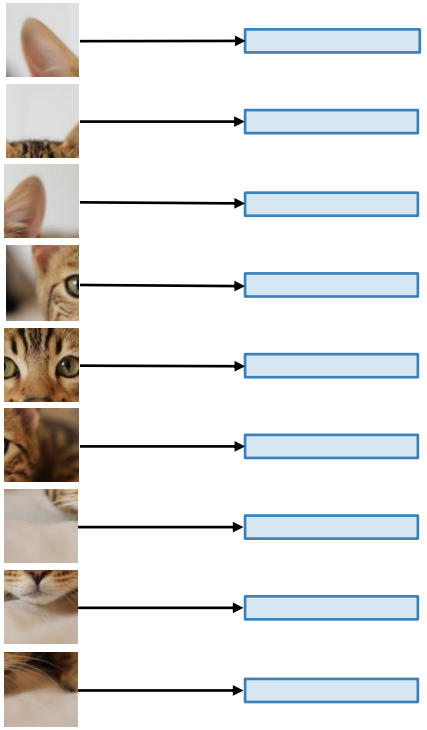
Video Understanding: Evolution of the Field

2014 – 2021 Era of 3D CNNs + RNNs

2021 – 2026 Transformers

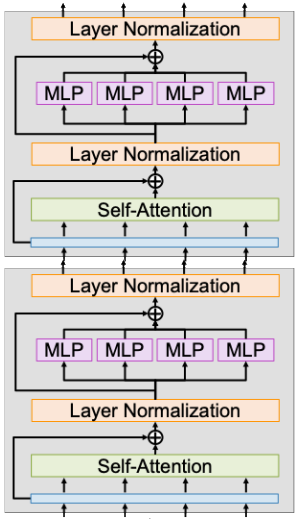
Image Level: Vision Transformers (ViT)

Input image:
e.g. 224x224x3



Break into patches
e.g. 16x16x3

Flatten and apply a linear transform 768 => D



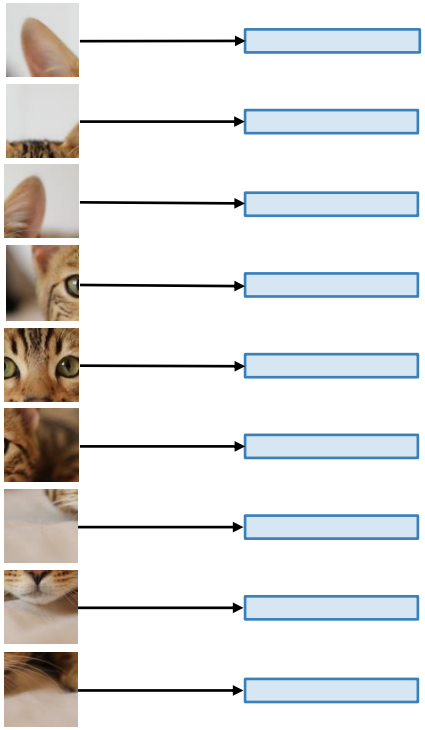
D-dim vector per patch are the input vectors to the Transformer

Use positional encoding to tell the transformer the 2D position of each patch

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

Video Level: Vision Transformers (ViT)

Input Frame:
e.g. 224x224x3



Break into patches
e.g. 16x16x3

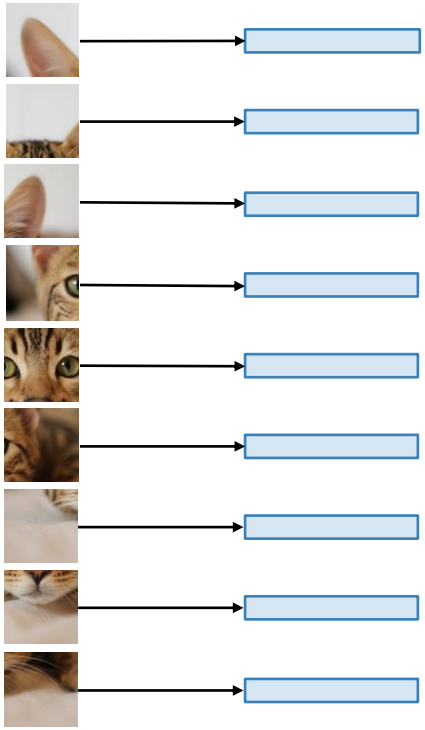
14 x 14 = **196 tokens per image**. This is a ton!

Small clip: T = 16 → 3136 tokens
~book chapter

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

Video Level: Vision Transformers (ViT)

Input Frame:
e.g. 224x224x3



Break into patches
e.g. 16x16x3

14 x 14 = **196 tokens per image**. This is a ton!

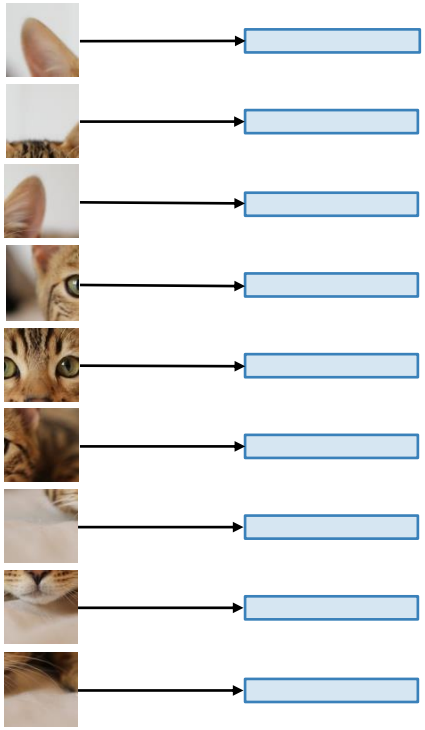
Small clip: T = 16 → 3136 tokens
~book chapter

5 minutes at 1fps: T = 300 → 58.8k tokens
~short novel

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

Video Level: Vision Transformers (ViT)

Input Frame:
e.g. 224x224x3



Break into patches
e.g. 16x16x3

14 x 14 = **196 tokens per image**. This is a ton!

Small clip: T = 16 → 3136 tokens
~book chapter

5 minutes at 1fps: T = 300 → 58.8k tokens
~short novel

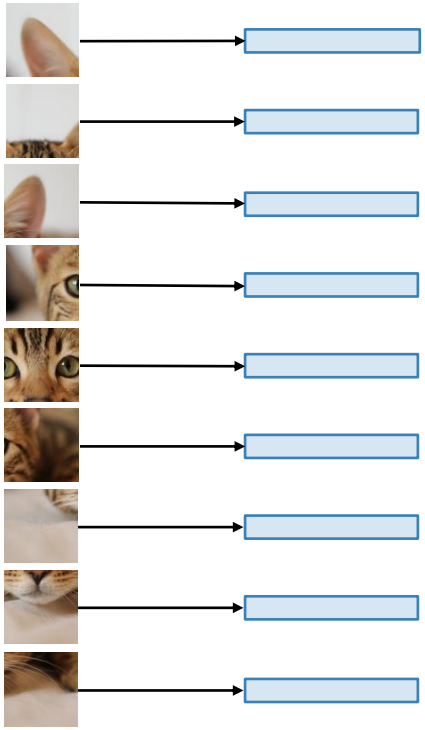
5 minutes at 24fps → 1,411,200 tokens
(~context limit of most modern LLMs)

Compare to human perception speed…

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

Video Level: Vision Transformers (ViT)

Input Frame:
e.g. 224x224x3



Break into patches
e.g. 16x16x3

14 x 14 = **196 tokens per image**. This is a ton!

Small clip: T = 16 → 3136 tokens
~book chapter

5 minutes at 1fps: T = 300 → 58.8k tokens
~short novel

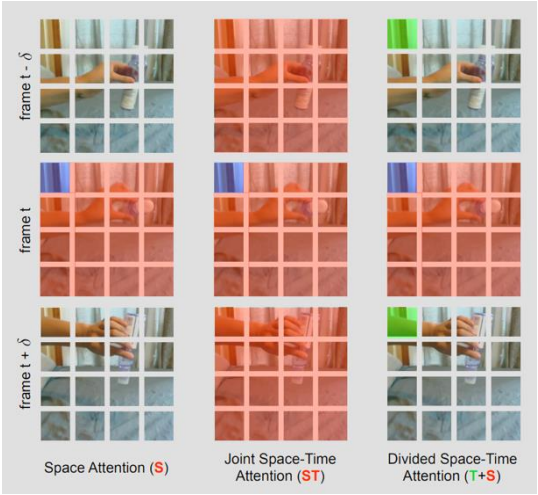
5 minutes at 24fps → 1,411,200 tokens
(~context limit of most modern LLMs)

Naïve patch-level attention scales very poorly! How to solve this?

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

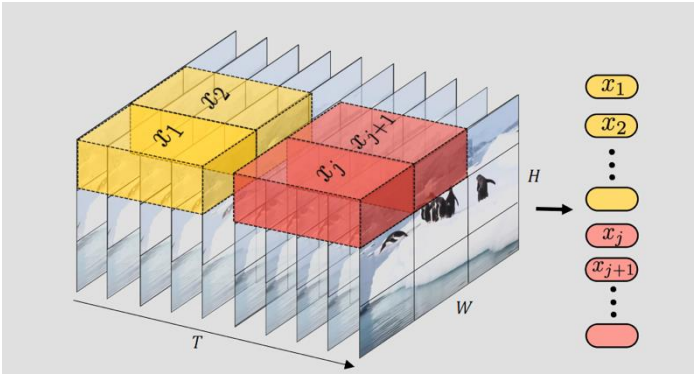
Two Broad Strategies for Improving Video Efficiency

Modify attention operator



Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

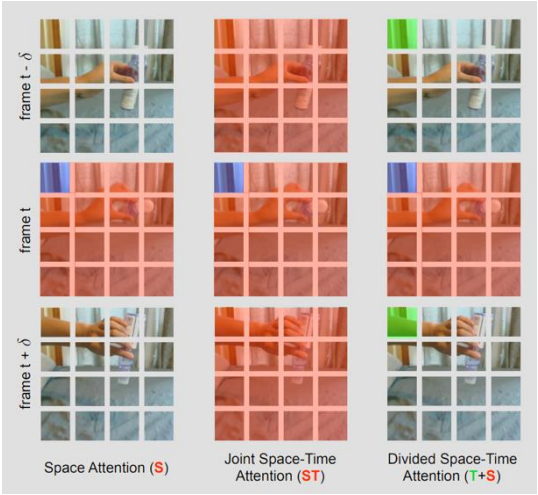
Reduce the number of tokens



Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

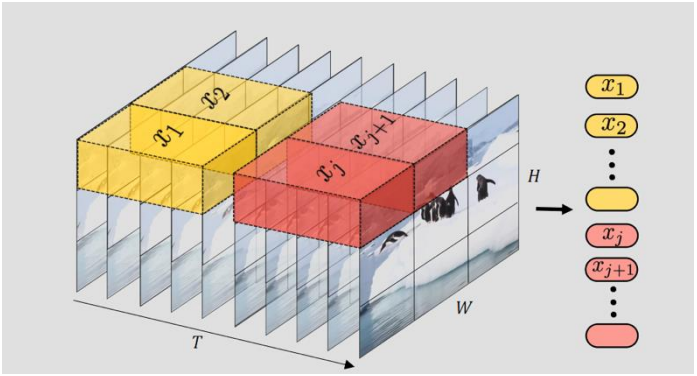
Two Broad Strategies for Improving Video Efficiency

Modify attention operator



Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

Reduce the number of tokens

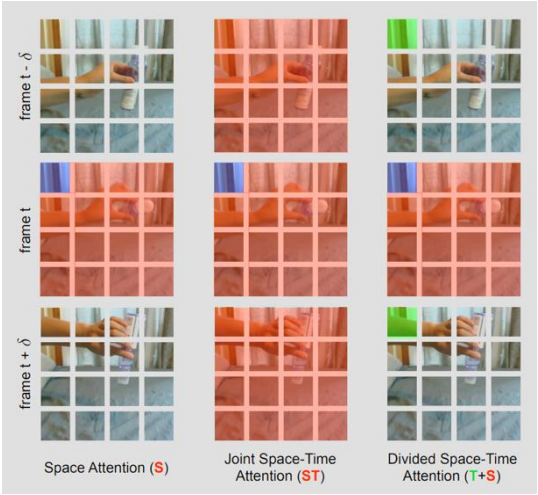


Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

Many different approaches for both -- we will only cover a couple in each category today!

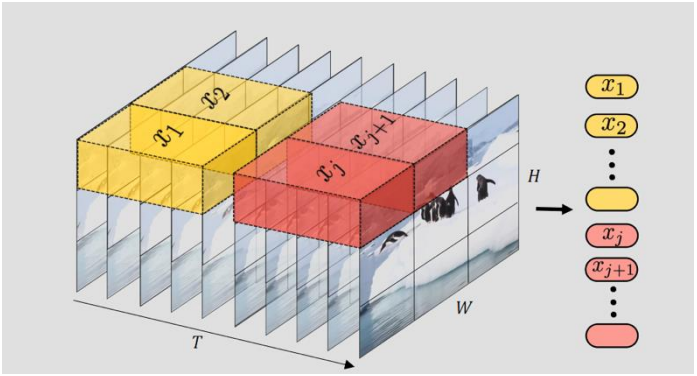
Two Broad Strategies for Improving Video Efficiency

Modify attention operator



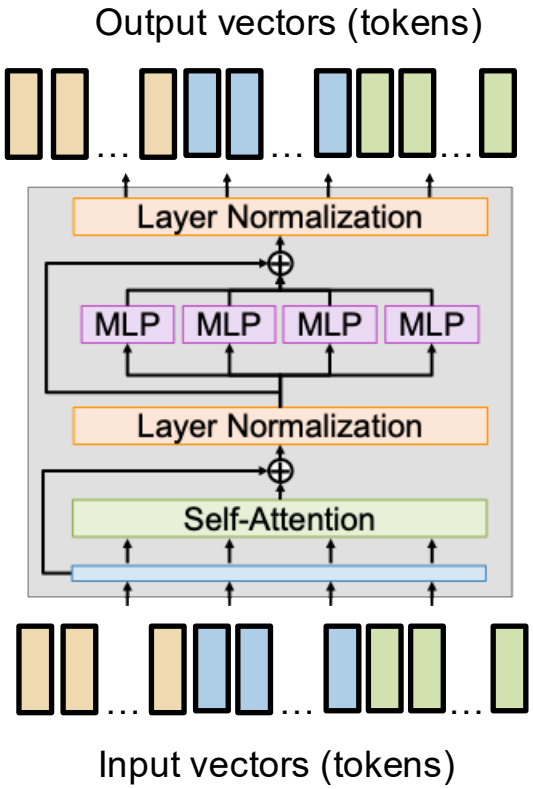
Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

Reduce the number of tokens



Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

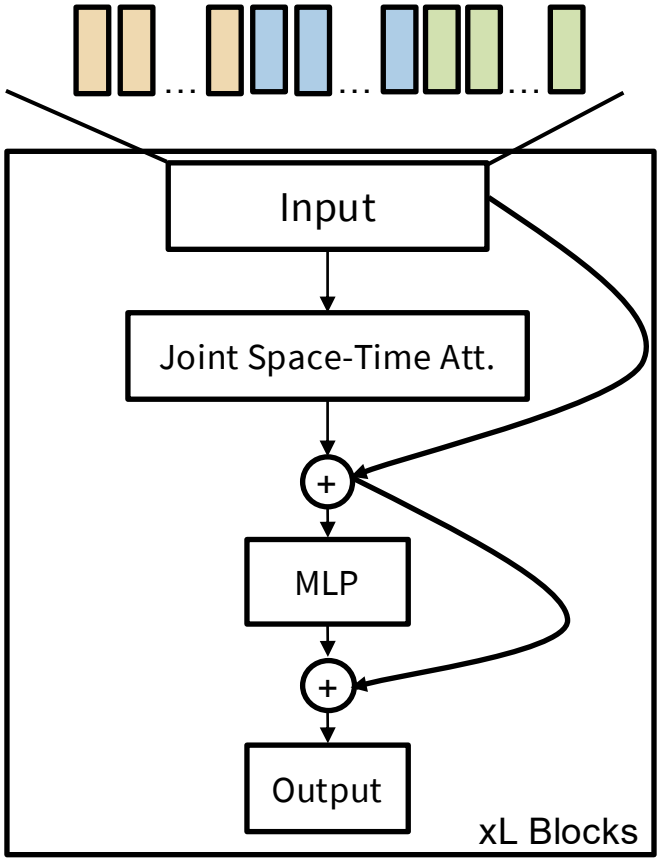
Recall: Transformer Block



Assume fixed N patches per frame as before (3 shown).

Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Joint Space-Time Attention



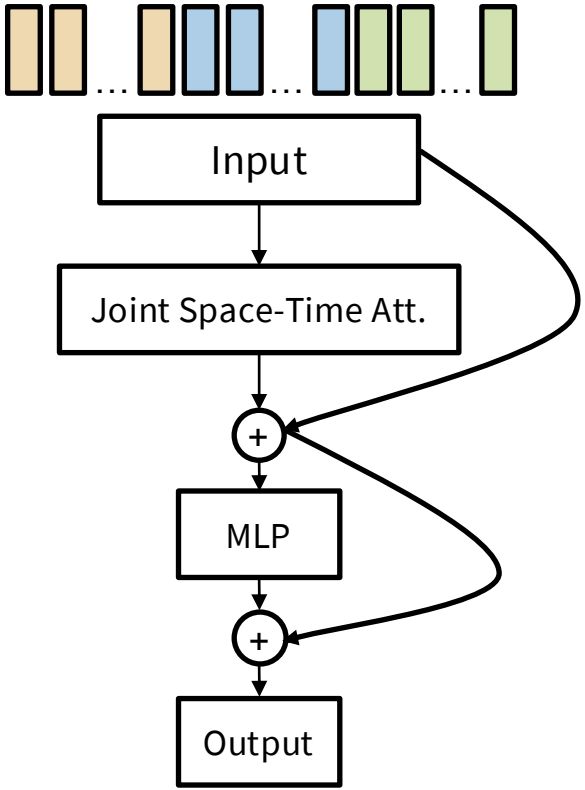
Extremely inefficient!

Recall: 5 minutes at 24fps → 1,411,200 tokens

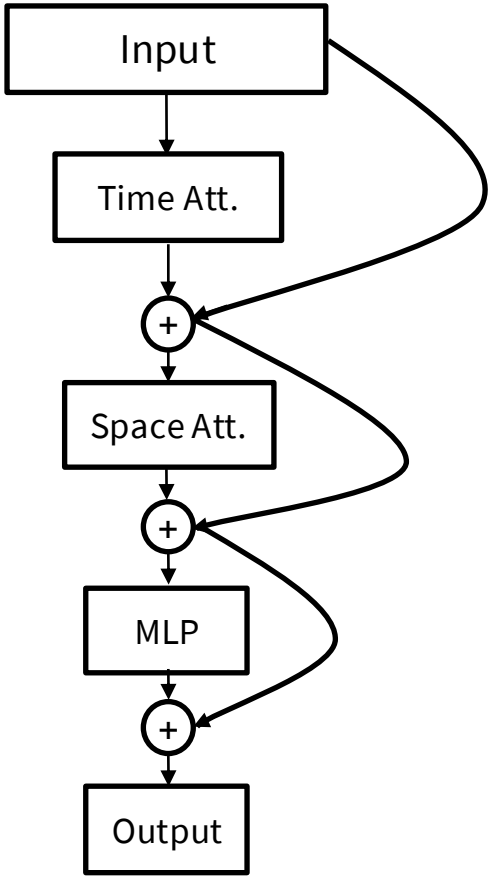
For simplicity, normalization layers not shown in diagram.

Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Joint vs Divided Space-Time Attention

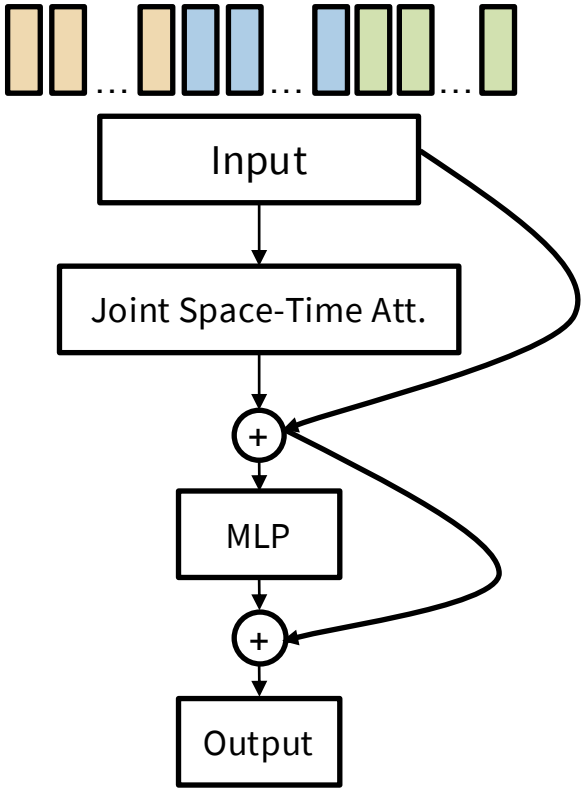


Key Idea: We can Divide Time and Space Attention Operators into 2 steps

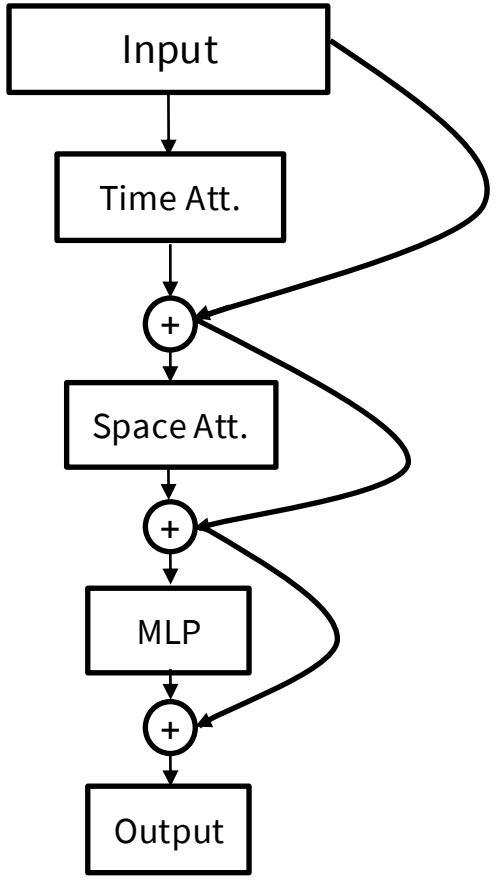


Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Joint vs Divided Space-Time Attention



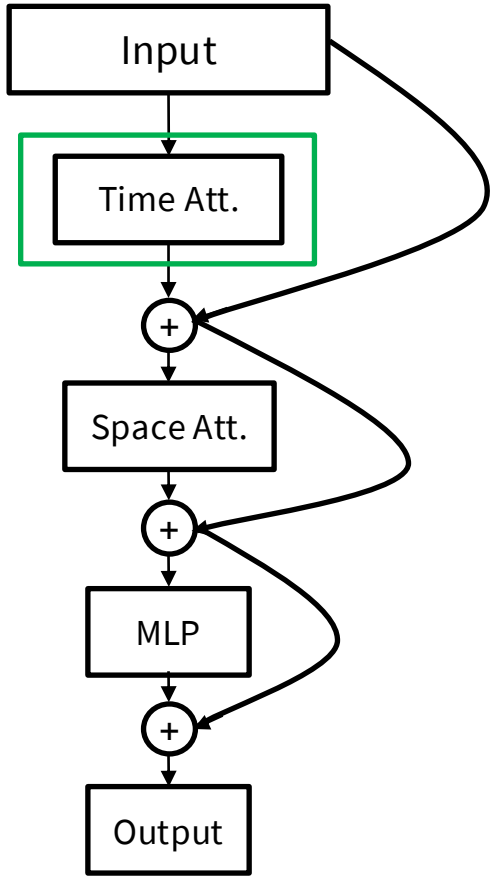
How will this help?
Each can operate on a different set of tokens

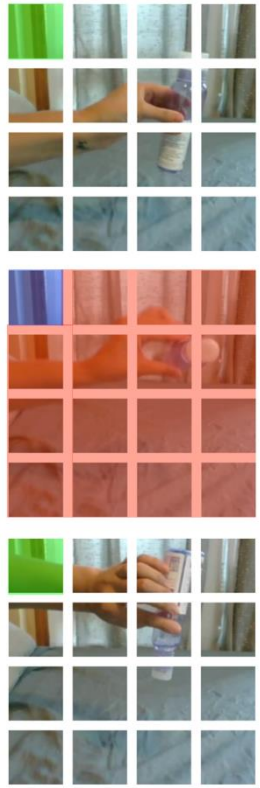


Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Divided Space-Time Attention

During "Time" Attention Operation, each token attends to the same spatial location across different frames.



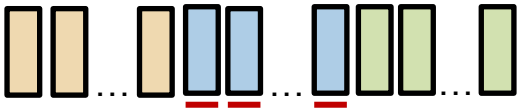


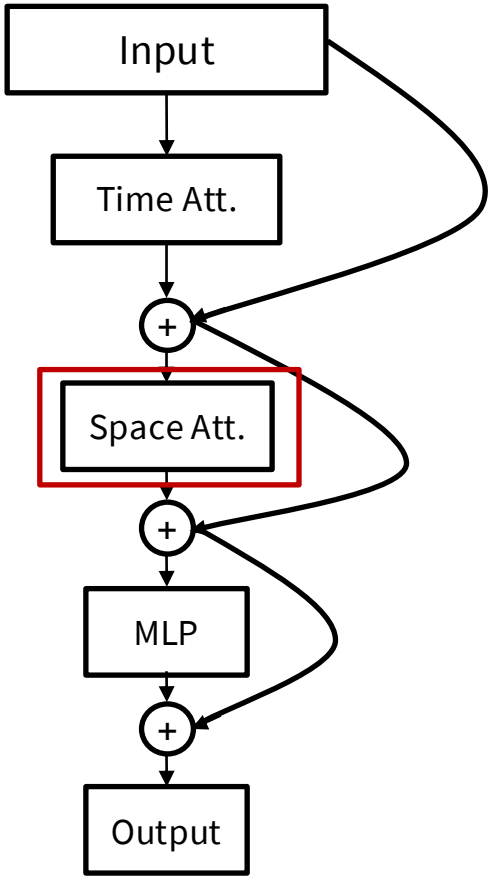
Divided Space-Time Attention (T+S)

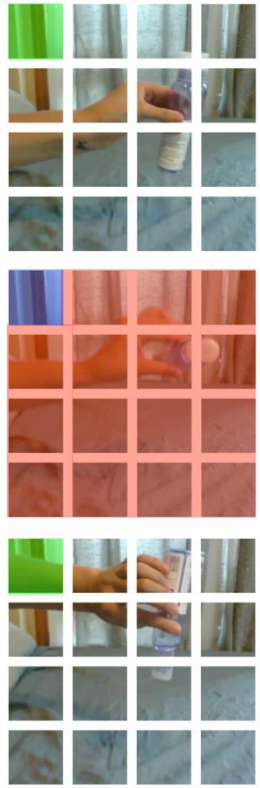
Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Divided Space-Time Attention

During "Space" Attention Operation, each token attends to other tokens within the same frame







Divided Space-Time Attention (T+S)

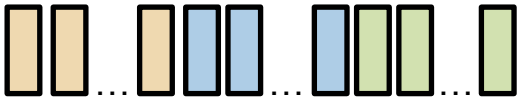
Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

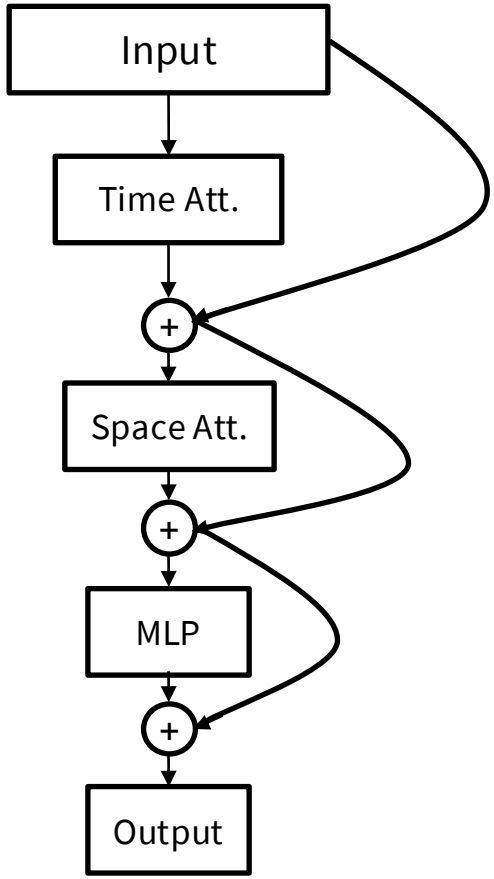
Divided Space Time Attention

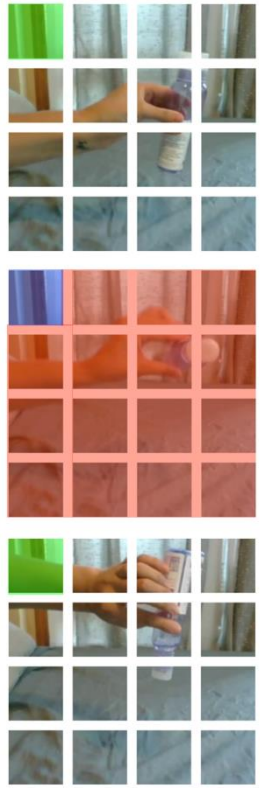
Why does this work well?

1. Reduces per-token computational cost from O(NT) to O(N + T)

2. Information can still propagate across space/time after many blocks.





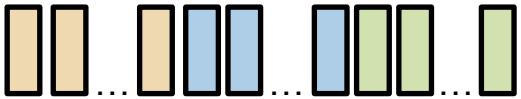


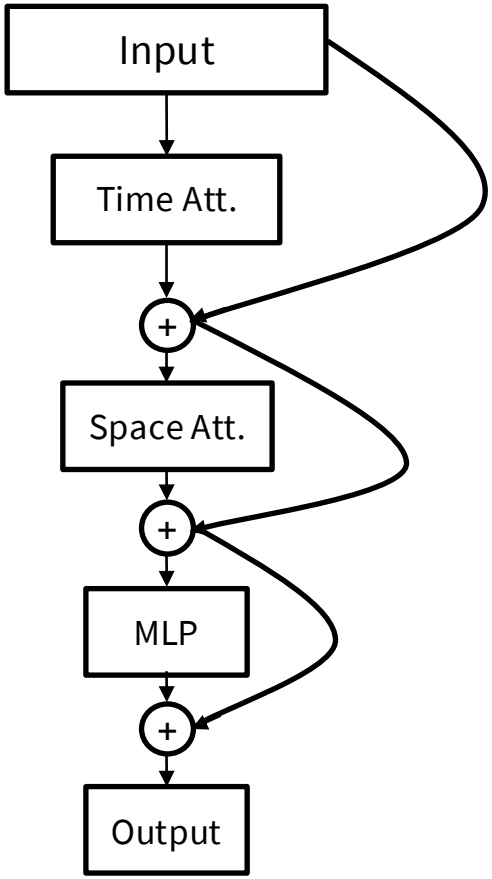
Divided Space-Time Attention (T+S)

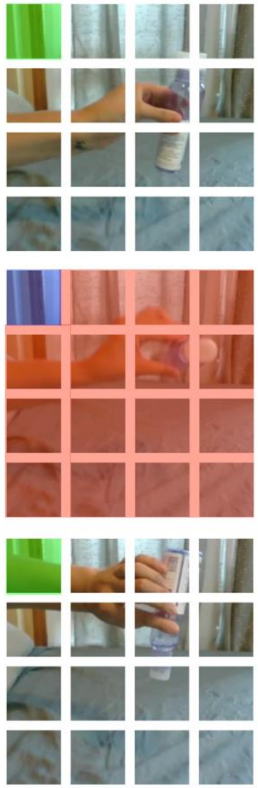
Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Divided Space Time Attention

But, there are other efficient attention operators that operate over small sets of tokens!







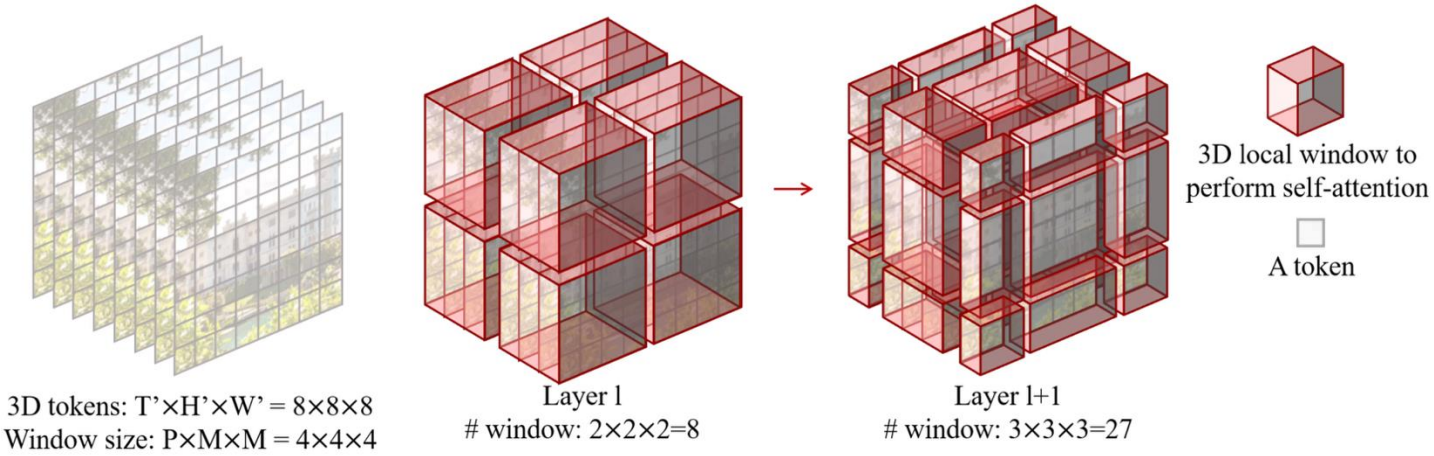
Divided Space-Time Attention (T+S)

Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

Approach #2: Video Swin Transformer

Key idea: Restrict self-attention to small local space-time cubes.
Very similar to 3D CNNs, but with self attention inside each cube.

Shift cubes between layers to allow information to pass across cube boundaries.



Figures from: Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

Approach #3: MViT

Multiscale Vision Transformers

Key idea: **Compress the size of K and V.**

(Output length remains unchanged)

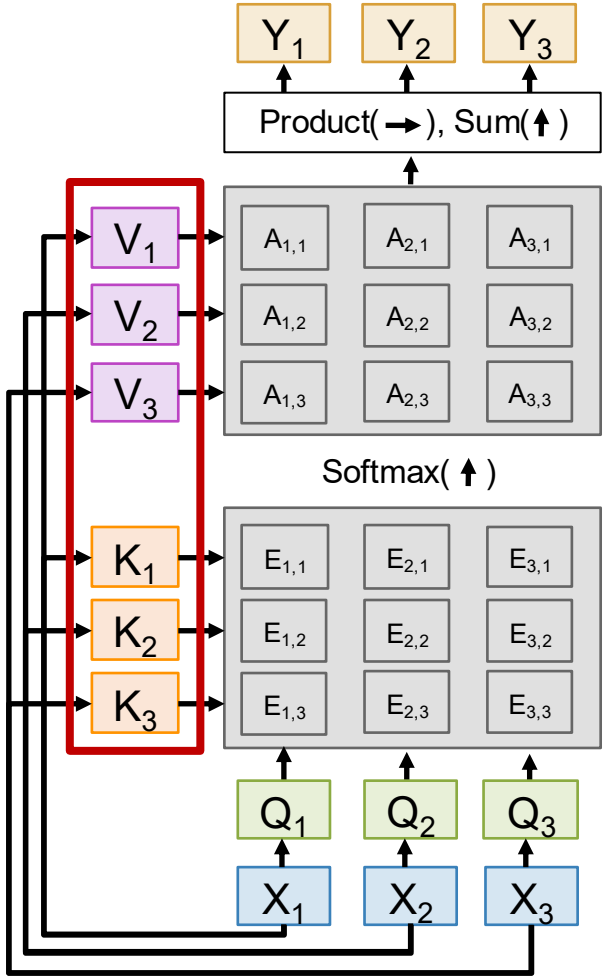
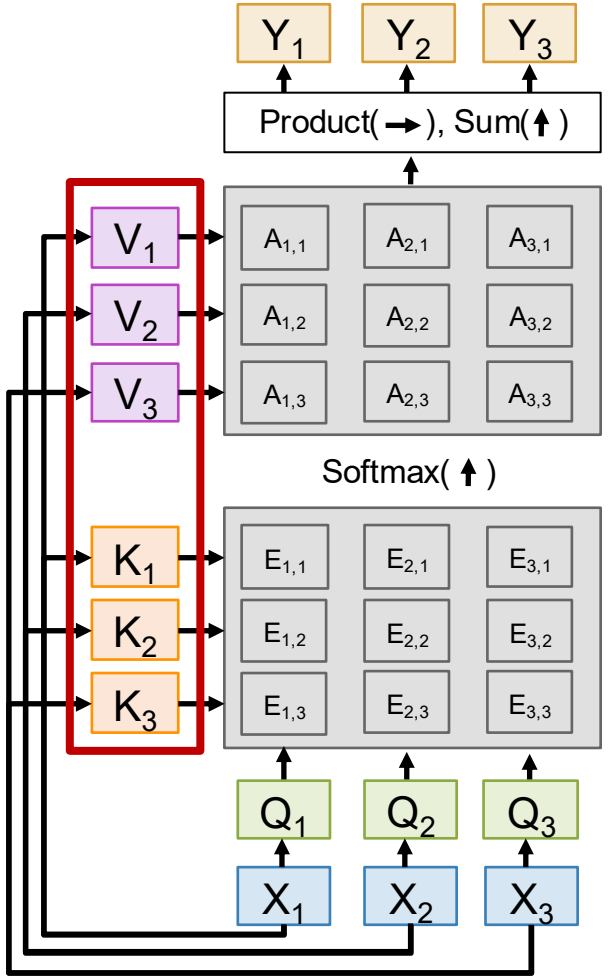


Figure from: Fan et al., "Multiscale Vision Transformers," ICCV 2021.

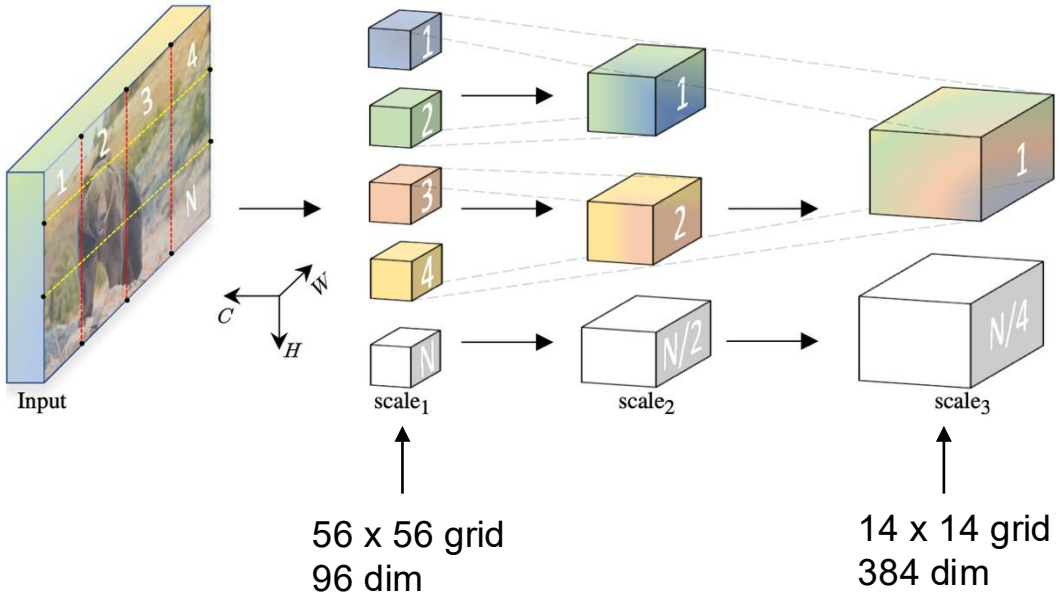
Multiscale Vision Transformers

Aggregate and shrink the K and V sequences via convolution before computing attention.

56 x 56 grid of K/V vectors becomes 14 x 14 grid (e.g., 4×4 kernel with stride 4)



Multiscale Vision Transformers

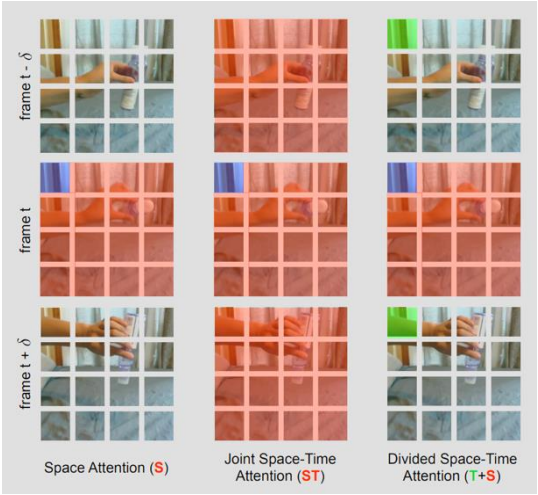


Progressively half spatial dimensions + double channels (same as ResNets)

Figure adapted from: Fan et al., "Multiscale Vision Transformers," ICCV 2021.

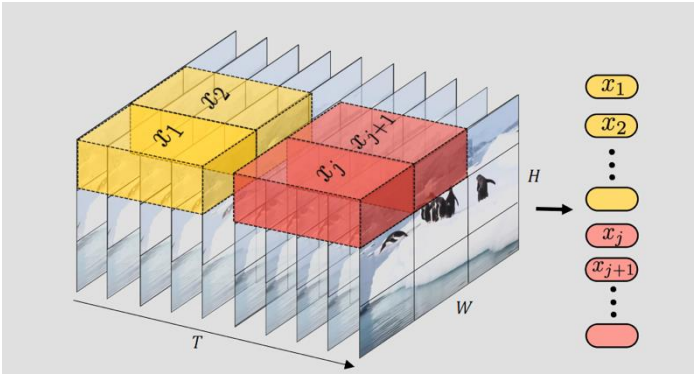
Two Broad Strategies for Improving Video Efficiency

Modify attention operator



Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

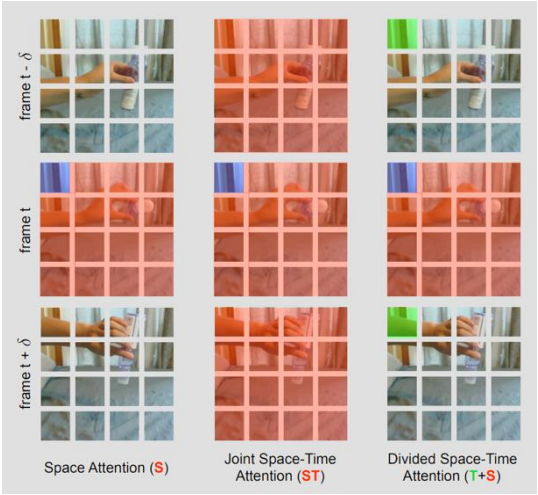
Reduce the number of tokens



Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

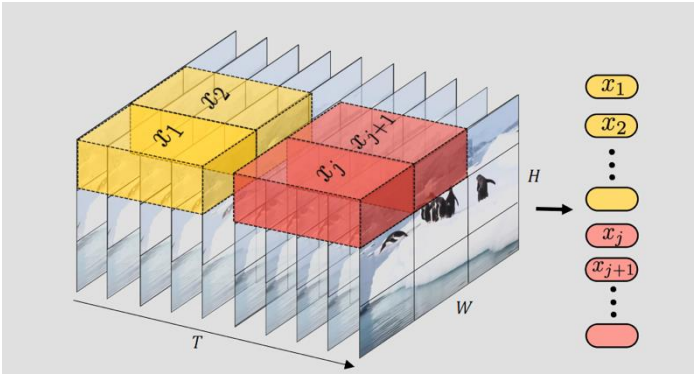
Two Broad Strategies for Improving Video Efficiency

Modify attention operator



Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

Reduce the number of tokens



Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

How to reduce the number of tokens?

One solution: Tubelets

More efficient than patches!

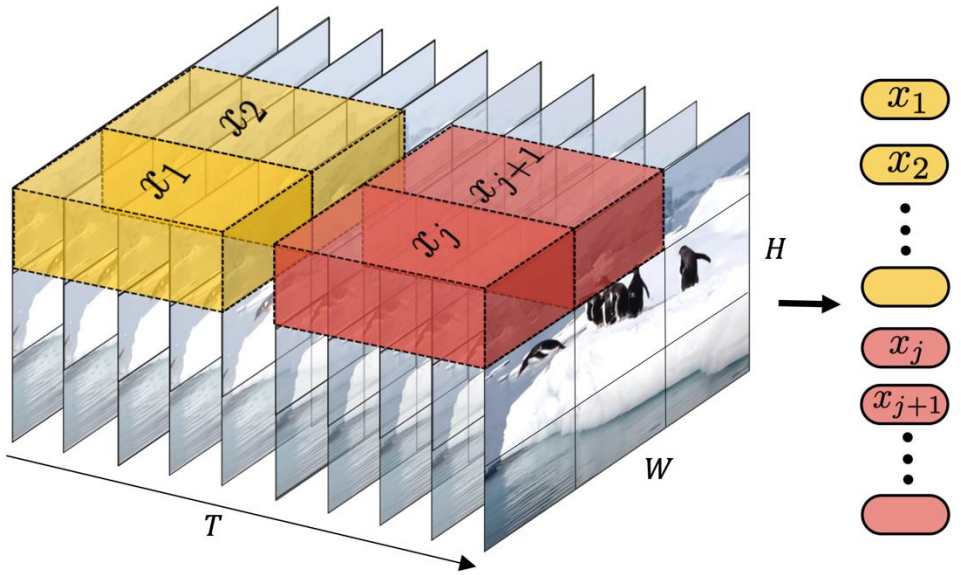


Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

How to reduce the number of tokens?

One solution: Tubelets

More efficient than patches!

Some intuition:

- Patches do not contain motion information, but tubelets do
- Significantly fewer tokens than patches!

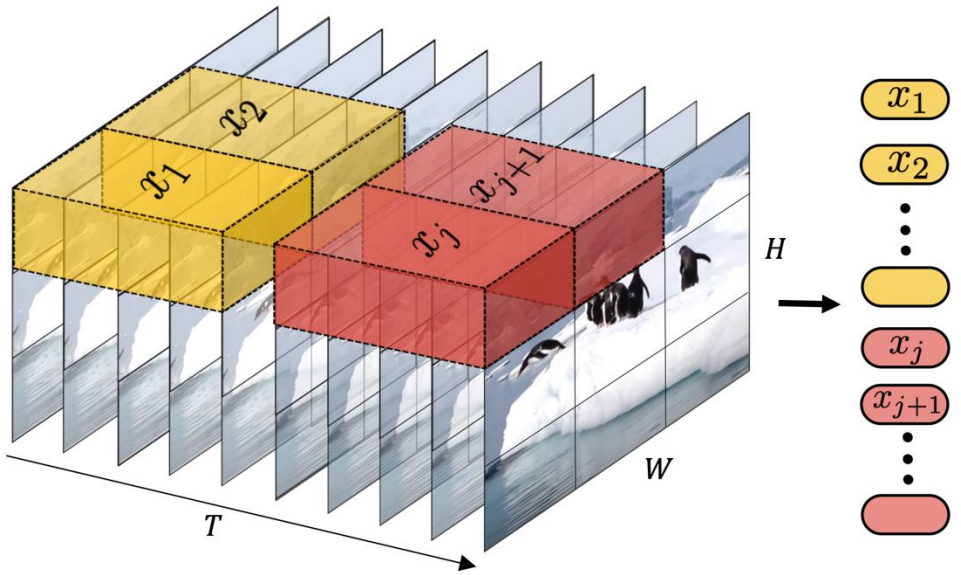


Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

How to reduce the number of tokens?

One solution: Tubelets

More efficient than patches!

Some intuition:
- Patches do not contain motion information, but tubelets do
- Significantly fewer tokens than patches!

Tubelets over 4 frames:
4x reduction in tokens → 16x reduction in compute

Still project from tubelet dim to the transformer dim.

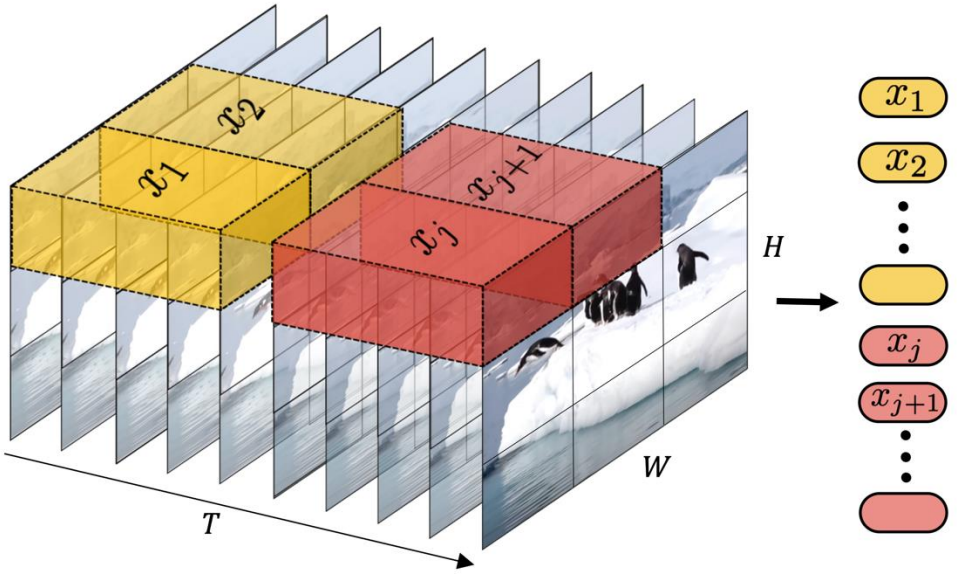


Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

How to reduce the number of tokens?

Employed by *tons* of video understanding transformers (e.g. ViViT, VideoMAE, Video Swin, MViT, V-JEPA)

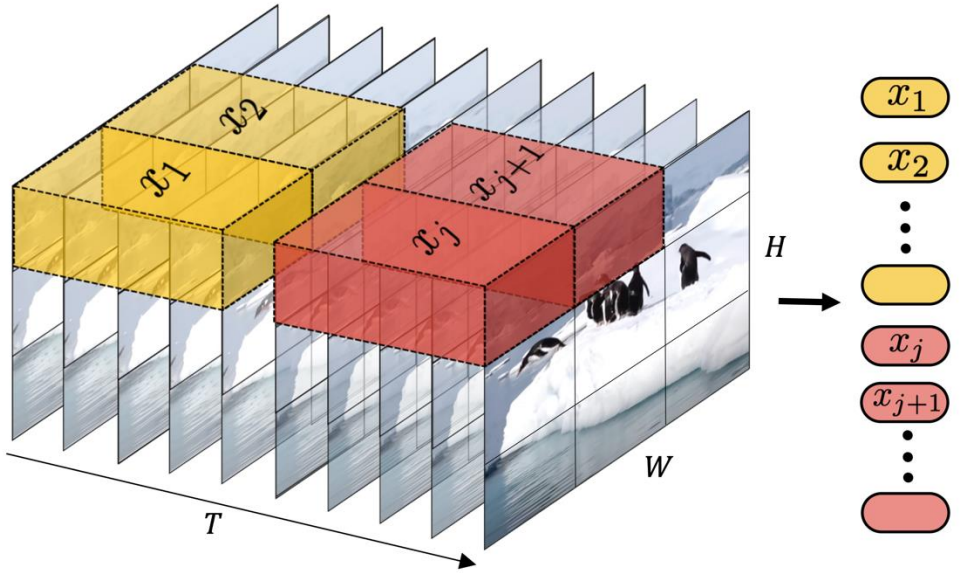


Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

How to reduce the number of tokens?

Lots of other approaches to reduce token count

(e.g. adaptive token selection, token merging, learned compression, etc.)

Will discuss some in the lecture 16!

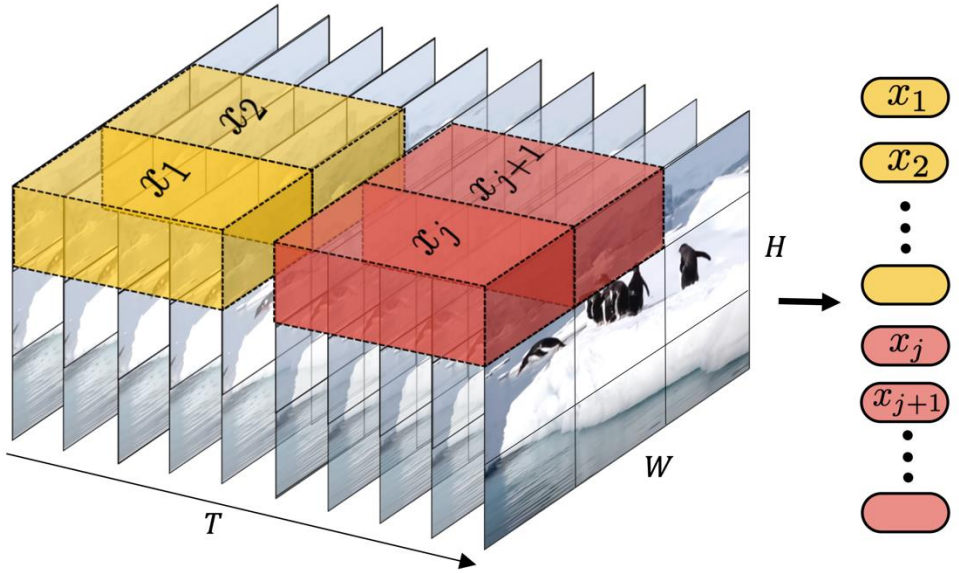


Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

So far: Classify short clips

Videos: Recognize actions

Swimming
Running
Jumping
Eating
Standing

Temporal Action Localization

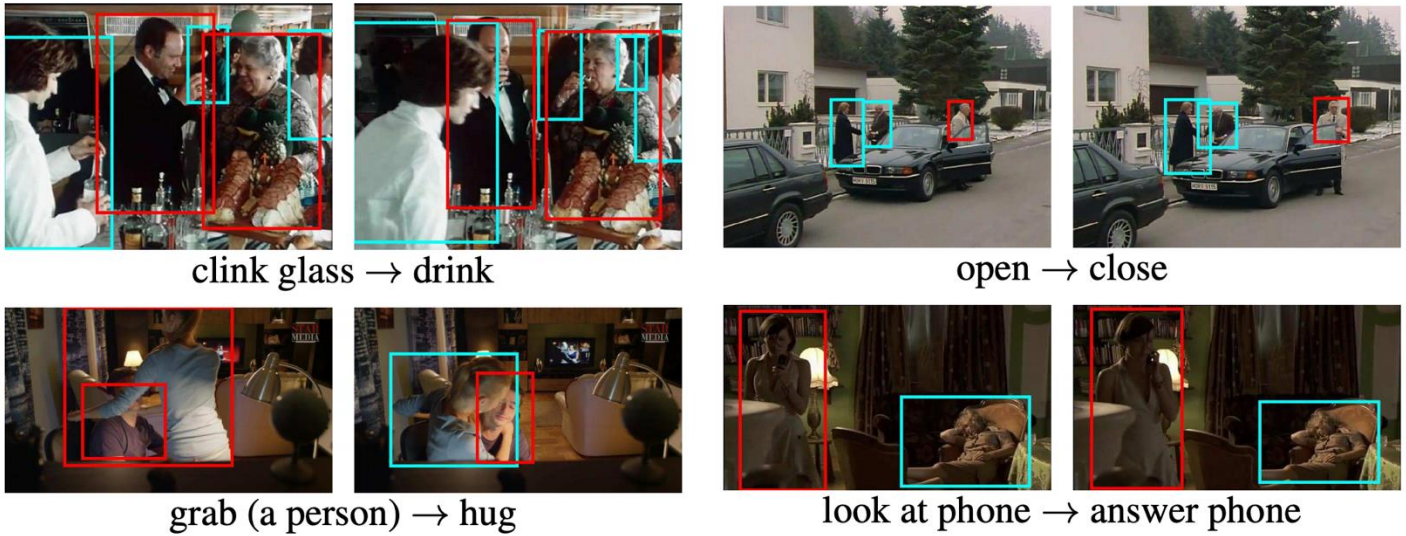
Given a long untrimmed video sequence, identify frames corresponding to different actions

Can use architectures similar to Faster R-CNN: first generate temporal proposals then classify

Chao et al, " Rethinking the Faster R-CNN Architecture for Temporal Action Localization", CVPR 2018

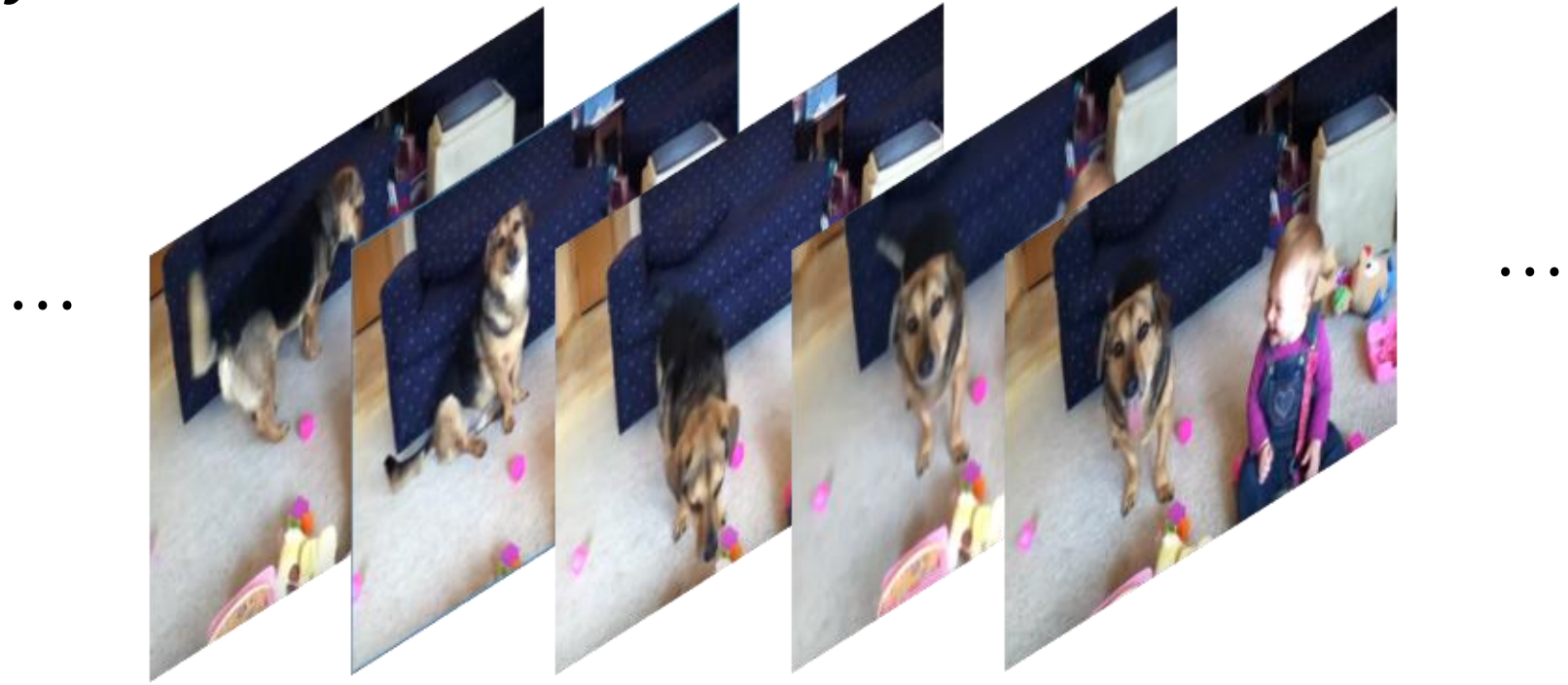
Spatio-Temporal Detection

Given a long untrimmed video, detect all the people in both space and time and classify the activities they are performing. Some examples from AVA Dataset:



Gu et al, "AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions", CVPR 2018

Beyond Short Video Streams



Audio, Language, and Long-form Video Understanding…

Video source: BBC

(McGurk & McDonald 1976)

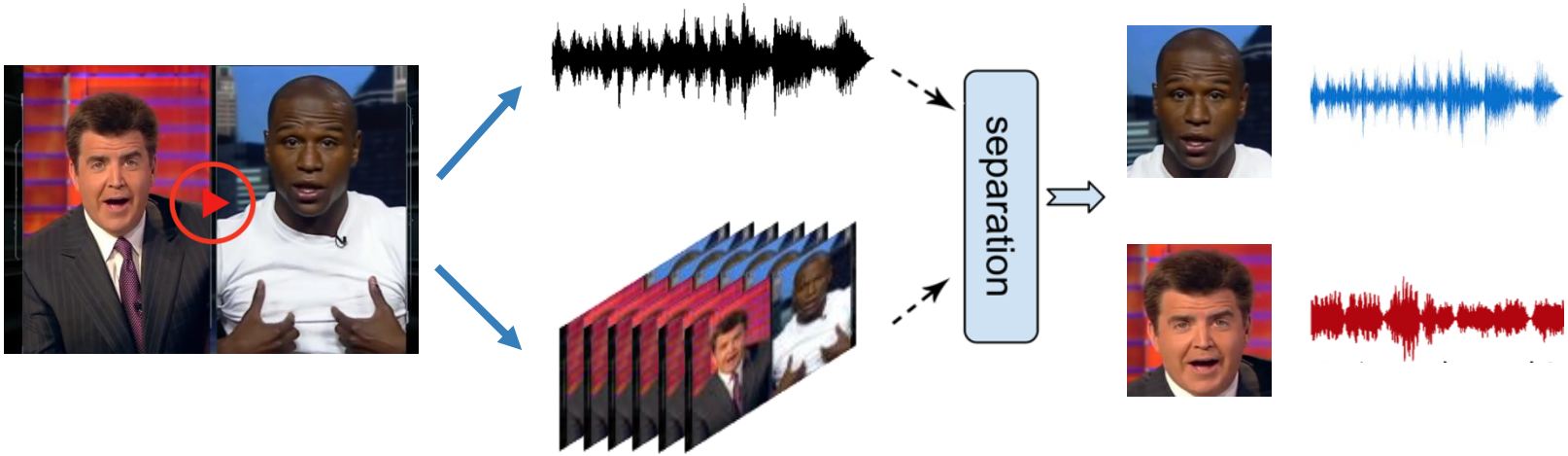
Video source: BBC

(McGurk & McDonald 1976)

Video source: BBC

(McGurk & McDonald 1976)

Visually-guided audio source separation



[Gao et al. ECCV 2018, Afouras et al. Interspeech'18, Gabby et al. Interspeech'18, Owens & Efros ECCV'18, Ephrat et al. SIGGRAPH'18, Zhao et al. ECCV 2018, Gao & Grauman ICCV 2019, Zhao et al. ICCV 2019, Xu et al. ICCV 2019, Gan et al. CVPR 2020, Gao et al. CVPR 2021, Tzinis et al. ECCV 2022, Chen et al. CVPR 2023]

Gao et al., VisualVoice, CVPR 2021

Gao et al., VisualVoice, CVPR 2021

Gao et al., VisualVoice, CVPR 2021

Musical instruments source separation

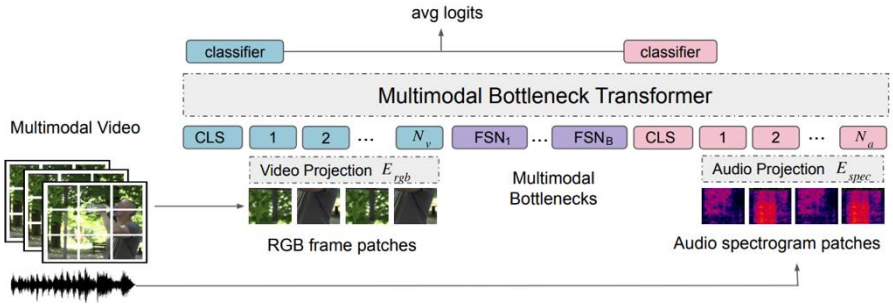
Train on 100,000 unlabeled multi-source video clips,
then separate audio for novel video.

original video
(before separation)

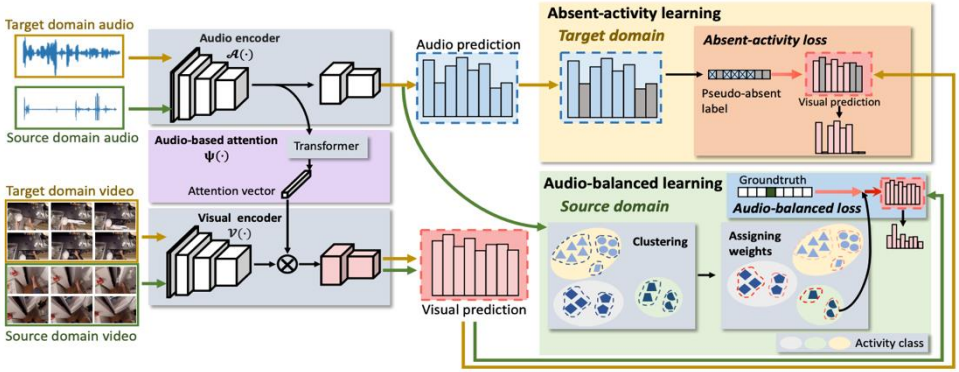
object detections:
violin & flute

Gao & Grauman, Co-Separating Sounds of Visual Objects, ICCV 2019

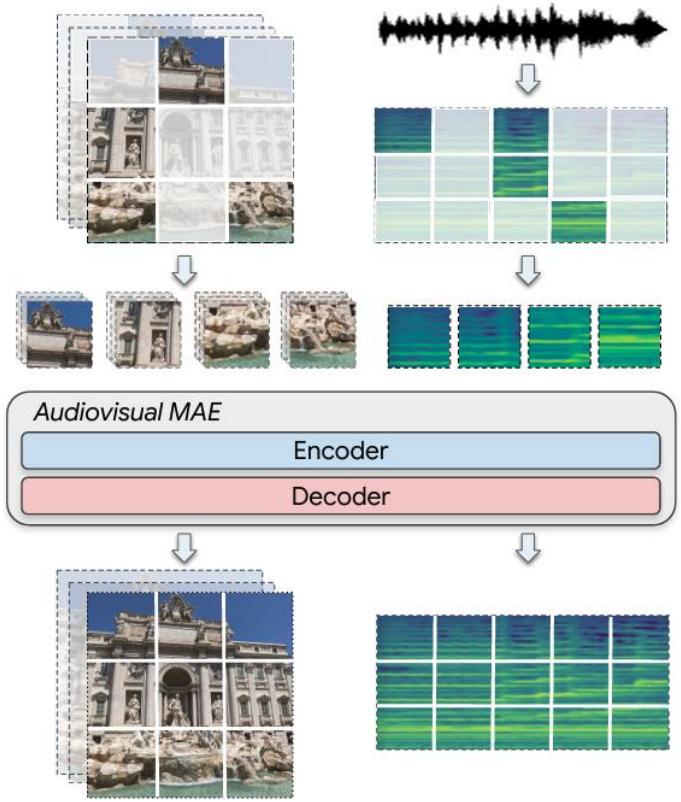
Audio-Visual Video Understanding



Attention Bottlenecks for Multimodal Fusion, Nagrani et al. NeurIPS 2021



Audio-Adaptive Activity Recognition Across Video Domains, Zhang et al. CVPR 2022



Audio-Visual Masked Autoencoders. Georgescu et al. ICCV 2023.

Efficient Video Understanding

Action recognition in long videos

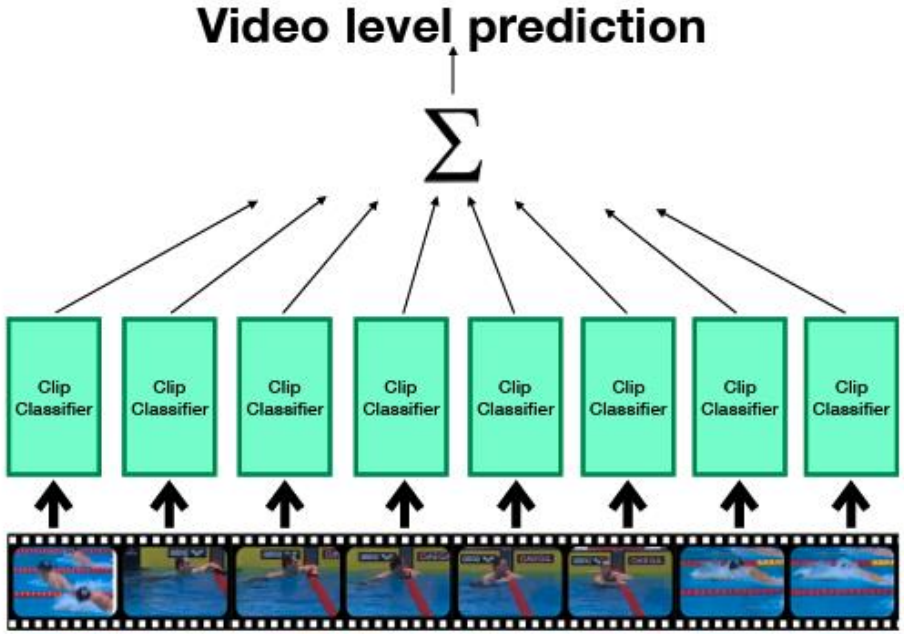
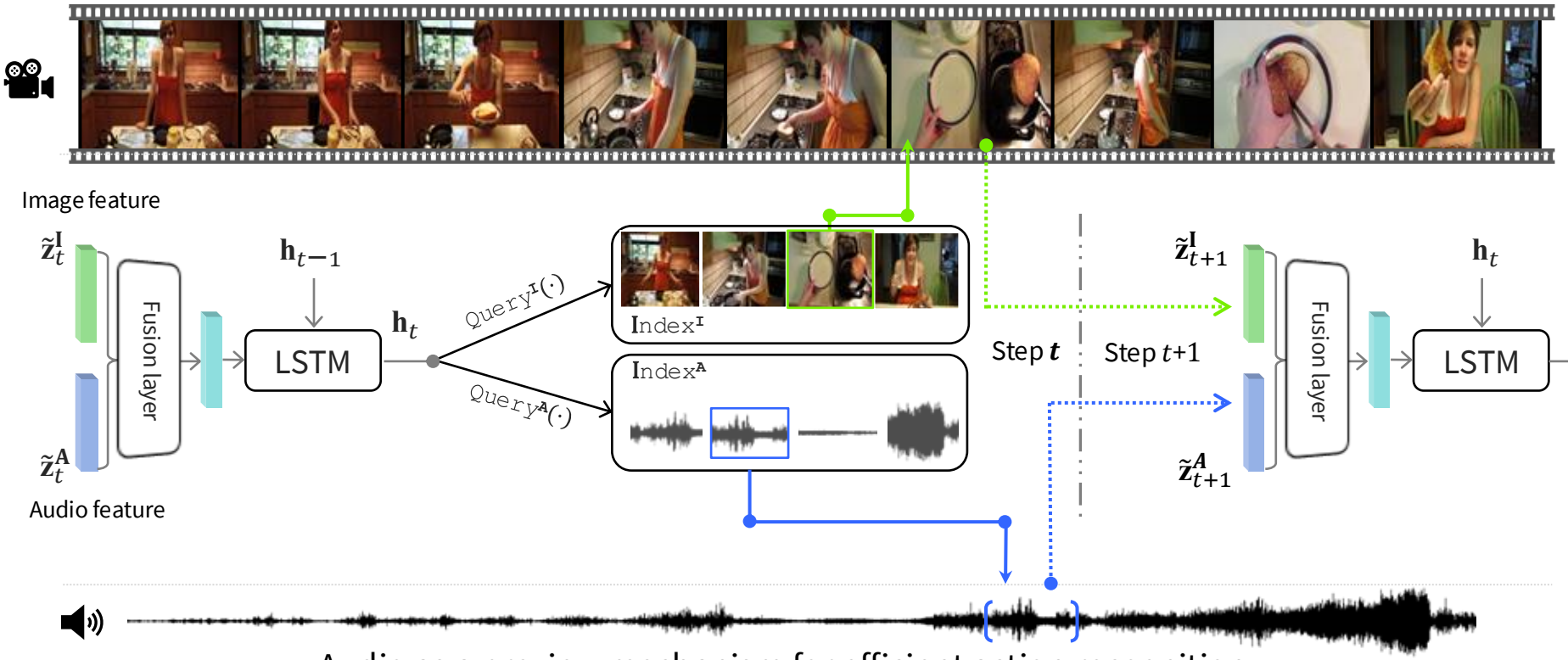


Image Credit: Korbar et al.

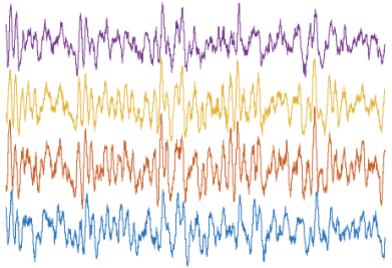
Efficient Video Understanding



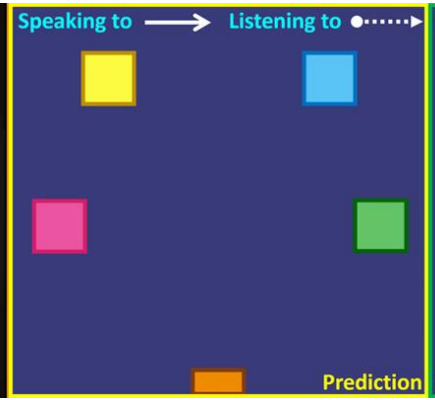
Audio as a preview mechanism for efficient action recognition

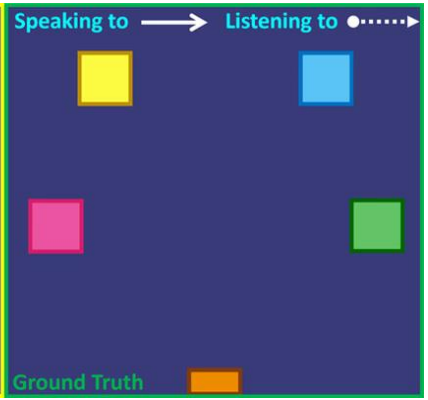
Gao et al., Listen to Look: Action Recognition by Previewing Audio, CVPR 2020

Multimodal Egocentric Video Understanding



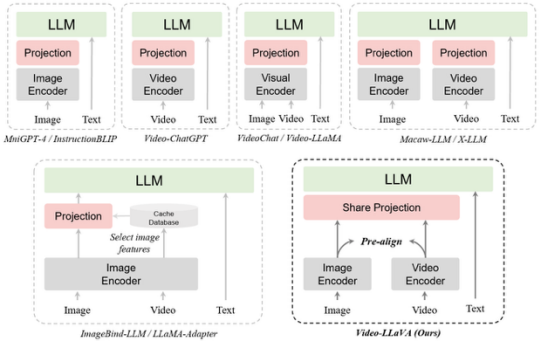
Ego-Exo Conversational Graph Prediction



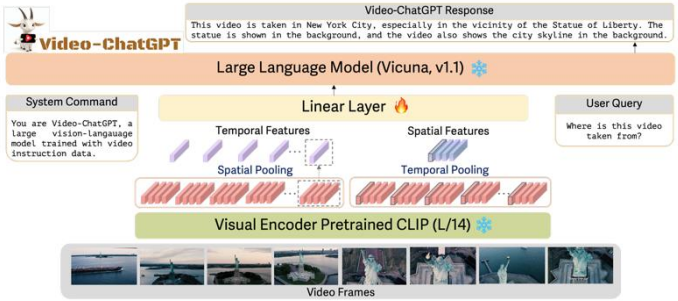


The Audio-Visual Conversational Graph: From and Egocentric-Exocentric Perspective. Jia et al. CVPR 2024

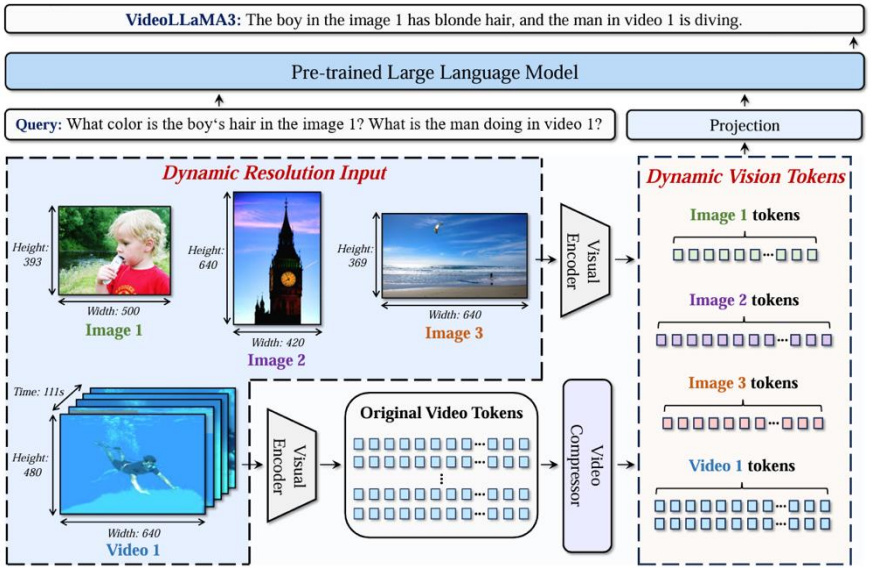
Video Language Understanding (VideoLLMs)



Video-LLaVA: Learning United Visual Representations by Alignment Before Projection. Lin et al. EMNLP 2024

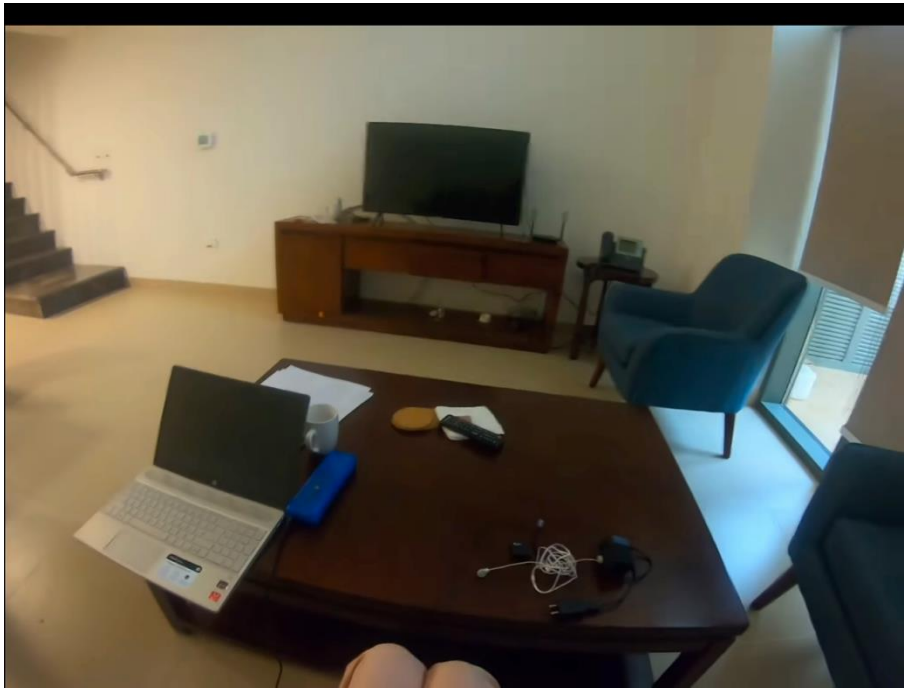


Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. Maaz et al. ACL 2024.



VideoLLaMA 3: Frontier Multimodal Foundation Models for Video Understanding. Zhang et al. arXiv 2025

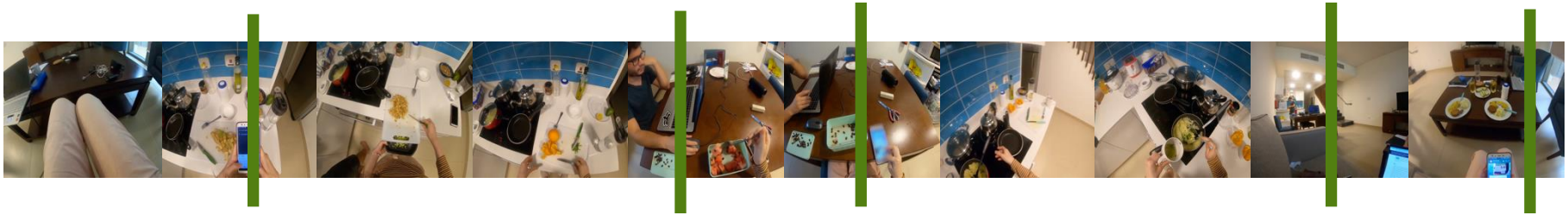
Long-form Video Understanding



Humans demonstrate a remarkable ability to process visual stimuli over long time horizons, enabling them to perceive, plan and act in the real world. Can today's computer vision systems understand long-form videos?

Long-form Video Understanding

01:10:26



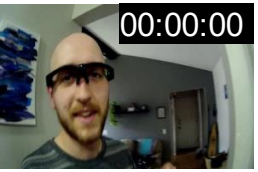
Where did I leave my AirPods after working out?

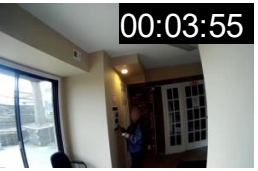
How to test? -> Ask Questions about the Video!

Long-form Video Understanding (Example from HourVideo)

Identify the unique individuals the camera wearer interacted with.

00:30:00





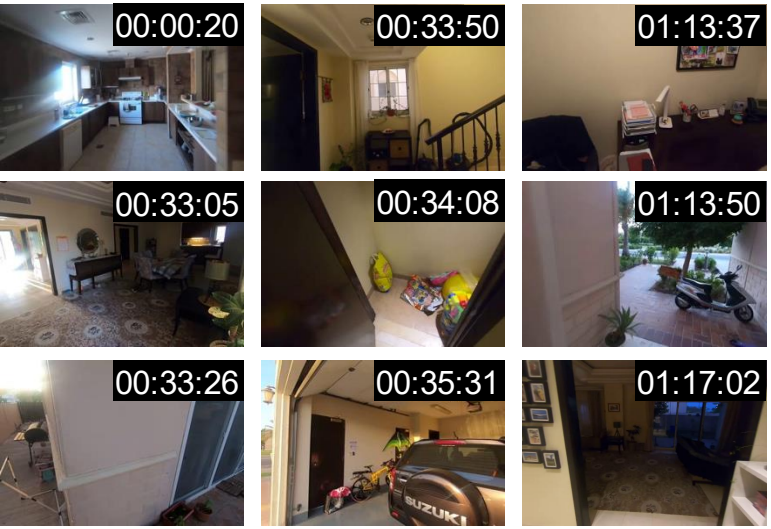
- [ ] 2 Adults
- [ ] 1 Adult
- [ ] 4 Adults
- [ ] 5 Adults
- [x] 3 Adults

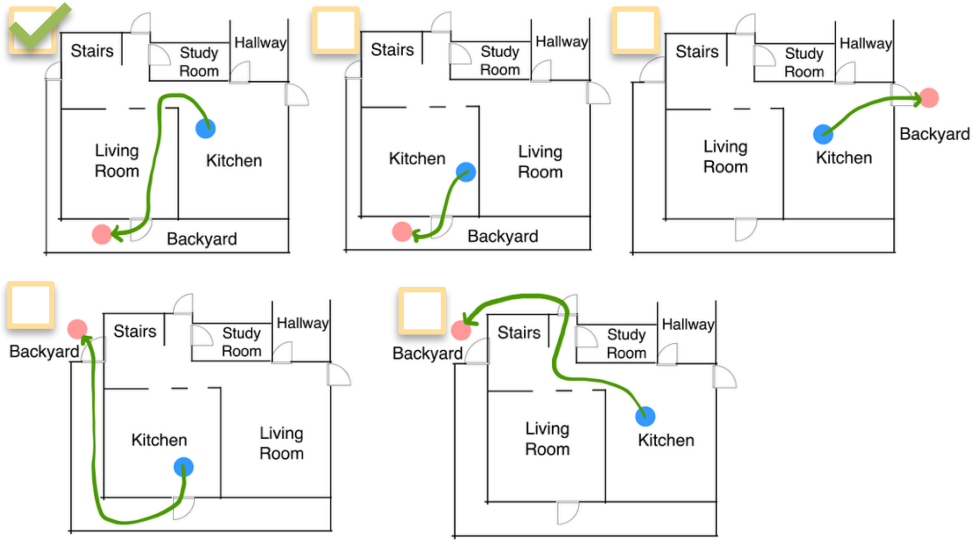
Chandrasegaran et al. & Fei-Fei. HourVideo: 1-Hour Video-Language Understanding. NeurIPS 2024

Long-form Video Understanding (Example from HourVideo)

How can the camera wearer get to the backyard from the kitchen?

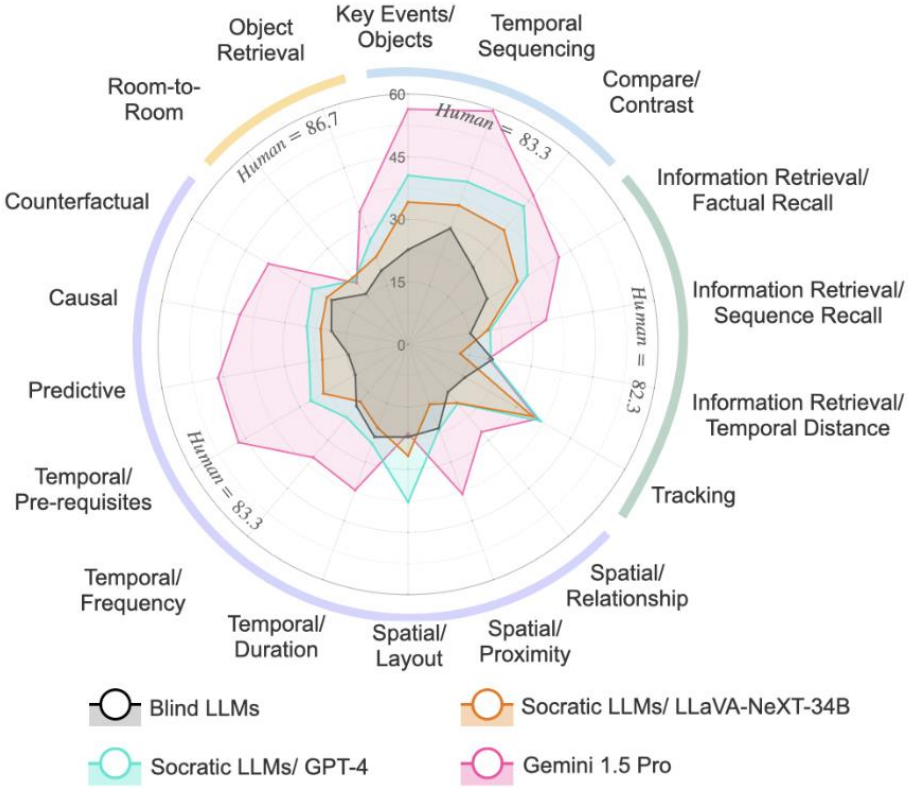
01:17:11





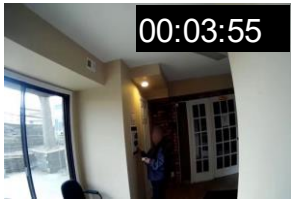
Chandrasegaran et al. & Fei-Fei. HourVideo: 1-Hour Video-Language Understanding. NeurIPS 2024

Long-form Video Understanding (Results from HourVideo)



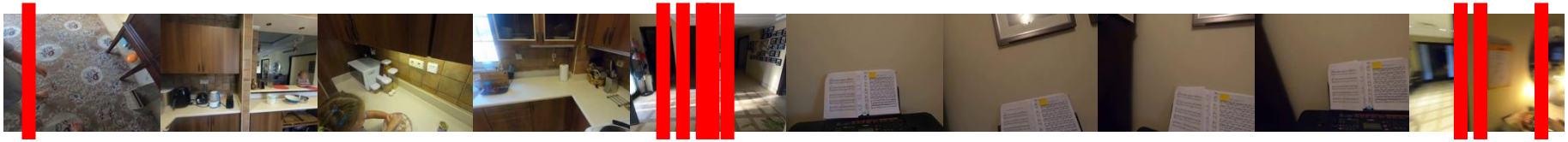
Takeaway: Significant gap in long-form video understanding capabilities -> Lots of work to do!

Identify the unique individuals the camera wearer interacted with.

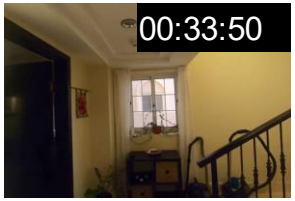


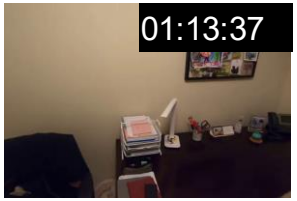
MCQ Test

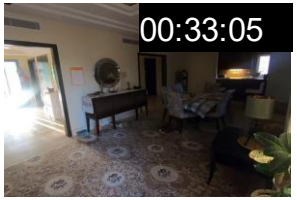
- [ ] 2 Adults
- [ ] 1 Adult
- [ ] 4 Adults
- [ ] 5 Adults
- [x] 3 Adults

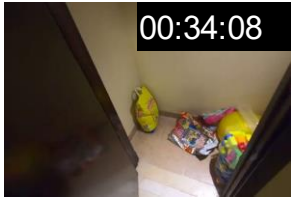


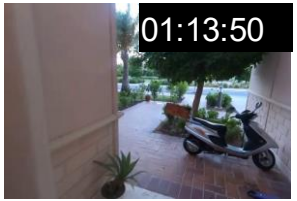
How can the camera wearer get to the backyard from the kitchen?

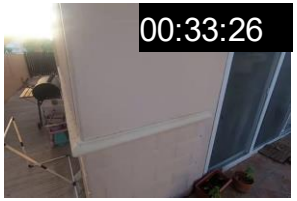


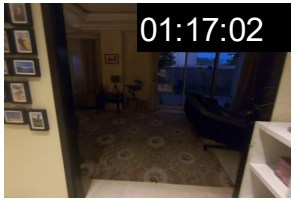




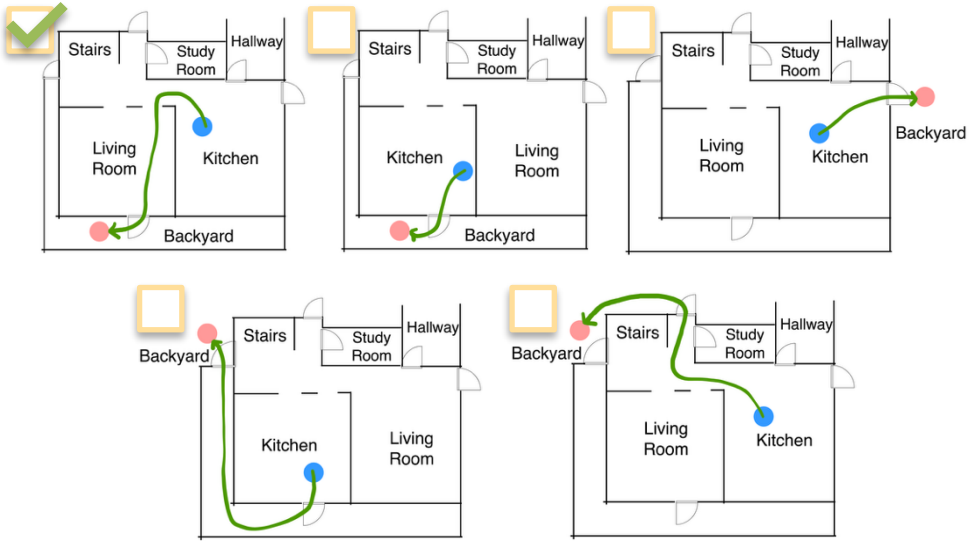








MCQ Test



Next time: Large Scale Distributed Training