Week 14

Video Understanding

[ECEA0649/ECE40049] Deep Learning for Image Processing | Spring 2026

Kihyun Na (Research Professor)
BK21 AI Education and Research Group &
Institute for Information and Communication Technology,
Handong Global University

HANDONG GLOBAL UNIVERSITY

Curriculum

** Adjusted based on survey results*

Wk 1	OT + Introduction
Wk 2	DL Fundamentals Review
Wk 3	CNN
Wk 4	From Sequence Modeling to Transformer
Wk 5	Transformer in Vision
Wk 6	Detection & Segmentation
Wk 7	Self-Supervised Learning
Wk 8	Review Literacy + Role Explanation
Wk 9	Foundation Models (CLIP, SAM) + Paper #1
Wk 10	Diffusion Models + Paper #2
Wk 11	Diffusion Models (2) + Paper #3
Wk 12	Vision-Language Models + Paper #4
Wk 13	VLM Applications (Paper #5)
Wk 14	Video Understanding + Paper #6
Wk 15	Embodied AI & Robot Vision (Paper #7)
Wk 16	Miniconference (Final Project)

Lecture Lecture + Paper Miniconference

Today's Agenda

01

Video = Image + Time

What changes when we add the temporal dimension?

10 min

02

The CNN Era

From single frames to 3D convolutions

15 min

03

The Transformer Era

From local convolutions to global attention

15 min

04

Video + Foundation Models

Generation and understanding in the modern era

10 min

Part 1

Video = Image + Time

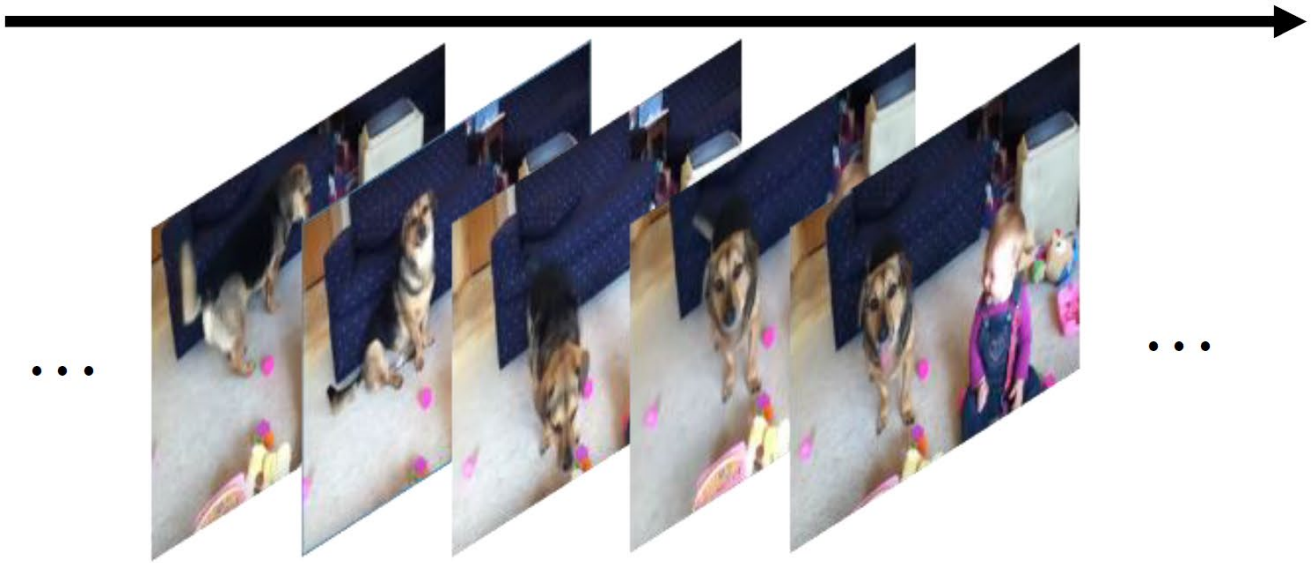
What changes when we add the temporal dimension?

What is Video?

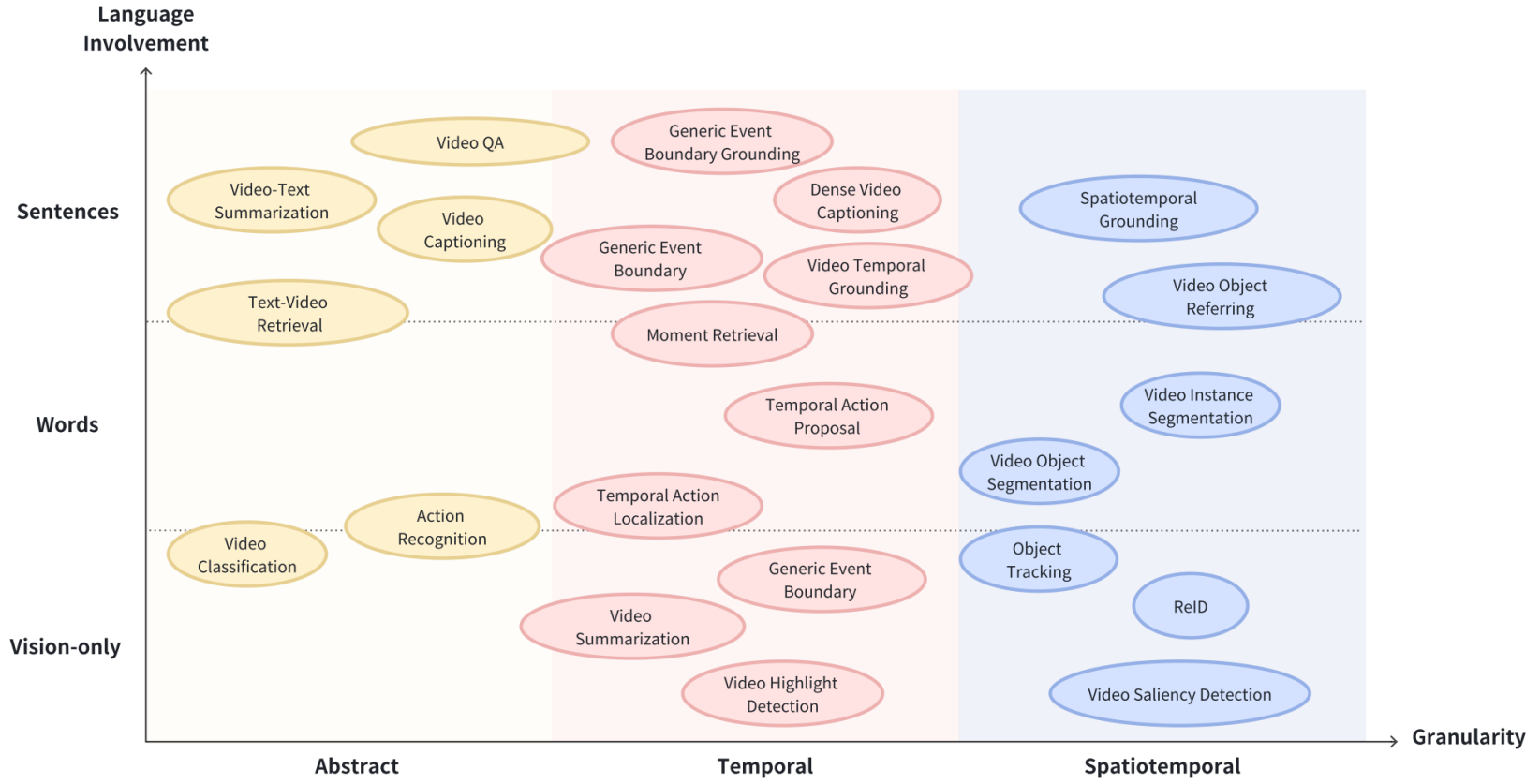
Video = Image + Time

A video is a sequence of images

4D tensor: T x 3 x H x W



Tasks in video understanding



Source: Tang et al., 2024

Example task: Video Classification

Image Classification vs Video Classification

Images: Recognize objects

Dog
Cat
Fish
Truck

Videos: Recognize actions

Swimming
Running
Jumping
Eating
Standing

Running video is in the public domain

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 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

From This Course to Video

Every technique we learned extends to video

	Image	Video (today)
CNN (Wk 3)	2D conv, 3×3 kernel	3D CNN: 3×3×3 kernel (C3D, I3D)
ViT (Wk 5)	Patch embedding, self-attention	TimeSformer: spatiotemporal patches
SSL (Wk 7)	MAE: mask 75%	VideoMAE: mask 90-95% (use redundancy!)
CLIP (Wk 9)	Image-text alignment	VideoCLIP, InternVideo
Diffusion (Wk 10-11)	2D latent diffusion	Sora: 3D spatiotemporal latent
VLM (Wk 12)	Image tokens → LLM	Video-LLM: frame tokens → LLM

Part 2

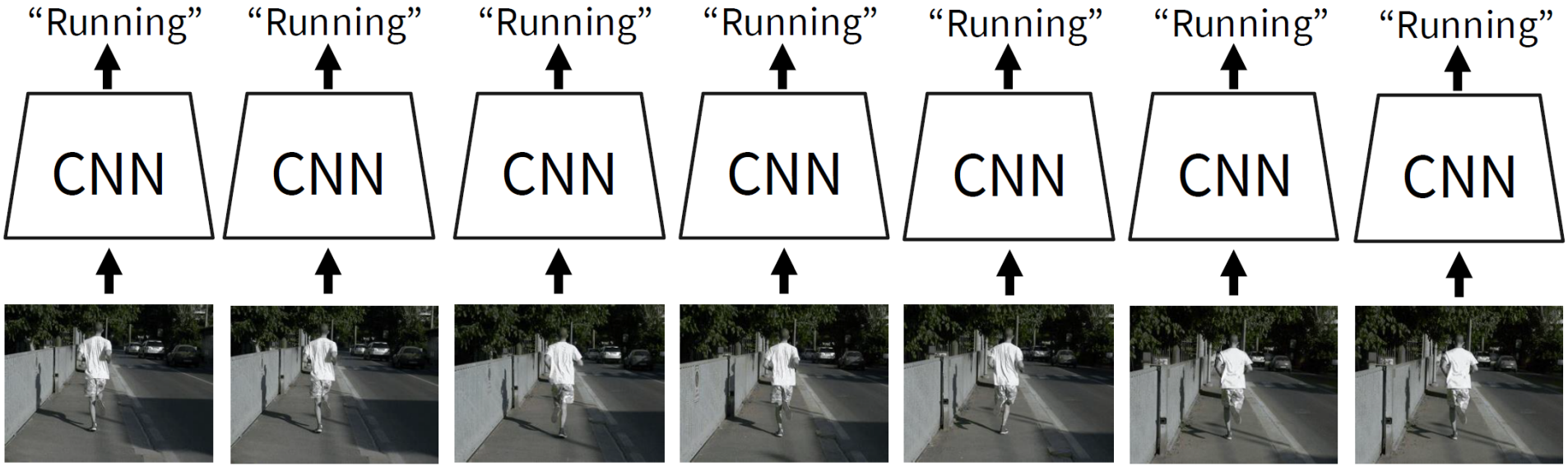
The CNN Era

From single frames to 3D convolutions

How to Handle Time?: Single-Frame CNN

Process each frame independently and average them

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



How to Handle Time?: Late Fusion (MLP)

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

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

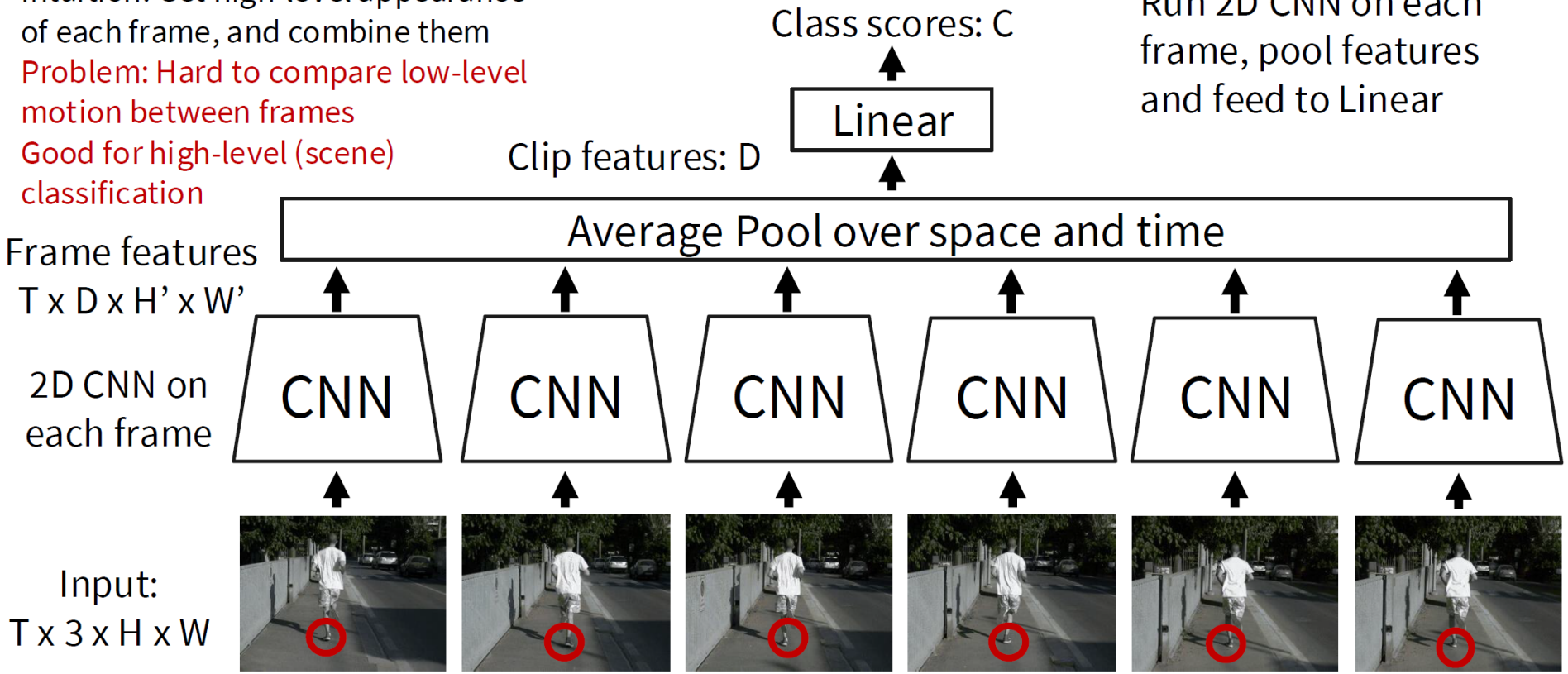
How to Handle Time?: Late Fusion (Pooling)

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



How to Handle Time?: Early Fusion (2D CNN)

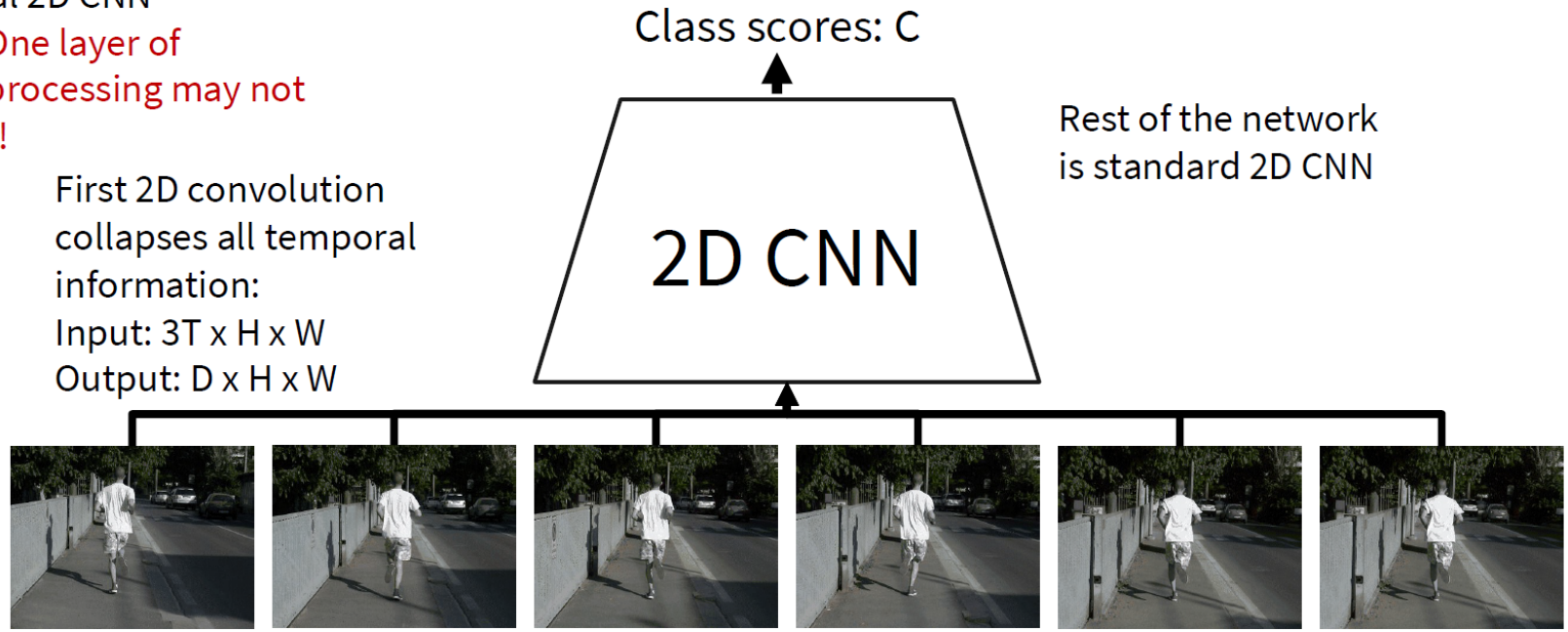
Intuition: Compare frames with very first conv layer, after that normal 2D CNN

Problem: One layer of temporal processing may not be enough!

First 2D convolution collapses all temporal information:
Input: 3T x H x W
Output: D x H x W

Reshape:
3T x H x W

Input:
T x 3 x H x W

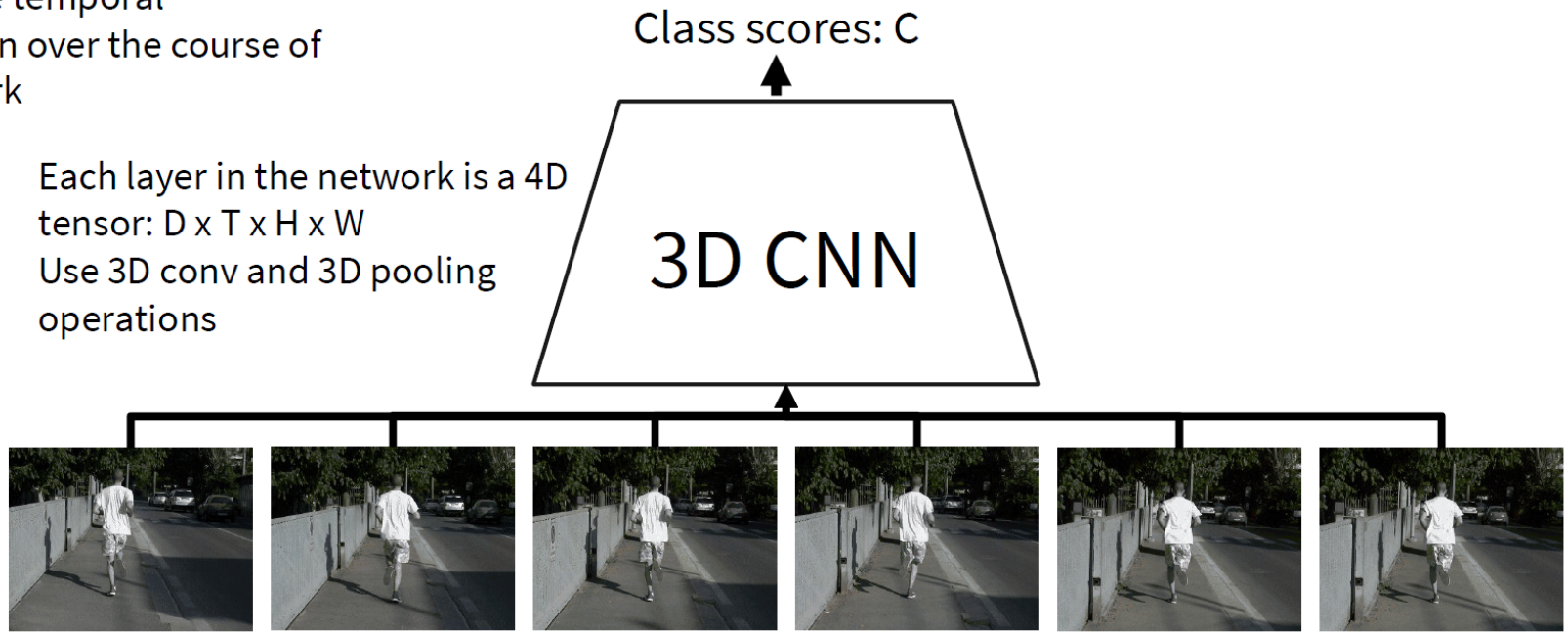


Karpathy et al, "Large-scale Video Classification with Convolutional Neural Networks", CVPR 2014

How to Handle Time?: Early Fusion (3D CNN)

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

Each layer 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

3D Convolution: The Key Insight

2D Conv (Early Fusion)

Input: C_in × T × H × W

Kernel: C_out × C_in × T × 3 × 3

slides over space only

Problem: no temporal shift-invariance!

Needs to **learn separate filters** for the same motion at different times

3D Conv

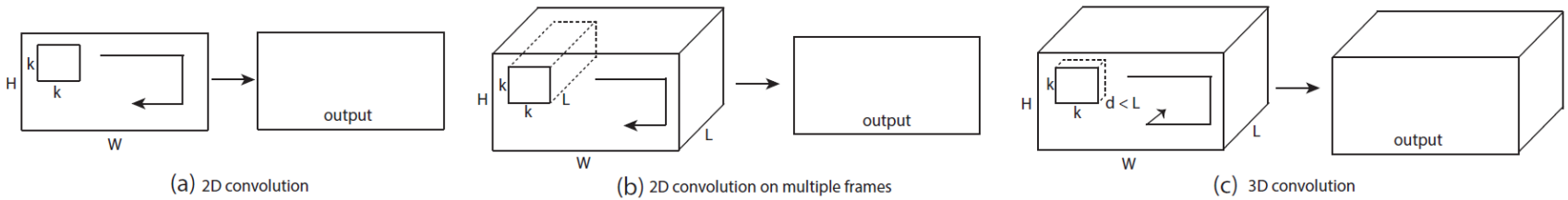
Input: C_in × T × H × W

Kernel: C_out × C_in × 3 × 3 × 3

slides over space AND time

Temporal shift-invariant!

Same filter detects the same motion pattern regardless of when it occurs

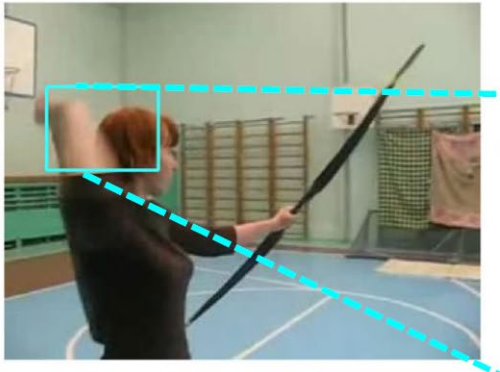


(a) 2D convolution (b) 2D convolution on multiple frames (c) 3D convolution

Figure 1. **2D and 3D convolution operations**. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

Optical Flow & Two-Stream Networks

Image at frame t



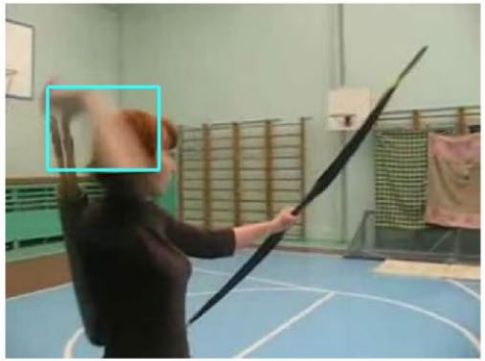
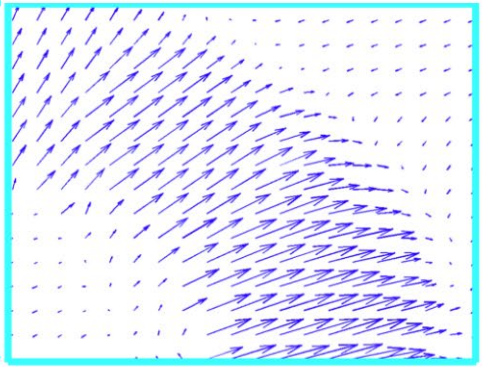


Image at frame t+1

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

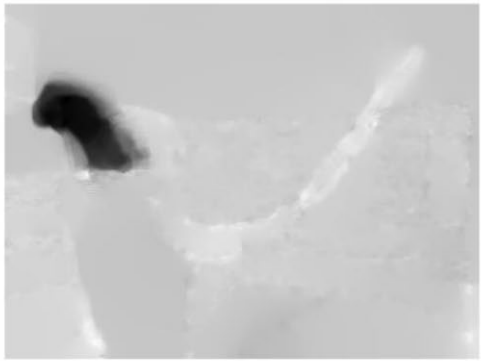


Tells where each pixel will move in the next frame:

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

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

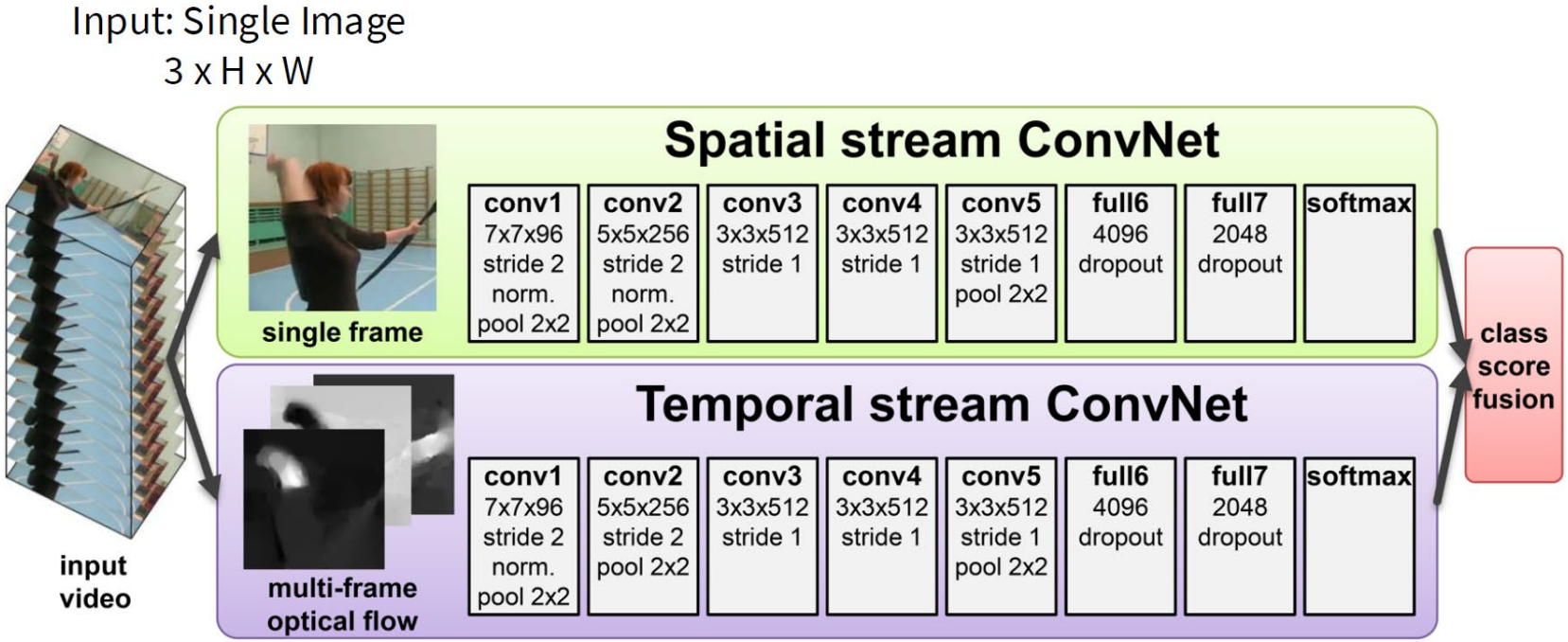
Horizontal flow dx



Vertical Flow dy

Optical Flow & Two-Stream Networks

Simonyan & Zisserman, NeurIPS 2014 — Separate appearance and motion



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

Early fusion: First 2D conv
processes all flow images

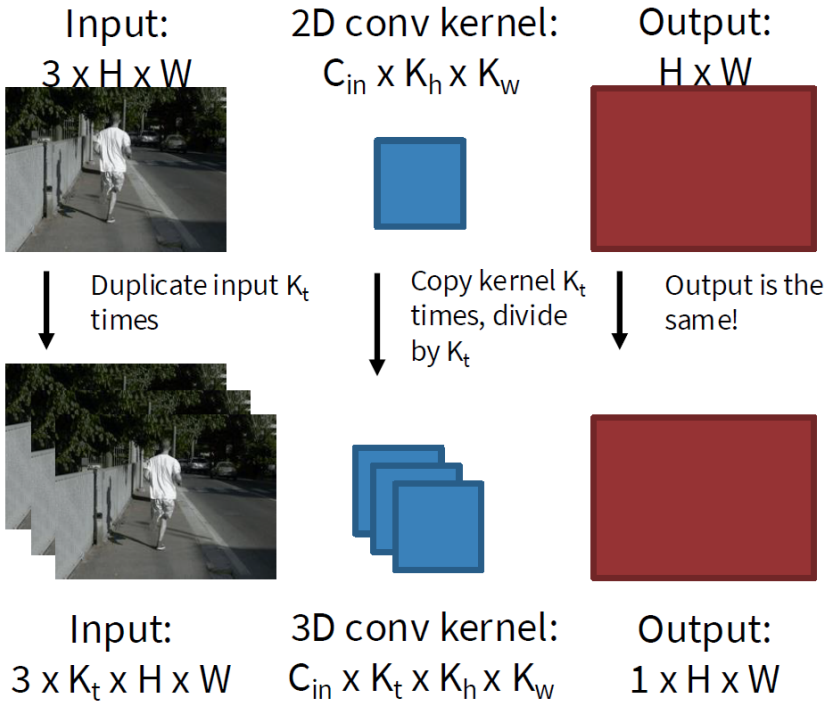
Inflating 2D Networks to 3D (I3D)

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

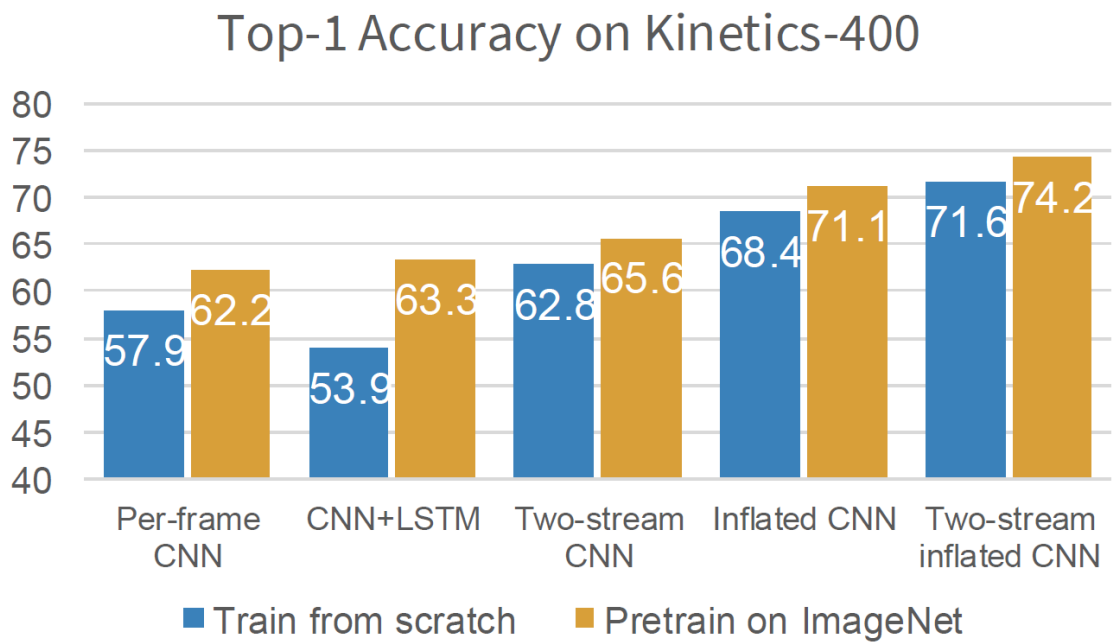
Inflating 2D Networks to 3D (I3D)

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

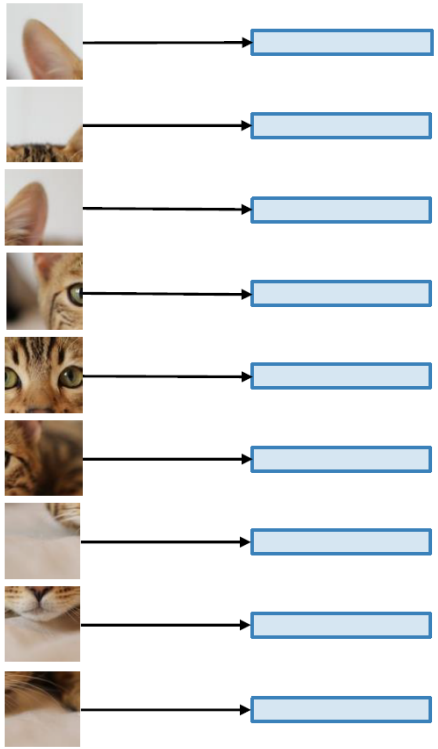
Part 3

The Transformer Era

From local convolutions to global attention

Vision Transformers in Video

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

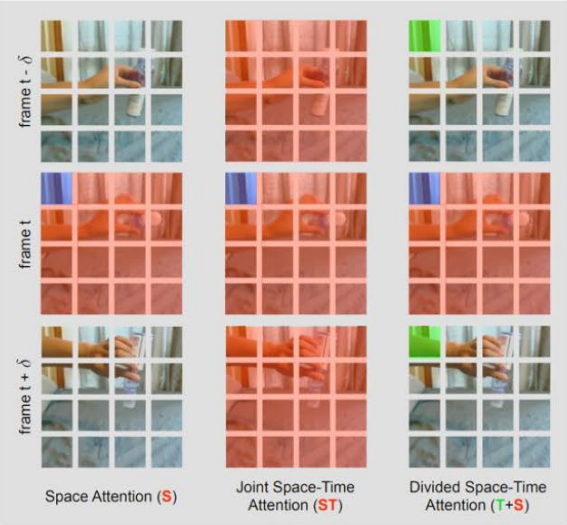
5 minutes at 24fps → 1,411,200 tokens
(~context limit of most modern LLMs)

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

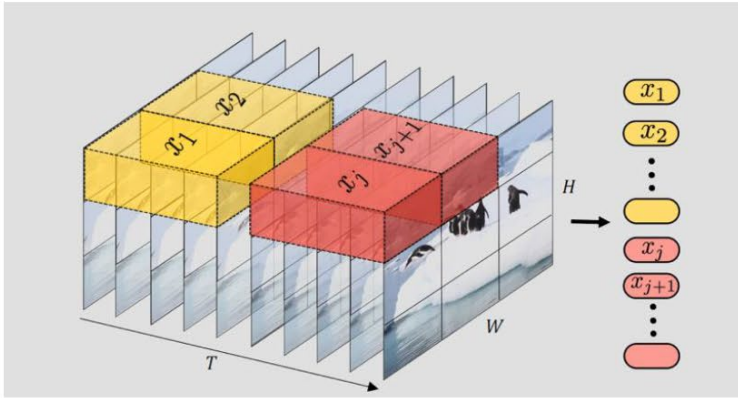
Two Broad Strategies for Video ViT

Modify attention operator



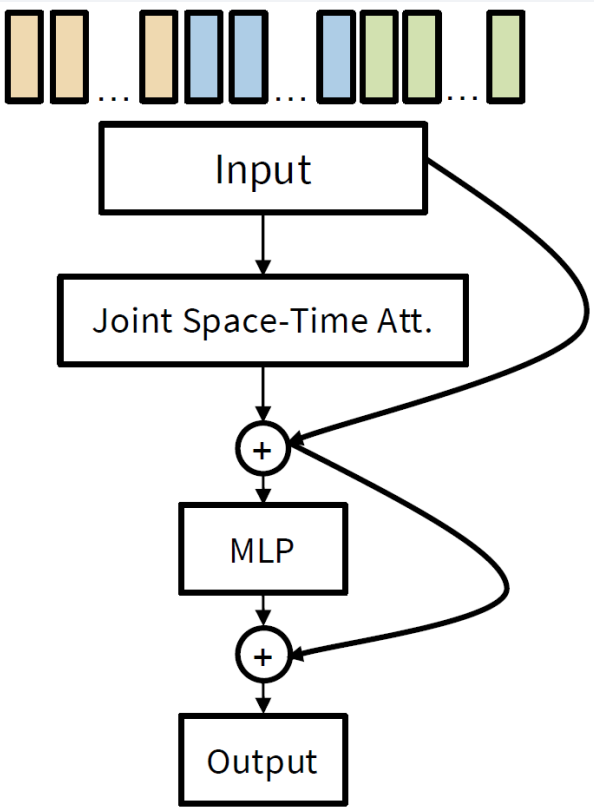
Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

Reduce the number of tokens

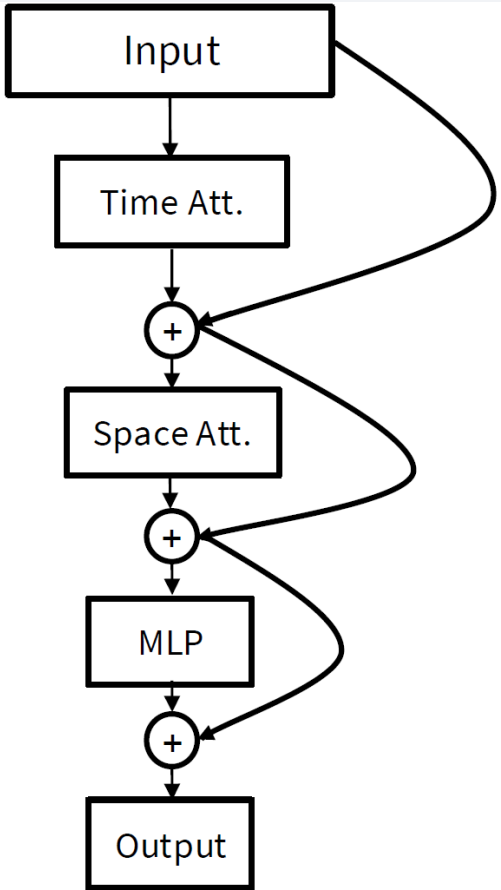


Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Fan et al, "Multiscale Vision Transformers", ICCV 2021
Li et al, "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection", CVPR 2022

#1-1. Joint vs Divided Space-Time Attention



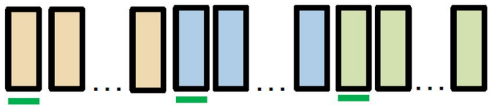
Key Idea: We can Divide Time and Space Attention Operators into 2 steps

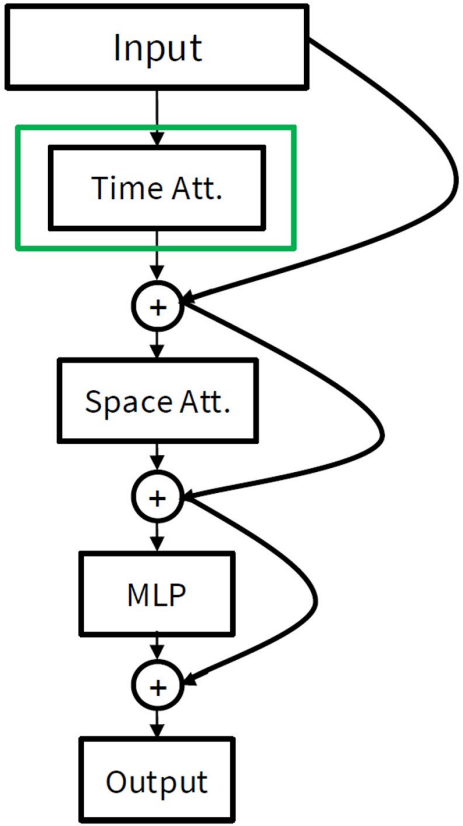


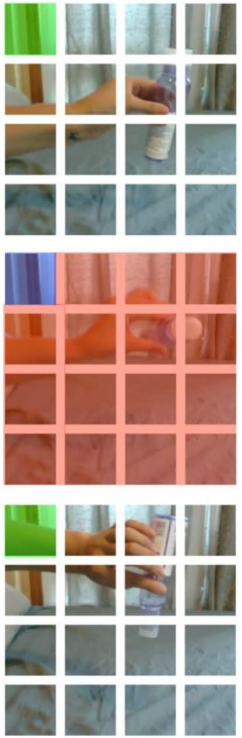
Figures modified from: Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021

#1-1. Divided Space-Time Attention

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





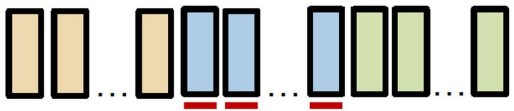


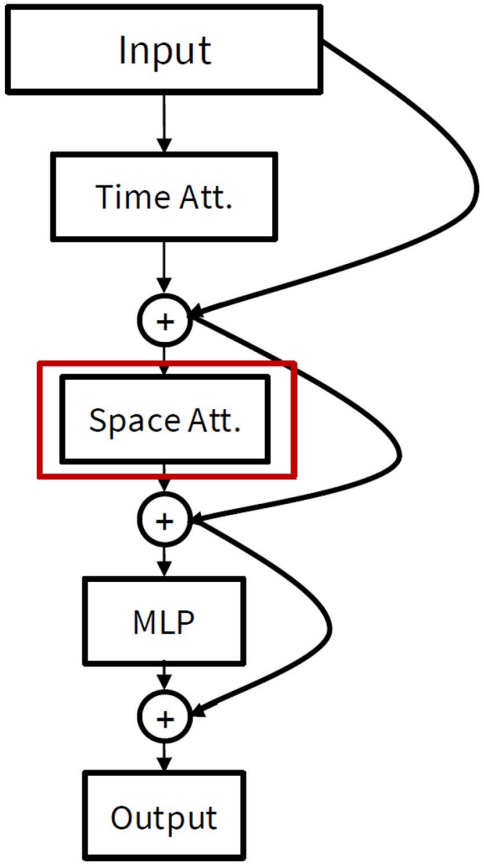
Divided Space-Time Attention (T+S)

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

#1-1. Divided Space-Time Attention

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





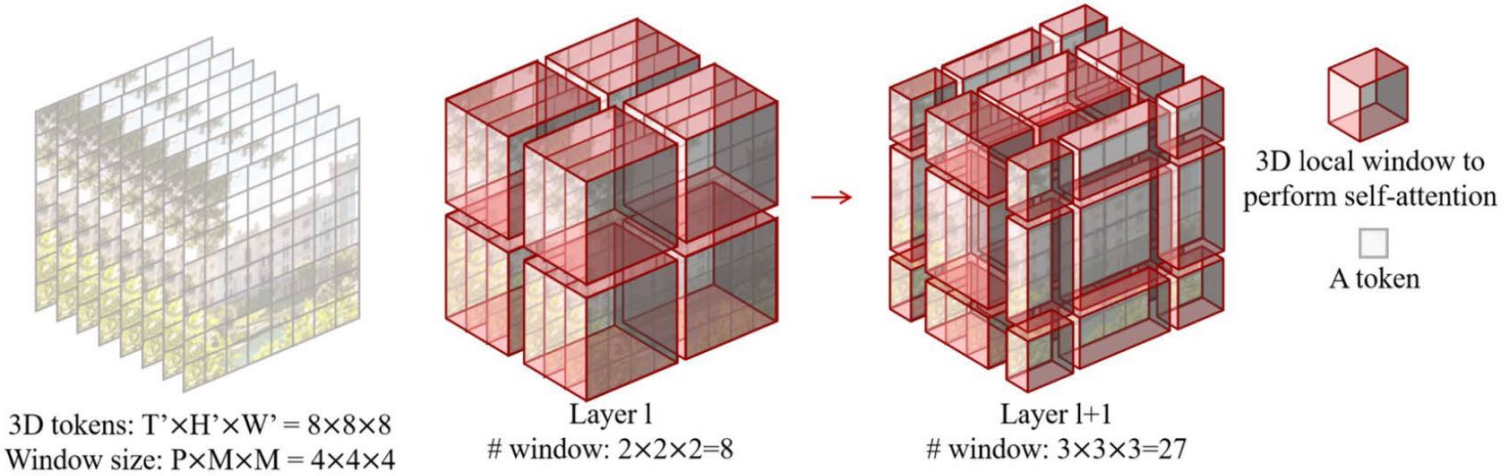
Divided Space-Time Attention (T+S)

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

#1-2. 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.



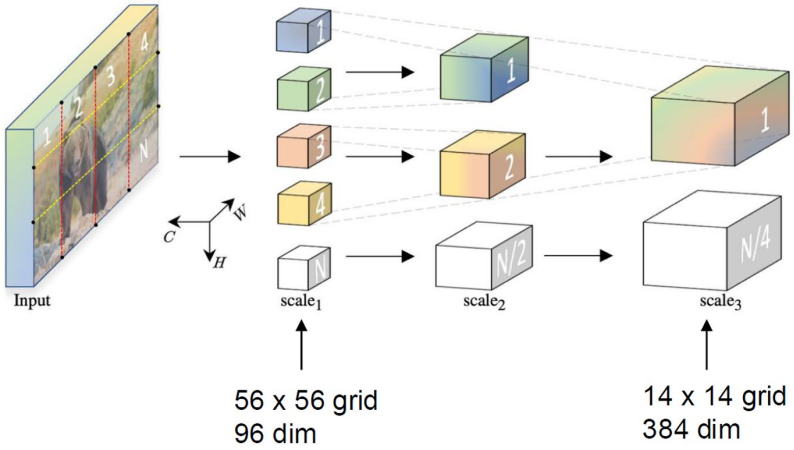
Figures from: Liu, Ze, et al. "Video Swin Transformer" CVPR 2022.

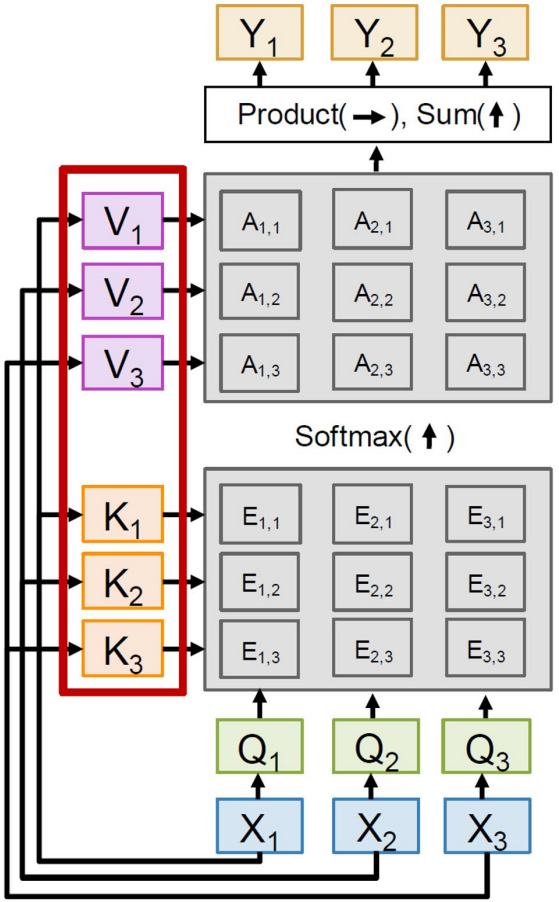
#1-3. Multiscale Vision Transformers

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

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)

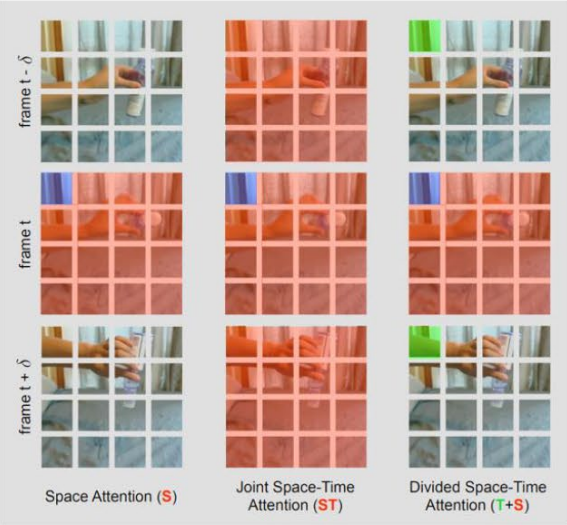




Progressively half spatial dimensions + double channels (same as ResNets)

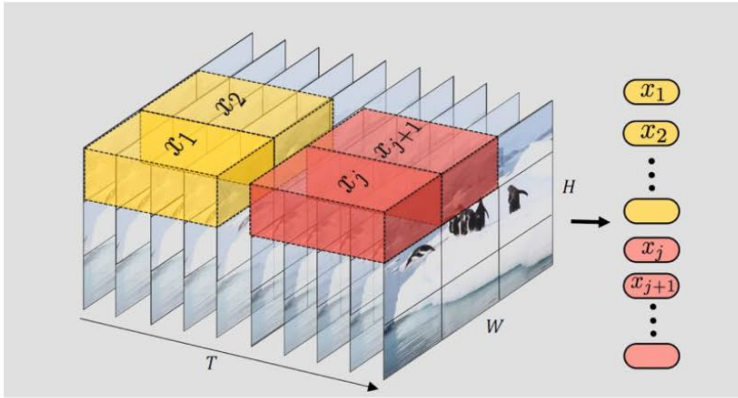
Two Broad Strategies for Video ViT

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 Video ViT

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.

Lots of other approaches to reduce token count

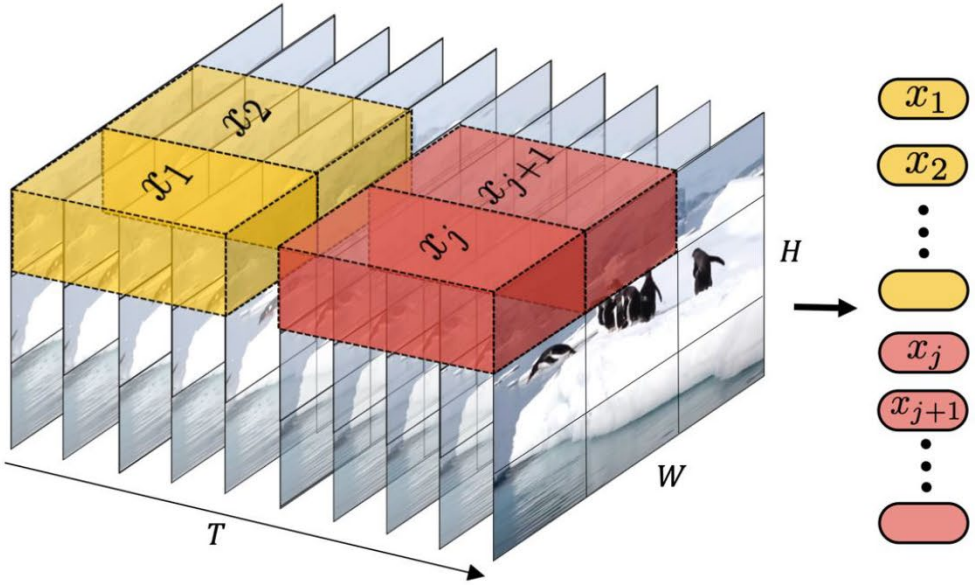


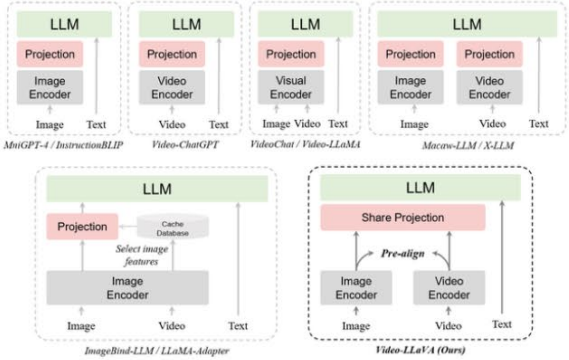
Figure from: Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021

Part 4

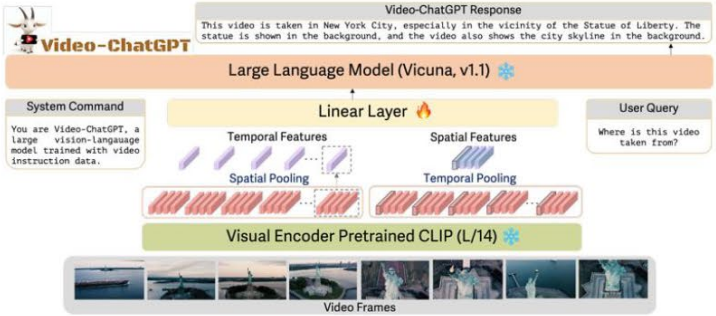
Video + Foundation Models

Generation and understanding in the modern era

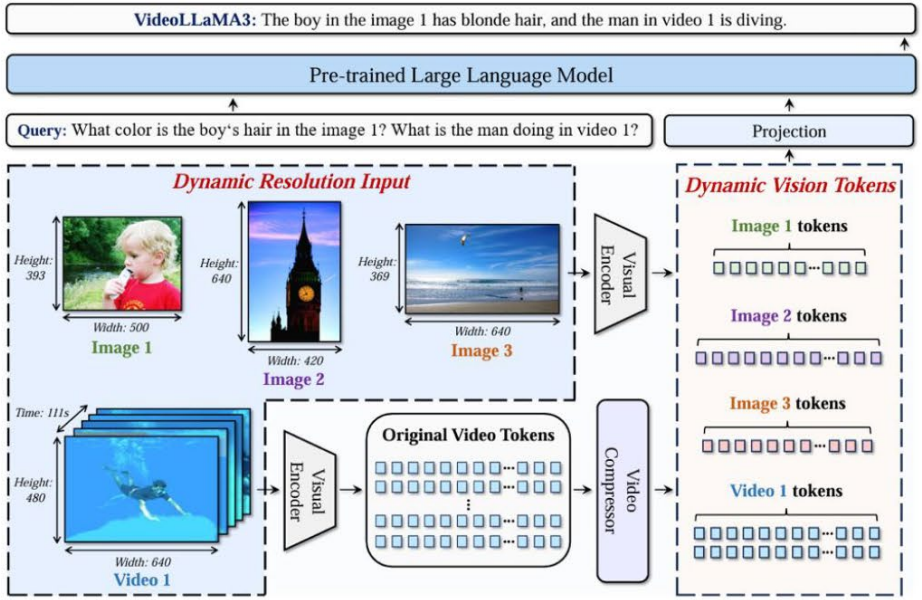
Video Understanding LLMs (Video LLMs)



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

Video Generation: From Diffusion to Sora

How Sora Works

Everything from Wk 10-11:

- Diffusion / Flow Matching
- + LDM (latent space)
- + DiT (Transformer backbone)
- + CFG (text guidance)

New for video:

- 2D patches → 3D spacetime patches
- Spatial attn → + temporal attn
- 2D VAE → 3D VAE
- (compress space AND time)

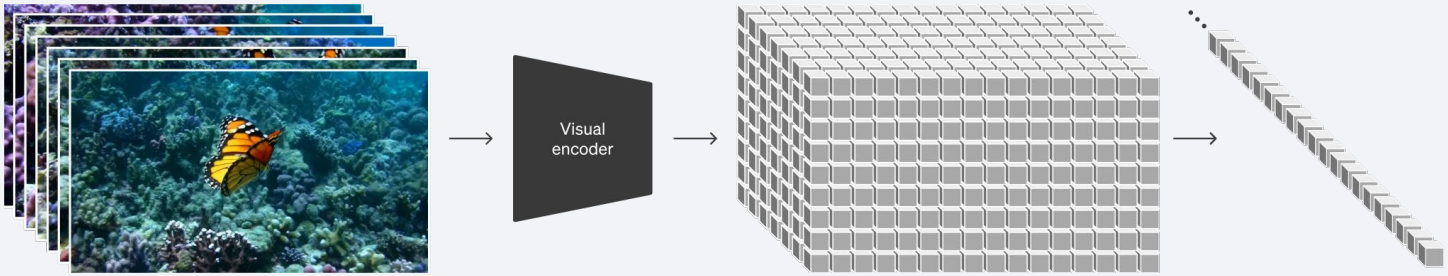
Scale & Challenges

Scale:

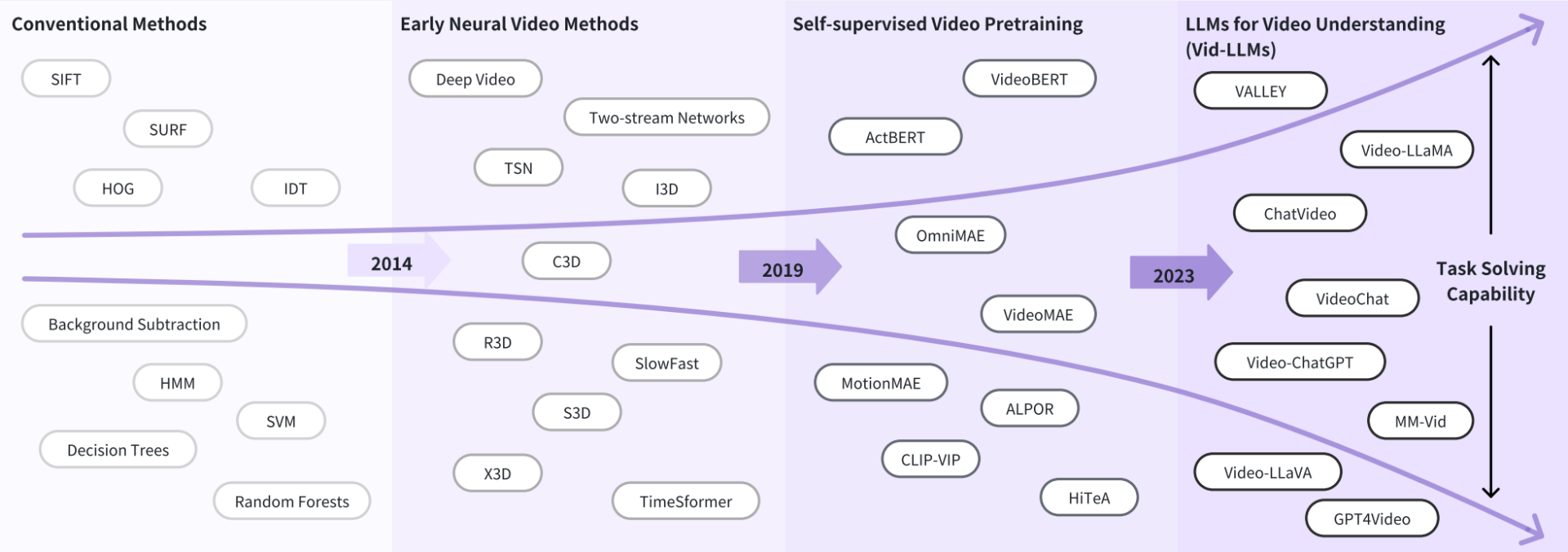
- 1 image: ~50K latent values
- 1 video: ~2M latent values
- 40× more compute

Coherence:

- Objects must persist
- Physics must be plausible
- Camera motion must be smooth



Video Understanding methods



Source: Tang et al., 2024

Summary & Resources

What We Covered

Part 1: Video = Image + Time
  Data structure, training on clips
Part 2: CNN Era
  Optical flow + Two-stream
  I3D (inflate)
Part 3: Transformer Era
  Modify attention operator
    - Joint / Divided attention
    - Video swin transformer
    - Multiscale ViT
  Reduce the number of tokens
Part 4: Foundation Model Era
  Sora = Diffusion + Video
  Video-LLM = VLM + frame sampling

Further Reading

CNN Era:
  Two-Stream: Simonyan & Zisserman, 2014
  C3D: Tran et al., 2015
  I3D: Carreira & Zisserman, 2017
  R(2+1)D: Tran et al., 2018
  Non-Local: Wang et al., 2018
  TSM: Lin et al., 2019
  SlowFast: Feichtenhofer et al., 2019
Transformer Era:
  TimeSformer: Bertasius et al., 2021
  ViViT: Arnab et al., 2021
  VideoMAE: Tong et al., 2022
Modern:
  Sora: OpenAI, 2024
  Survey: Tang et al., 2024

Next Week Week 15: Embodied AI & Robot vision

Acknowledgments

Some slides and figures in this course are adapted from or inspired by the following open resources.

- **Stanford CS231n: Deep Learning for Computer Vision (Spring 2026, Sprint 2025)**
  Fei-Fei Li, Ehsan Adeli, et al. | cs231n.stanford.edu
- **MIT 6.S184: Generative AI with Stochastic Differential Equations (2026)**
  Peter Holderrieth and Ezra Erives | https://diffusion.csail.mit.edu/
- **MIT 6.7960: Deep Learning (Fall 2024, Fall 2025)**
  Phillip Isola, Sara Beery, Jeremy Bernstein | ocw.mit.edu/courses/6-7960-deep-learning-fall-2024/, https://deeplearning6-7960.github.io/
- **CMU 16-824: Visual Learning and Recognition (Fall 2025)**
  Jun-Yan Zhu | visual-learning.cs.cmu.edu
- **Key papers: C3D, Two-Stream Networks, I3D, ViViT, Multiscale ViT, VideoMAE, …**