Learning from Videos & Video Generation

Naeemullah Khan
naeemullah.khan@kaust.edu.sa

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

LMH
Lady Margaret Hall

July 25, 2025

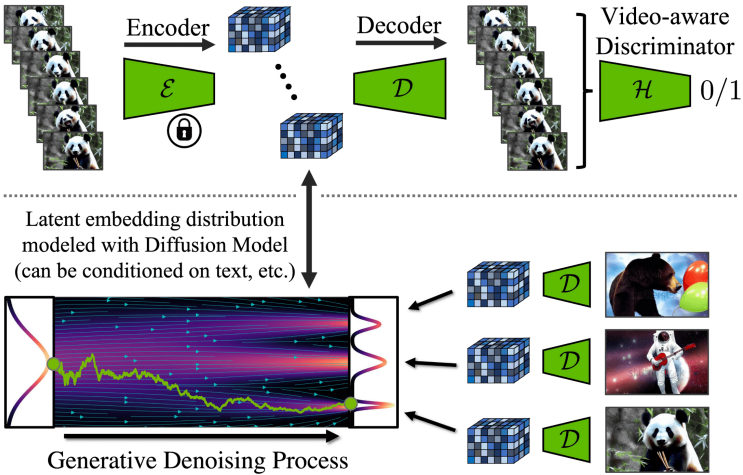

Encoder
Decoder
Video-aware
Discriminator
0/1
Latent embedding distribution
modeled with Diffusion Model
(can be conditioned on text, etc.)
Generative Denoising Process
(shown here for individual frames, see
Fig. 2 for video fine-tuning)

Table of Contents

Table of Contents (cont.)

Table of Contents (cont.)

Table of Contents (cont.)

Motivation

Why focus on video data?

- **Ubiquity of video data:** streaming, social media, autonomous systems
- **Need for temporal understanding:** beyond static images to dynamic events
- **Generative potential:** create, predict, and simulate future frames
- **Real-world impact:** surveillance, sports analytics, healthcare, AR/VR, content creation

⇒ Learning Outcomes

By the end of this session, you will be able to:

- Define video data and temporal concepts.
- Identify challenges in acquiring and processing video.
- Implement models from single-frame CNNs to spatio-temporal Transformers.
- Extract motion features and recognize actions.
- Apply self-supervised video representation learning.
- Connect vision and language using CLIP and zero-shot.
- Build generative models: VQ-GAN, StyleCLIP, Diffusion.
- Critically assess limitations and propose future research.

Video Learning & Generation: **Introduction**

What is Video Data?

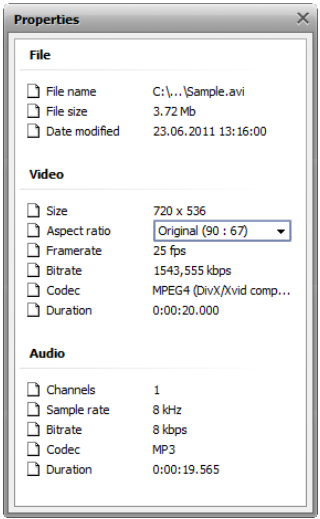
- **Video** is a sequence of images (frames) captured over time.
- Each frame is a static image, but the sequence captures motion and temporal dynamics.
- Videos can be continuous (e.g., surveillance footage) or discrete (e.g., video clips).
- Key components:
  - Frame rate: number of frames per second (FPS).
  - Resolution: dimensions of each frame.
  - Duration: total length of the video.

What is Video Data? (cont.)

Videos are all around us

Span an enormous space of spatial and temporal signals

What is Video Data? (cont.)



What is Video Data? (cont.)

RESOLUTION	AVERAGE DATA USAGE PER HOUR
480P (SD)	360 MB
720P (HD)	900 MB
1080P (FULL HD)	1.5 GB
4K (ULTRA HD)	7.2 GB

What is Video Data? (cont.)

A video is a **sequence** of images

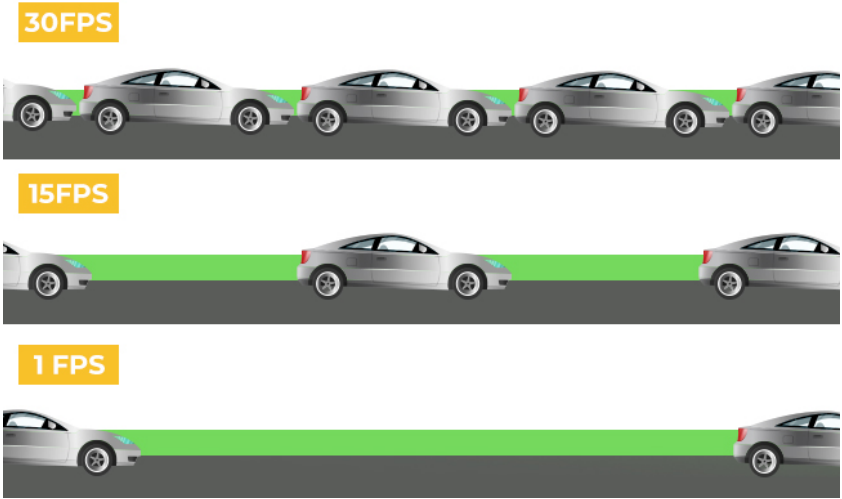
4D tensor: T x 3 x H x W

(or 3 x T x H x W)

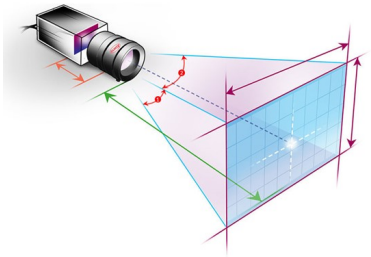
Camera Properties

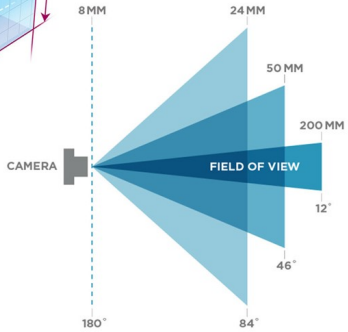
- Cameras capture video data with various properties that affect quality and usability.
- Key properties include:
  - **FPS (Frames per Second):** Determines the temporal resolution of the video; higher FPS captures smoother motion.
  - **Spatial Resolution:** Refers to the pixel dimensions of each frame (e.g., 1920x1080); affects image detail.
  - **Field of View (FOV):** The angular extent of the observable scene captured by the camera; wide FOV covers more area.
  - **Depth Sensors:** Devices like RGB-D cameras and LiDAR provide depth information, enabling 3D scene understanding.

Camera Properties (cont.)



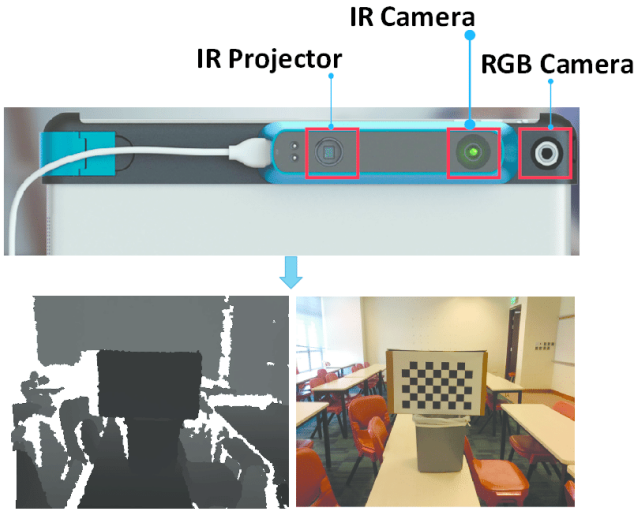
Camera Properties (cont.)





Camera Properties (cont.)

Camera Properties (cont.)



What Does Temporal Mean?

- **Single Frame:** Static content; no motion cues.
- **Short Clip:** Captures motion trajectories.
- **Temporal Window:** Enables extraction of velocity and acceleration.
- **Temporal Features:** Techniques such as frame differencing and optical flow capture motion information.

What Does Temporal Mean? (cont.)

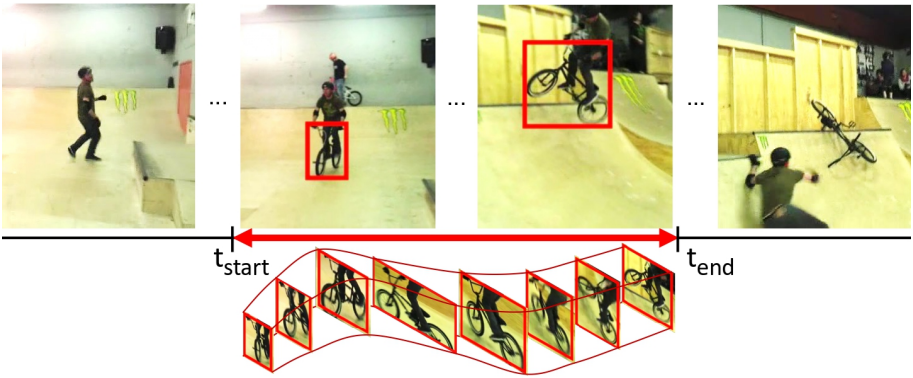


Illustration of temporal features in video: single frame, short clip, and motion extraction.

Video Learning & Generation: **Data Acquisition & Challenges**

Data Acquisition

- Video data can be acquired from various sources, including:
  - **Cameras:** Digital cameras, smartphones, and webcams.
  - **Drones:** Aerial footage for surveillance or mapping.
  - **Webcams:** Live streaming or recorded video.
  - **Public Datasets:** Pre-recorded videos available for research.
- Key considerations in data acquisition:
  - Quality of the recording device.
  - Environmental conditions (lighting, weather).
  - Ethical considerations (privacy, consent).

Data Acquisition (cont.)

- **Modern sources for video data:**
  - **User-generated content:** Platforms like YouTube and TikTok provide vast amounts of diverse video data.
  - **Surveillance & IoT:** Cameras embedded in city infrastructure, public spaces, and smart devices continuously capture video streams.
  - **Automotive:** Dashcams and sensors in autonomous vehicles generate large-scale driving and traffic video datasets.
- **Metadata and tags:** Videos are often accompanied by metadata (timestamps, GPS, tags) that facilitate search, retrieval, and organization.

Data Acquisition (cont.)

Popular Video Datasets:

- **UCF101:** 13,000+ video clips covering 101 human action categories.
- **Kinetics-700:** 650,000 video clips annotated with 700 action classes.
- **ActivityNet:** Large-scale dataset with untrimmed videos for temporal action localization.
- **Something-Something v2:** Focuses on fine-grained, object-centric actions.
- **Ego4D:** Egocentric videos capturing daily activities from a first-person perspective.

Data Acquisition (cont.)

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

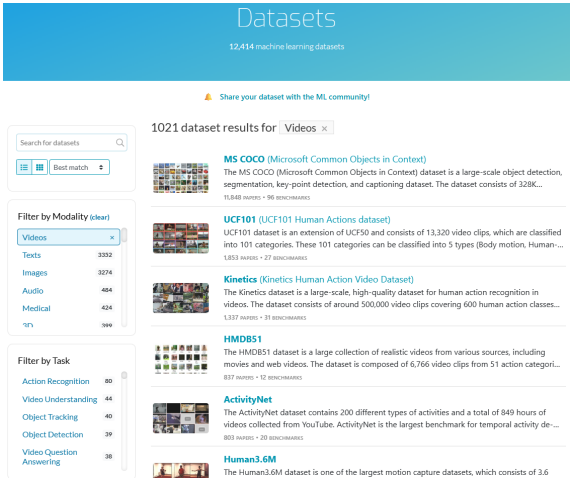
8M video clips, Machine-generated annotations from 3,862 classes

Sports-1M

YouTube videos

1,133,157 videos, 487 sports labels

Data Acquisition (cont.)



[Source: https://paperswithcode.com/datasets?mod=videos]

Challenges in Video Data

- **Issues Working with Videos:**
  - **High Storage:** Video files are large, requiring significant storage and bandwidth.
  - **Variable FPS & Resolution:** Differences in frame rates and resolutions increase preprocessing complexity.
  - **Annotation Cost:** Frame-level labeling is time-consuming and expensive.
  - **Data Imbalance:** Rare events or classes are often underrepresented in datasets.

Challenges in Video Data (cont.)

Input video:
T x 3 x H x W

Videos are ~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**

Challenges in Video Data (cont.)

Input video:
T x 3 x H x W

Videos are ~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**

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)

Challenges in Video Data (cont.)

- **Solutions for Video Data:**
  - **Compression & Sampling:** Techniques like keyframe extraction and frame sampling help reduce storage and computational requirements.
  - **Data Augmentation:** Methods such as frame dropout and temporal jittering increase data diversity and robustness.
  - **Synthetic Data:** Simulated environments (e.g., CARLA, AirSim) generate realistic video data for training and testing.
  - **Weak & Self-Supervision:** Leveraging unlabeled or partially labeled data reduces reliance on expensive manual annotations.

Video Learning & Generation: **Video Understanding Models**

Video Understanding Models

- Video understanding involves analyzing and interpreting video data to extract meaningful information.
- Key tasks include:
  - **Action recognition**: Identifying actions or activities in videos.
  - **Object detection**: Locating and classifying objects within video frames.
  - **Scene understanding**: Analyzing the context and environment in which actions occur.
  - **Temporal analysis**: Understanding how actions evolve over time.

Video Understanding Models (cont.)

Input video:
T x 3 x H x W

Swimming
Running
Jumping
Eating
Standing

Video Understanding Models (cont.)

- To understand videos, models need to look at both:
  - What's in each frame (spatial info).
  - How things move or change (temporal info).
- How do we do this?
  - Use CNNs to spot patterns in images.
  - Use RNNs or LSTMs to follow changes over time.
  - Use 3D CNNs to look at several frames together.
  - Use Transformers to connect information across the whole video.

Video Understanding Models (cont.)

Images: Recognize **objects**

Dog
Cat
Fish
Truck

Videos: Recognize **actions**

Swimming
Running
Jumping
Eating
Standing

Video Understanding Models (cont.)

- These models learn from lots of labeled videos.
- We check how well they work using accuracy, precision, recall, and F1-score.
- New research is making models faster, more reliable, and better at handling different types of videos.
- Where is this useful?
  - Security and surveillance.
  - Self-driving cars.
  - Sports analysis.
  - Healthcare monitoring.
  - Recommending videos online.

Training on Clips: Key Considerations

- **Clip Length Selection:** Typically, short clips of 16–64 frames are used to balance computational cost and temporal coverage.
- **Batch Size vs. Temporal Context:** Increasing clip length provides more temporal context but reduces the possible batch size due to memory constraints.
- **Pretraining for Transfer Learning:** Models are often pretrained on large-scale video datasets (e.g., Kinetics, Sports-1M) to improve performance on downstream tasks.

Training on Clips: Key Considerations (cont.)

Raw video: Long, high FPS

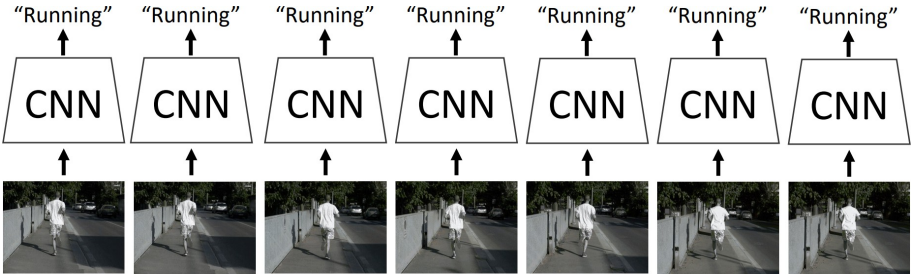
Training: Train model to classify short **clips** with low FPS

Testing: Run model on different clips, average predictions

Single-Frame CNN

- Treat each frame independently
- Use pretrained ImageNet backbones
- Ignores temporal context; serves as a baseline

Single-Frame CNN (cont.)

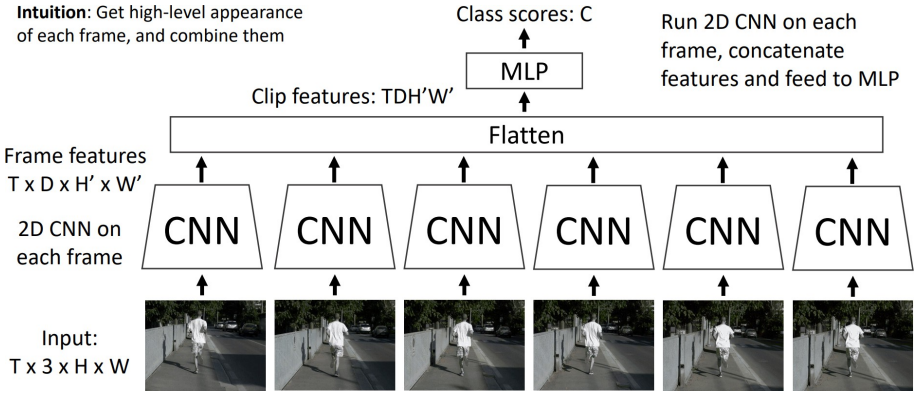


Late Fusion: Overview

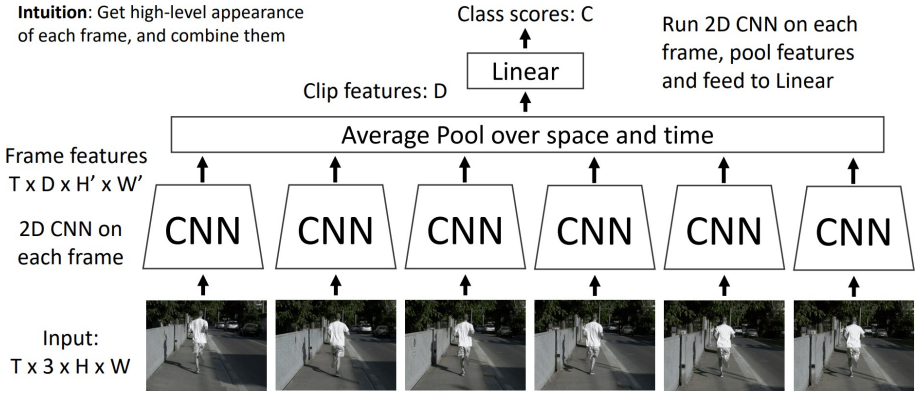
- **Process frames separately**, then combine their representations.
- **Fully Connected Fusion**: Concatenate frame features, then use fully connected (FC) layers.
- **Pooling Fusion**: Apply average or max pooling across time dimension.

Pros: Simple and easy to implement.
Cons: Weak temporal modeling; ignores complex temporal dependencies.

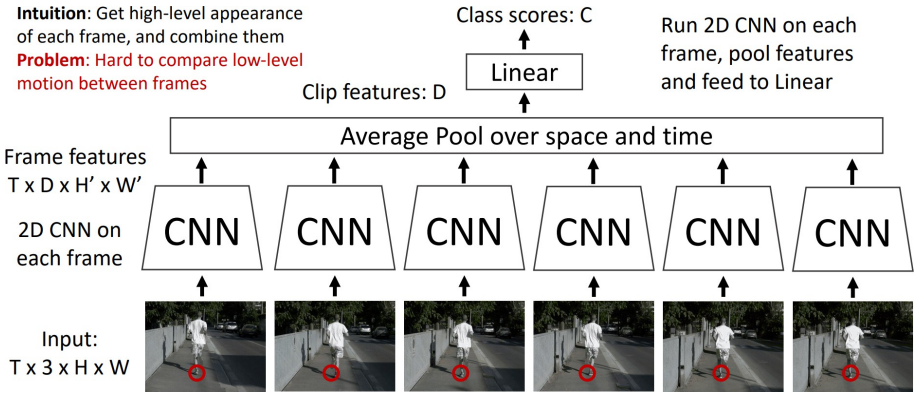
Late Fusion (with FC layers)



Late Fusion (with pooling)



Late Fusion (with pooling) (cont.)

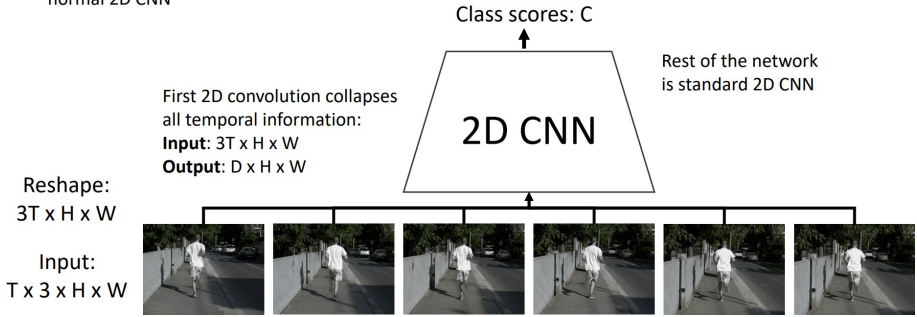


Early Fusion: Key Ideas

- **Stack frames as input channels to 2D CNN**
- Learns short-term motion patterns
- Limited receptive field along time

Early Fusion: Key Ideas (cont.)

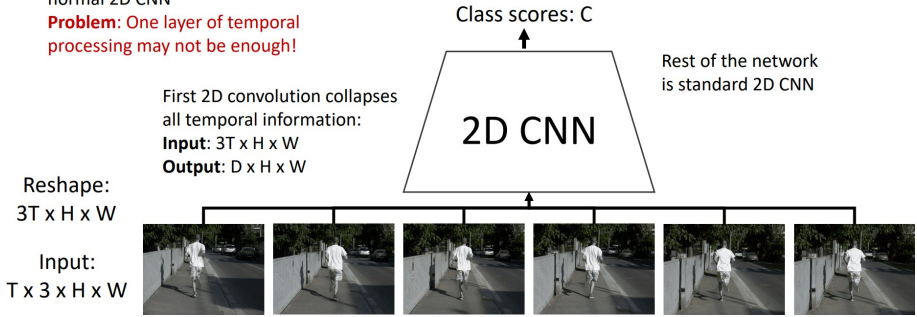
Intuition: Compare frames with very first conv layer, after that normal 2D CNN



Early Fusion: Key Ideas (cont.)

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

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



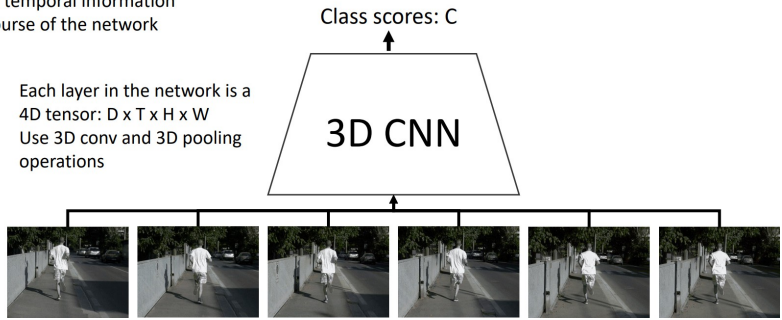
3D CNN: Key Ideas

- **3D Convolutional Networks**: Apply convolutional kernels across both spatial and temporal dimensions.
- **3D Kernels over Space-Time**: Filters operate on width, height, and time, enabling motion modeling.
- **Capture Local Spatio-Temporal Features**: Learn patterns that span multiple frames, such as movement or actions.
- **High Computational Cost**: Increased parameters and operations compared to 2D CNNs, leading to higher resource requirements.

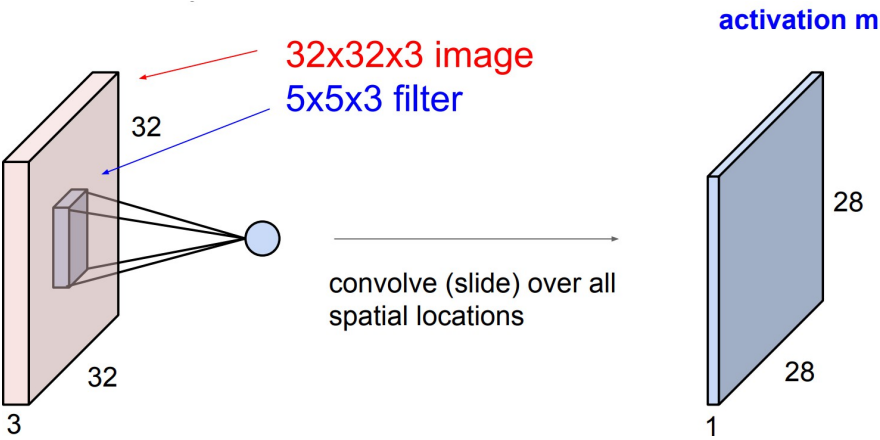
3D CNN: Key Ideas (cont.)

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

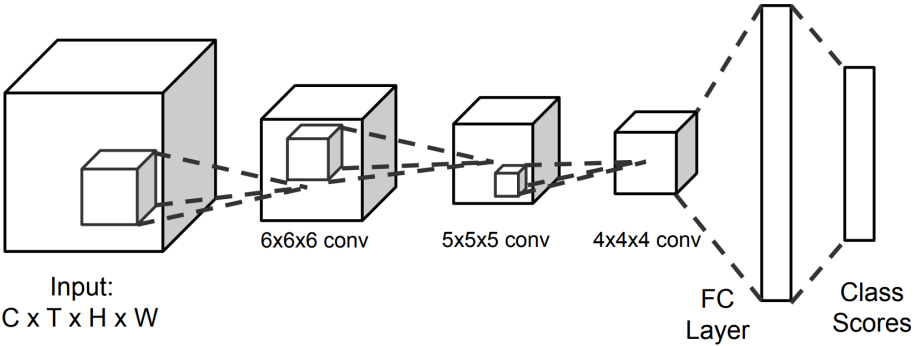
Each layer in the network is a 4D tensor: D x T x H x W
Use 3D conv and 3D pooling operations



3D CNN: Key Ideas (cont.)



3D CNN: Key Ideas (cont.)



3D CNN: Key Ideas (cont.)

- **3D CNNs** are powerful for video understanding, capturing both spatial and temporal features.
- They are particularly effective for tasks like action recognition, where understanding motion is crucial.
- However, they require significant computational resources and large datasets for training.
- **Applications**:
  - Action recognition in videos.
  - Gesture recognition.
  - Video classification.
  - Sports analysis.

C3D: Key Ideas

Introduced by Tran et al., ICCV 2015

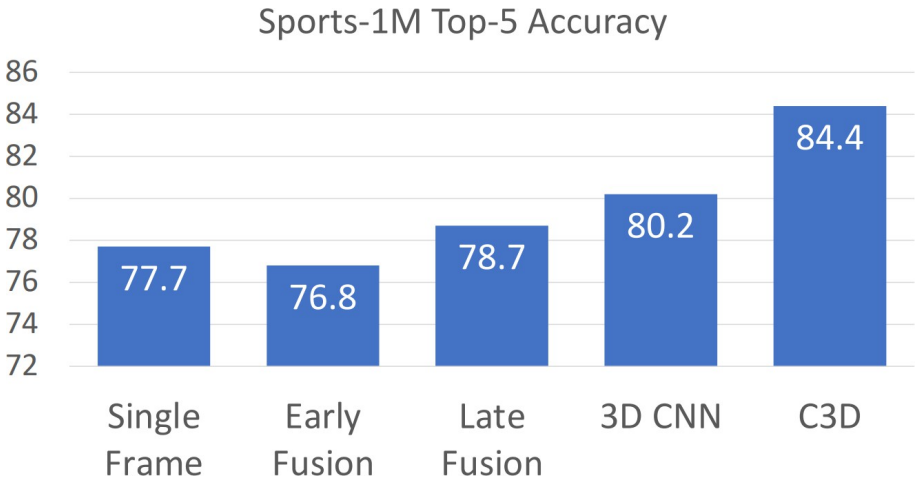
- Uses $3 \times 3 \times 3$ convolutions on 16-frame video clips.
- Demonstrated strong performance on the UCF101 action recognition benchmark.
- **3D Convolutional Networks**: Apply 3D convolutional kernels to capture spatio-temporal features.
- **Temporal Context**: C3D captures motion patterns by considering multiple frames simultaneously.
- **Pooling and Normalization**: Use pooling layers to reduce dimensionality and normalize features.
- **End-to-End Training**: Trained on large video datasets for tasks like action recognition.

C3D: Key Ideas (cont.)

- 3D CNN that uses all $3 \times 3 \times 3$ convolutions and $2 \times 2 \times 2$ pooling (except Pool1, which is $1 \times 2 \times 2$).
- Released model pretrained on Sports-1M: widely used as a video feature extractor.
- **Problem:** $3 \times 3 \times 3$ convolutions are very expensive!
  - AlexNet: 0.7 GFLOP
  - VGG-16: 13.6 GFLOP
  - C3D: 39.5 GFLOP (2.9× VGG!)

Layer	Size
Input	3 x 16 x 112 x 112
Conv1 (3x3x3)	64 x 16 x 112 x 112
Pool1 (1x2x2)	64 x 16 x 56 x 56
Conv2 (3x3x3)	128 x 16 x 56 x 56
Pool2 (2x2x2)	128 x 8 x 28 x 28
Conv3a (3x3x3)	256 x 8 x 28 x 28
Conv3b (3x3x3)	256 x 8 x 28 x 28
Pool3 (2x2x2)	256 x 4 x 14 x 14
Conv4a (3x3x3)	512 x 4 x 14 x 14
Conv4b (3x3x3)	512 x 4 x 14 x 14
Pool4 (2x2x2)	512 x 2 x 7 x 7
Conv5a (3x3x3)	512 x 2 x 7 x 7
Conv5b (3x3x3)	512 x 2 x 7 x 7
Pool5	512 x 1 x 3 x 3
FC6	4096
FC7	4096
FC8	C

C3D: Key Ideas (cont.)



Video Learning & Generation: **Motion & Temporal Dynamics**

Recognising Actions from Motion

- Motion as a primary cue for recognising actions in videos.
- Importance of separating the analysis of appearance and motion.
- Motion patterns often provide more discriminative information than static appearance.

Measuring Motion: Optical Flow

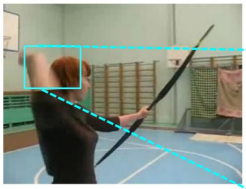
- Computes a dense motion field between consecutive frames.
- Popular algorithms:
  - Farneback
  - TV-L1
- Optical flow is commonly used as input to Two-Stream Networks for action recognition.

Optical Flow: Key Concepts

- **Dense Motion Field**: Represents motion vectors for every pixel between two frames.
- **Temporal Coherence**: Assumes that motion is smooth and continuous over time.
- **Applications**: Used in action recognition, video segmentation, and object tracking.

Optical Flow: Key Concepts (cont.)

Image at frame t



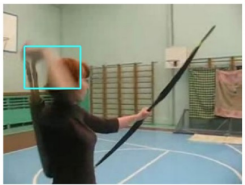
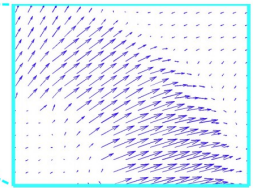


Image at frame t+1

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

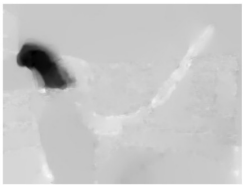


Tells where each pixel will move in the next frame:

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

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

Horizontal flow dx

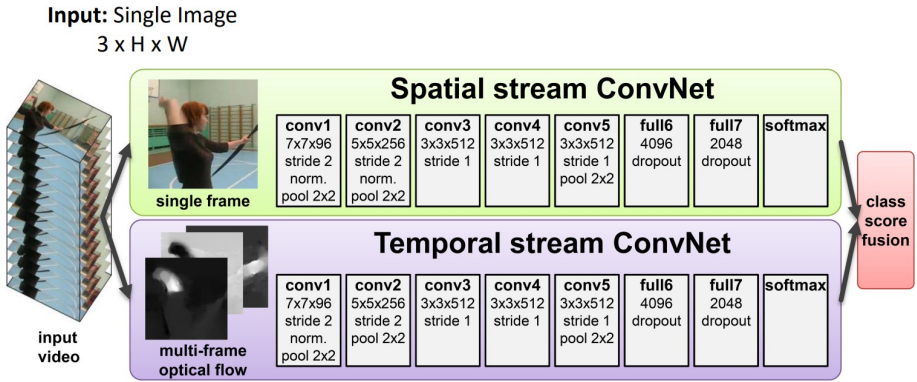


Vertical Flow dy

Separating Motion and Appearance

- **Two-Stream Network:** Processes both RGB (appearance) and optical flow (motion) streams.
- Streams can be fused at various depths:
  - *Early fusion:* Combine features at initial layers.
  - *Late fusion:* Combine predictions or high-level features.
- Improves robustness to background and enhances motion understanding.

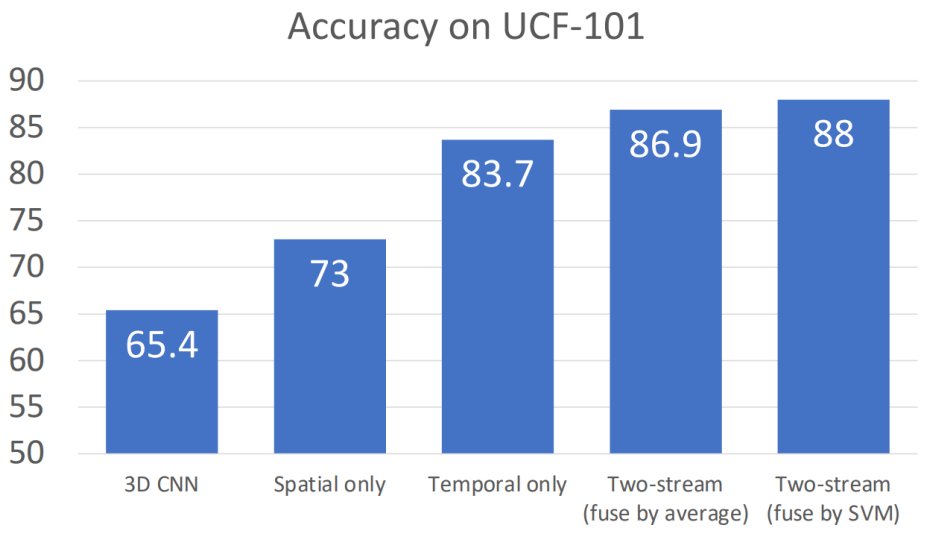
Two-Stream Network Architecture



Two-Stream Network Architecture (cont.)

- **RGB Stream:** Standard CNN (e.g., ResNet) for appearance features.
- **Optical Flow Stream:** Separate CNN for motion features.
- **Fusion:** Can be done at different stages:
  - Early: Concatenate features from both streams.
  - Late: Combine predictions from both streams.

Two-Stream Network Architecture (cont.)



Long-Term Temporal Structure

Understanding complex actions in videos often requires analyzing long-term temporal structure, rather than relying solely on short clips. This involves capturing dependencies and patterns that span extended periods, which are crucial for recognizing actions composed of multiple stages or events.

- **Temporal Segmentation:** Divides a video into meaningful segments based on changes in activity.
- **Temporal Action Proposals:** Identifies candidate time intervals where actions may occur.

These techniques enable models to focus on relevant portions of the video and improve the recognition of complex, long-duration actions.

Modelling Long-Term Structure

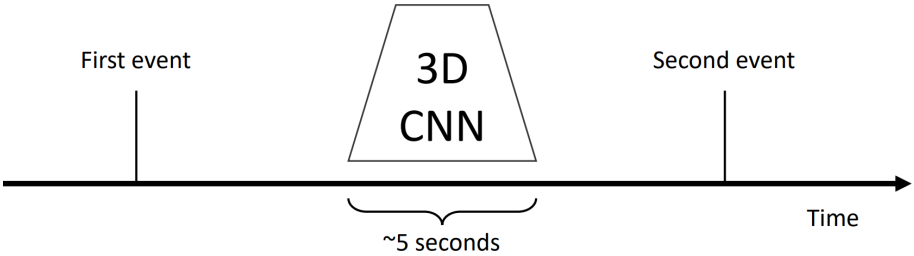
Several approaches have been developed to effectively model long-term temporal dependencies in videos:

- **Hierarchical RNNs and LSTMs:** Stack multiple recurrent layers to capture both short-term and long-term dependencies at different temporal scales.
- **Temporal Segment Networks (TSN):** Sample frames or segments sparsely across the entire video and aggregate their representations to model long-range structure.
- **Memory Networks:** Utilize external memory modules to store and retrieve information across hundreds of frames, enabling reasoning over extended temporal contexts.

These methods help in recognizing actions that unfold over long durations and involve multiple stages.

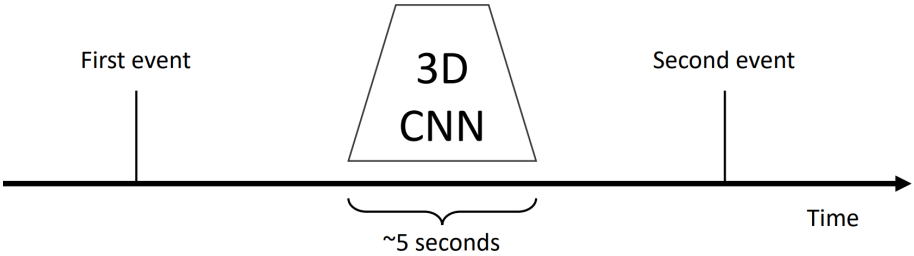
Modelling Long-Term Structure

- So far, all our temporal CNNs only model local motion between frames in very short clips of approximately 2–5 seconds.
- What about long-term structure?



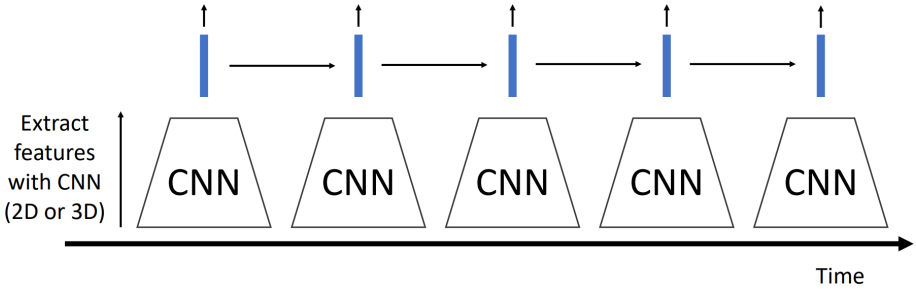
Modelling Long-Term Structure

- So far, all our temporal CNNs only model local motion between frames in very short clips of approximately 2–5 seconds.
- What about long-term structure?
- **We know how to handle sequences!**
- How about **recurrent networks**?

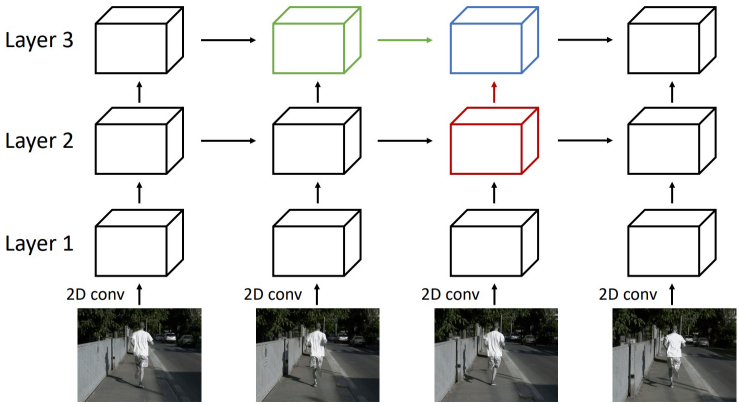


Modelling Long-Term Structure

- Process local features using recurrent network (e.g. LSTM)
- Many to many: one output per video frame



Recurrent Convolutional Network



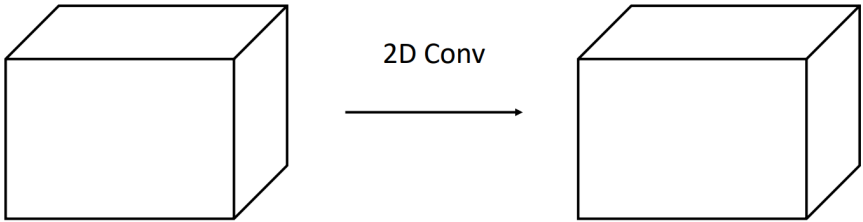
Entire network uses 2D feature maps: C x H x W

Each depends on two inputs:
1. Same layer, previous timestep
2. Prev layer, same timestep

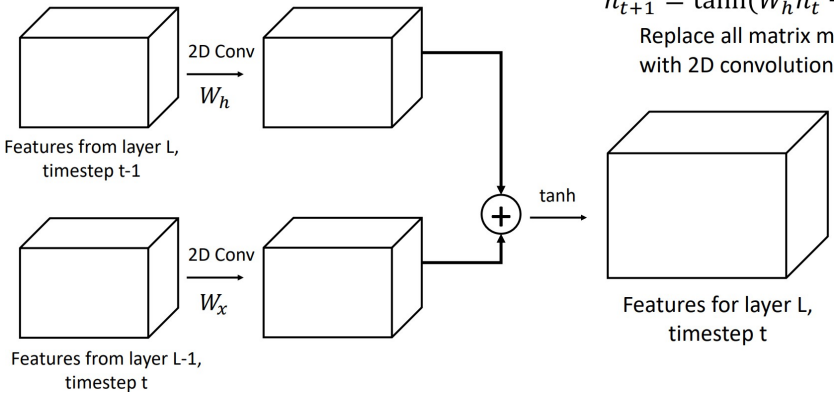
Use different weights at each layer, share weights across time

Ballas et al, "Delving Deeper into Convolutional Networks for Learning Video Representations", ICLR 2016

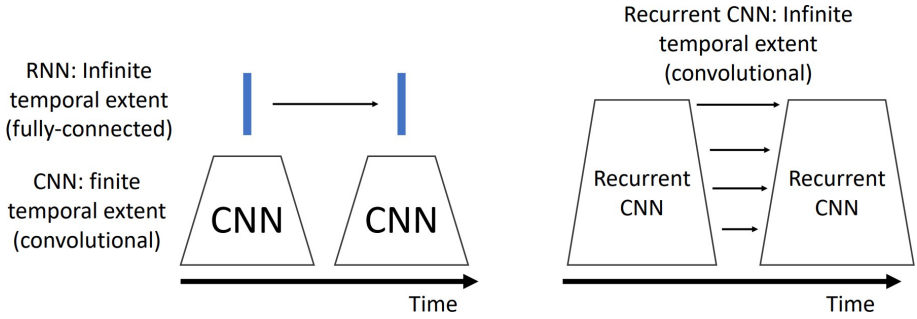
Normal 2D CNN:



Recurrent Convolutional Network (cont.)

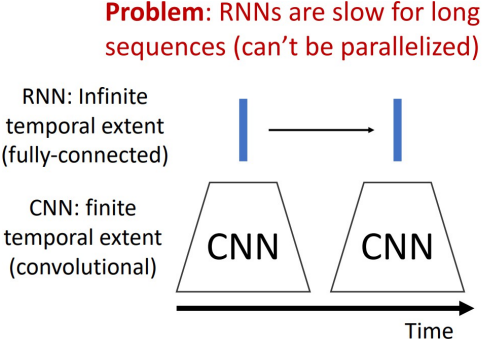


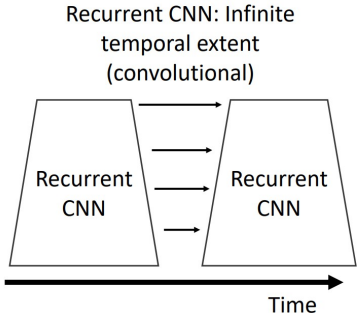
Modeling long-term temporal structure



Modeling long-term temporal structure (cont.)

Problem: RNNs are slow for long sequences (can't be parallelized)





Spatio-Temporal Self-Attention

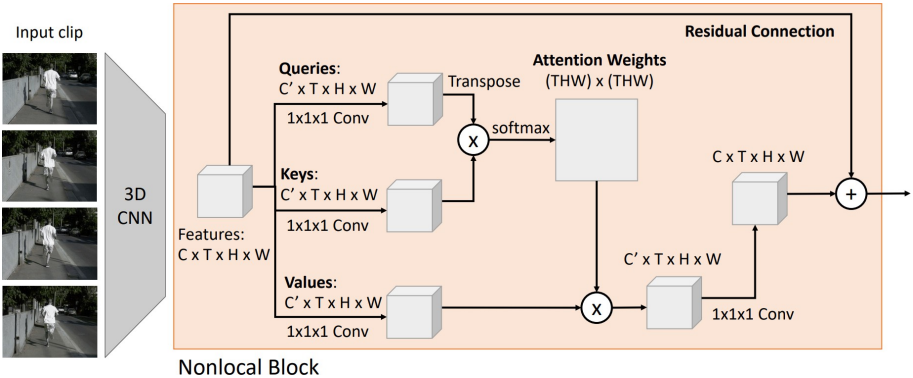
Spatio-Temporal Self-Attention extends the Transformer architecture to video data by modeling dependencies across both spatial and temporal dimensions.

- **Transformers applied to video:** Video frames are treated as sequences, enabling the use of self-attention to capture relationships across space and time.
- **Factorized Attention (e.g., TimeSformer):** Attention is applied separately along spatial and temporal axes, reducing computational cost compared to full spatio-temporal attention.
- **Benefits:** Enables modeling of global context and long-range dependencies in videos.
- **Costs:** Self-attention has quadratic complexity with respect to sequence length, making it computationally expensive for long videos.

Spatio-Temporal Self-Attention (cont.)

- **Self-Attention Mechanism:** Captures long-range dependencies in both spatial and temporal dimensions.
- **Multi-Head Attention:** Allows the model to focus on different parts of the input sequence simultaneously.

Spatio-Temporal Self-Attention (Nonlocal Block)



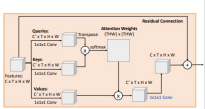
Nonlocal Block

Spatio-Temporal Self-Attention (Nonlocal Block) (cont

Input clip

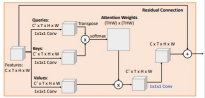
3D CNN

We can add nonlocal blocks into existing 3D CNN architectures.
But what is the best 3D CNN architecture?



Nonlocal Block

3D CNN



Nonlocal Block

3D CNN

Running

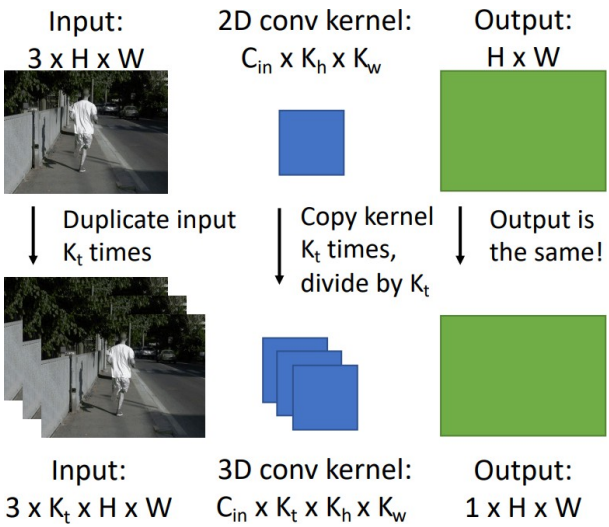
Inflating 2D Networks to 3D (I3D)

- 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.

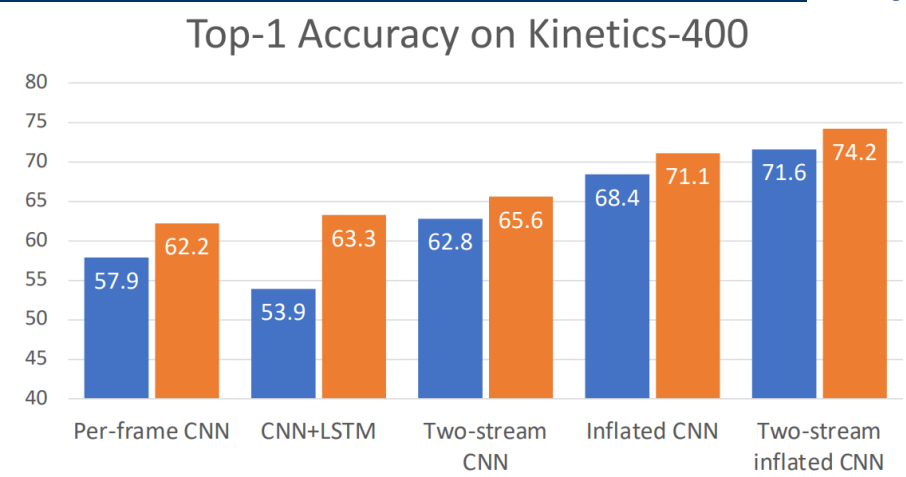
Inflating 2D Networks to 3D (I3D) (cont.)

- **Introduced by Carreira and Zisserman, CVPR 2017**
- **Key Ideas:**
  - **Inflation of 2D Filters**: Converts 2D convolutional filters into 3D by replicating them across time.
  - **Temporal Modeling**: Captures motion information effectively by leveraging pre-trained 2D networks.
  - **End-to-End Training**: Trained on large video datasets, allowing transfer learning to new tasks.
  - **Performance Boost**: Significantly improves action recognition accuracy compared to previous methods.

Inflating 2D Networks to 3D (I3D) (cont.)



Inflating 2D Networks to 3D (I3D) (cont.)



set", CVPR 2017

All using Inception CNN

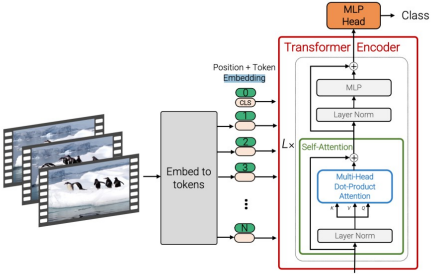
Vision Transformers for Video

Vision Transformers (ViTs) adapt the Transformer architecture for video data by treating video frames as sequences of patches, enabling the capture of long-range dependencies across both spatial and temporal dimensions.

- **Patch Embedding:** Each video frame is divided into patches, which are then linearly embedded into a sequence.
- **Temporal Encoding:** Positional encodings are added to capture temporal information.
- **Self-Attention Mechanism:** Allows the model to focus on different parts of the video sequence simultaneously.
- **Benefits:** Captures global context and long-range dependencies effectively.
- **Costs:** Computationally expensive due to quadratic complexity with respect to sequence length.

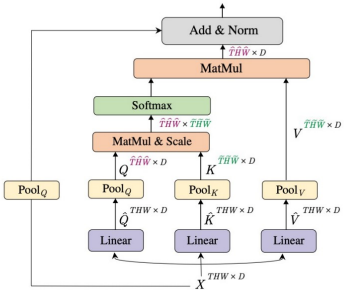
Vision Transformers for Video (cont.)

Factorized attention: Attend over space / time



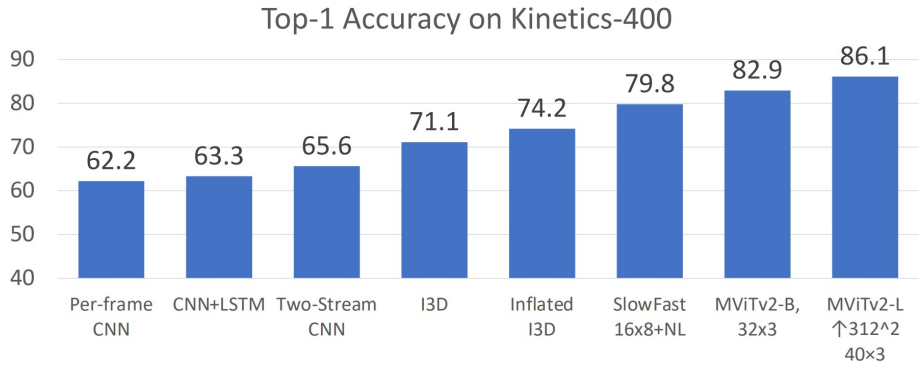
Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021
Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Neimark et al, "Video Transformer Network", ICCV 2021

Pooling module: Reduce number of tokens



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

Vision Transformers for Video (cont.)



Advanced Video Understanding

Video Understanding Models

- Video understanding models are designed to capture both spatial and temporal information in videos.
- They often combine convolutional neural networks (CNNs) for spatial feature extraction with recurrent neural networks (RNNs) or attention mechanisms for temporal dynamics.
- These models can be used for tasks such as action recognition, video captioning, and video retrieval.

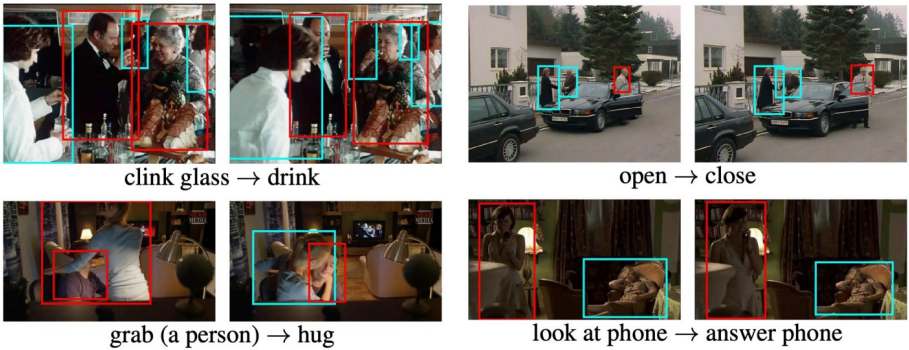
Spatio-Temporal Detection

Problem Statement:
Given a long untrimmed video, the goal is to detect all people in both space and time, and classify the activities they are performing.

- **Input:** Untrimmed video sequence.
- **Output:** For each person, a spatio-temporal tube (bounding boxes across frames) and an activity label.

Example: AVA Dataset

The AVA (Atomic Visual Actions) dataset provides annotated videos with bounding boxes and action labels for each person in every frame.



- **Annotations:** Spatial (bounding boxes) and temporal (frame-level) localization of actions.

Spatio-Temporal Detection (cont.)

Spatio-temporal detection involves identifying and localizing objects or events in both space and time within video sequences. This is crucial for tasks such as action recognition, event detection, and video surveillance.

- **Spatio-Temporal Features:** These features capture both the spatial layout of objects and their temporal dynamics.
- **3D Convolutional Networks (3D CNNs):** Extend traditional CNNs by adding a temporal dimension, allowing them to learn spatio-temporal patterns directly from video data.
- **Challenges:** High computational cost, data sparsity, and the need for large annotated datasets.
- **Applications:** Surveillance, human-computer interaction, sports analytics.

Ego4D: New Large-Scale Video Dataset

Ego4D is a large-scale dataset designed to advance the understanding of egocentric (first-person) video. It consists of over 3,670 hours of egocentric video footage captured using head-mounted cameras.

- **Long Videos:** Each video ranges from *1 to 10 hours* in length.
- **Diversity:** Data collected by *14 teams* across *9 countries*, *74 locations*, involving *931 unique camera wearers*.
- **Natural-Language Narrations:** Contains *3.85 million* sentences of narrations.
- **Rich Annotations:** Supports *5 different tasks*:
  - Episodic Memory
  - Hands and Objects
  - Audio-Video Diarization
  - Social Interactions
  - Forecasting

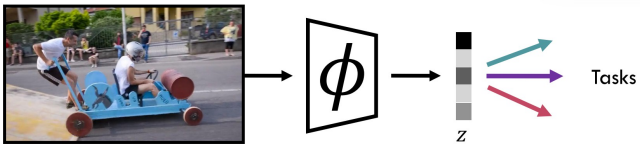
Ego4D: New Large-Scale Video Dataset (cont.)

Key Features of Ego4D:

- **First-Person Perspective:** Captures videos from the viewpoint of the person wearing the camera, providing a unique perspective on human activities.
- **Large Scale:** 3,670 hours of video data, making it one of the largest datasets for egocentric video understanding.
- **Diverse Activities and Participants:** Wide range of activities and a large, varied set of camera wearers, reflecting real-world scenarios.
- **Comprehensive Annotations:** Natural-language narrations and support for multiple tasks enable comprehensive model training and evaluation.

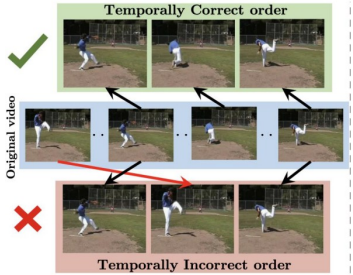
Ego4D: New Large-Scale Video Dataset (cont.)

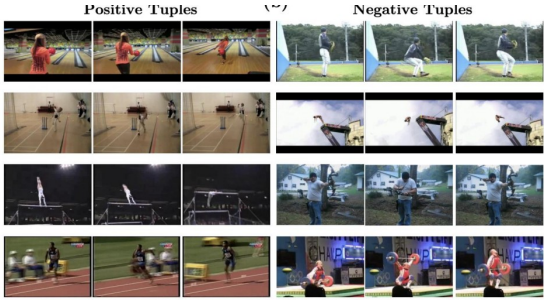
- **Temporal Order:** Learning to predict or verify the correct temporal sequence of video frames as a self-supervised task.
- **Cycle Consistency:** Enforcing that transformations applied to video frames or clips can be reversed, ensuring consistency in learned representations.
- **Video Speedup:** Training models to recognize or reconstruct videos at different playback speeds, encouraging temporal understanding.
- **Video Colorization:** Using the task of colorizing grayscale video frames as a self-supervised signal for learning spatiotemporal features.



Training data: Shuffled video frames, original video frames

Pretext task: Predict if the frames are in the correct temporal order (binary classification task).



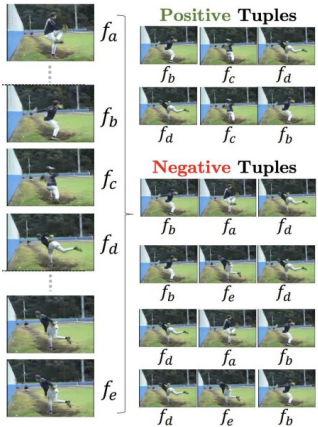


Frame Reordering (cont.)

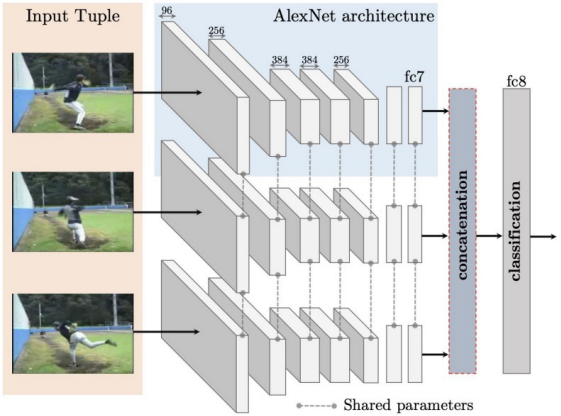
Objective: Learn temporal dependencies and motion patterns in videos.

- Given a sequence of video frames, randomly shuffle the order of frames.
- The model receives either the original or shuffled sequence.
- The objective is to classify whether the input sequence is temporally correct or not.
- This encourages the model to learn temporal dependencies and motion patterns in videos.

Frame Reordering (cont.)



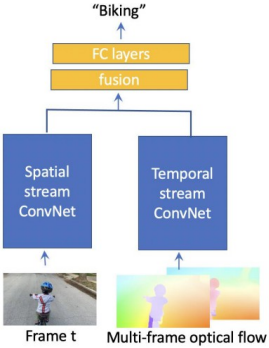
Generating positive and negative examples



Triplet Siamese network for sequence verification

Frame Reordering (cont.)

Transfer Learning: Fine-tune spatial stream for video classification.



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] **Benefits:**

Frame Reordering (cont.)

- No need for labeled data, as the task is self-supervised.
- Helps the model learn temporal dynamics and motion patterns in videos.
- Can be used as a pre-training step before fine-tuning on specific video tasks.

Learning correspondence

Training data: Video frames, corresponding optical flow maps

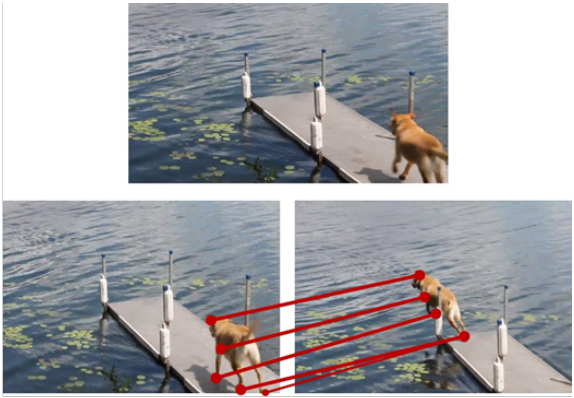
Pretext task: Predict the optical flow between pairs of video frames.

Learning Correspondence:

- Match patches across frames
- Applications: tracking, correspondence learning

Learning correspondence (cont.)

Ultimate Goal: Correspondence



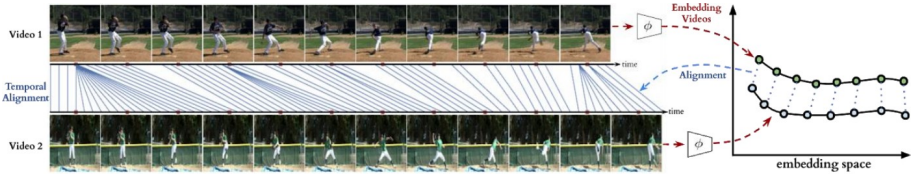
[Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019]

Temporal cycle consistency

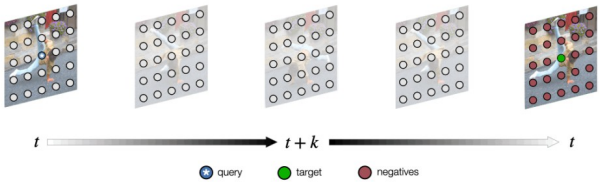
Temporal Cycle Consistency is a self-supervised learning approach that leverages the temporal structure of videos to learn representations by ensuring that transformations applied to video frames can be reversed.

- **Cycle Consistency:** The model learns to predict the optical flow between pairs of video frames, ensuring that the transformation can be reversed.
- **Applications:** Useful for tasks like tracking and correspondence learning.
- **Goal:** To learn robust representations that capture the temporal dynamics of video data.

Temporal cycle consistency (cont.)



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



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

Learning the Speediness in Videos

Learning Speediness is a self-supervised learning approach that focuses on understanding the temporal dynamics of videos by learning to recognize or reconstruct videos at different playback speeds.

- **Speed Variation:** The model learns to predict or reconstruct video frames at various speeds, enhancing its understanding of temporal relationships.
- **Applications:** Useful for tasks like action recognition, video summarization, and temporal segmentation.
- **Goal:** To learn robust representations that capture the dynamics of video data across different temporal scales.

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

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

Vision-Language Alignment

Vision-Language Alignment

- Vision-language alignment is crucial for tasks like image captioning, visual question answering, and video understanding.
- It involves aligning visual features with textual descriptions to enable models to understand and generate language based on visual content.
- Techniques include joint embedding spaces, cross-modal attention mechanisms, and pre-training on large datasets.

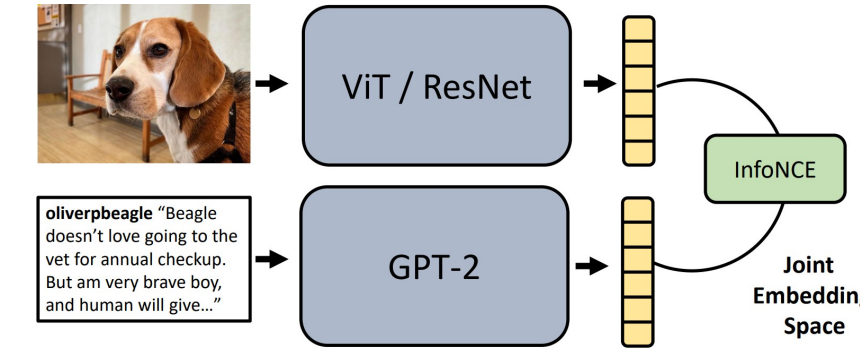
CLIP – Connecting Images and Text

CLIP (Contrastive Language-Image Pretraining) is a model that learns to connect images and text by training on a large dataset of image-text pairs.

- **Training Objective:** CLIP uses a contrastive loss to align image and text representations in a shared embedding space.
- **Zero-Shot Learning:** CLIP can perform zero-shot classification by comparing the similarity between image embeddings and text embeddings of class labels.
- **Applications:** Useful for tasks like image classification, object detection, and visual question answering without requiring task-specific training.

CLIP – Connecting Images and Text (cont.)

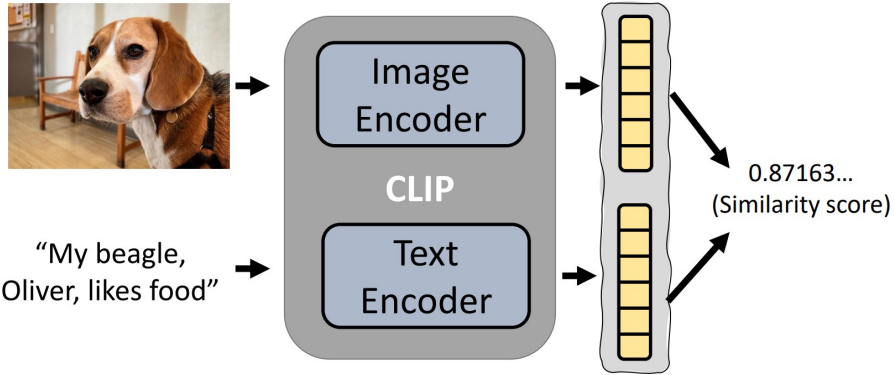
Architecture: CLIP consists of an image encoder (e.g., a Vision Transformer or CNN) and a text encoder (e.g., a Transformer), both trained jointly to produce embeddings in a shared space.



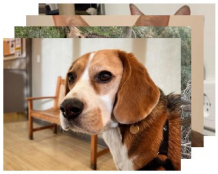
[Radford et al., Learning Transferable Visual Models From Natural Language Supervision, ICML 2021]

CLIP – Connecting Images and Text (cont.)

- **C**ontrastive **L**anguage **I**mage **P**retraining

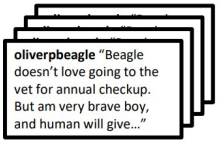


CLIP – Connecting Images and Text (cont.)



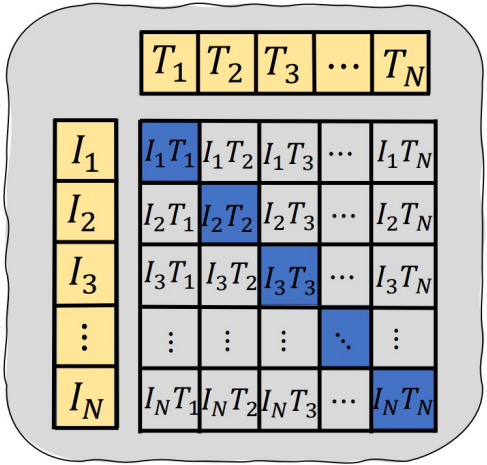
$I_{1...N}$

encode images



$T_{1...N}$

encode texts



joint embedding space

CLIP – Connecting Images and Text (cont.)

( , **oliverpbeagle** "Beagle doesn't love going to the vet for annual checkup. But am very brave boy, and human will give me many treats afterwards" 🐶 #oliverpbeagle #vet #beagle #beaglesofinstagram) × 400 Million

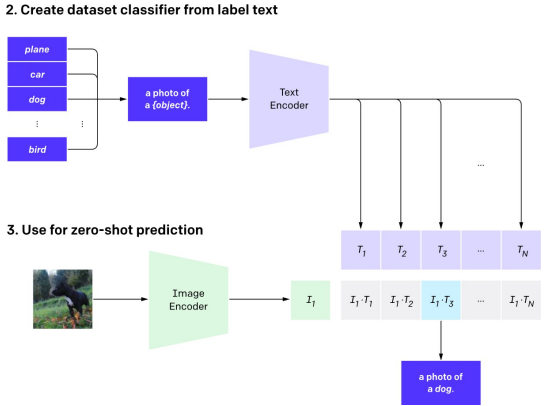
CLIP – Zero Shot Capabilities

Zero-Shot Learning with CLIP: CLIP can classify images into categories without any task-specific training by leveraging the learned image-text embeddings.

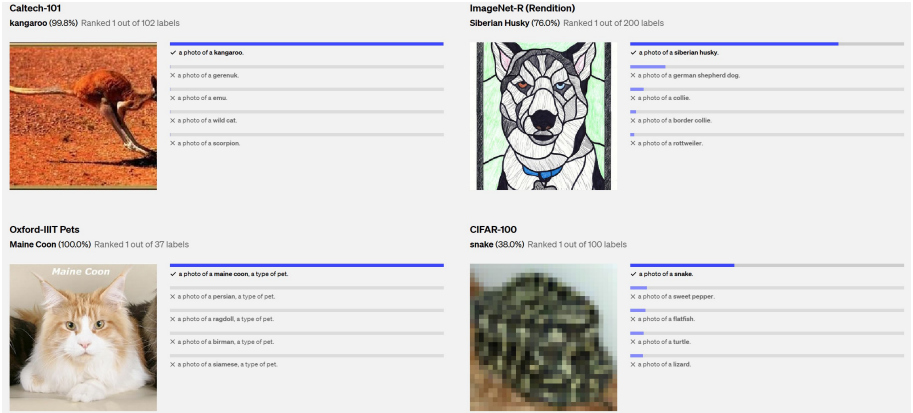
- **How it Works:** For a given image, CLIP computes its embedding and compares it with the embeddings of text descriptions of potential classes.
- **Similarity Measure:** The class with the highest similarity score is selected as the predicted label.
- **Advantages:** This approach allows CLIP to generalize to new tasks and categories without requiring additional training data.

CLIP – Zero Shot Capabilities (cont.)

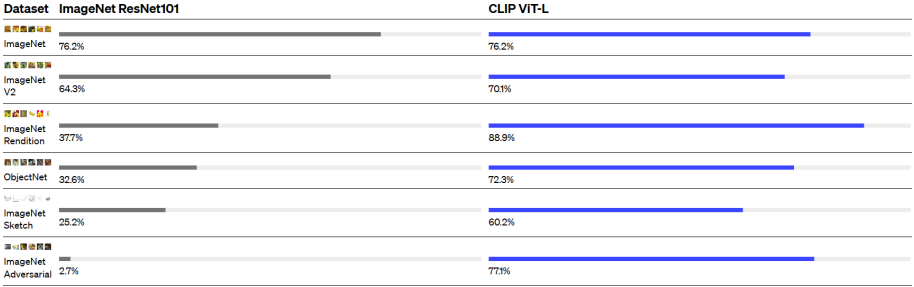
Example: CLIP can classify an image of a dog as "dog" by matching its embedding to text embeddings like "dog" or "cat" and picking the closest match.



CLIP – Zero Shot Capabilities (cont.)



CLIP – Zero Shot Capabilities (cont.)



Using CLIP for generative tasks

CLIP for Generative Tasks: CLIP can be used to guide generative models, such as GANs or diffusion models, by providing a text-based conditioning signal.

- **Text Conditioning:** The text encoder of CLIP can be used to condition the generative model on specific text prompts, allowing it to generate images that match the given descriptions.
- **Loss Function:** The similarity between generated image embeddings and text embeddings can be used as a loss function to optimize the generative model.
- **Applications:** This approach is useful for tasks like text-to-image synthesis, where the goal is to generate images based on textual descriptions.

Using CLIP for generative tasks (cont.)

Generation

A beautiful painting of a building in a serene landscape

Editing

"A cake made of ice"

Video Learning & Generation: **Generative Models for Video & Images**

- Generative models learn to generate new data samples that resemble the training data.
- They can be used for tasks like image synthesis, video generation, and style transfer.
- Common approaches include Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models.

VQ-GAN – Vector Quantized Generative Adversarial Network

VQ-GAN is a generative model that combines the strengths of vector quantization and GANs to generate high-quality images.

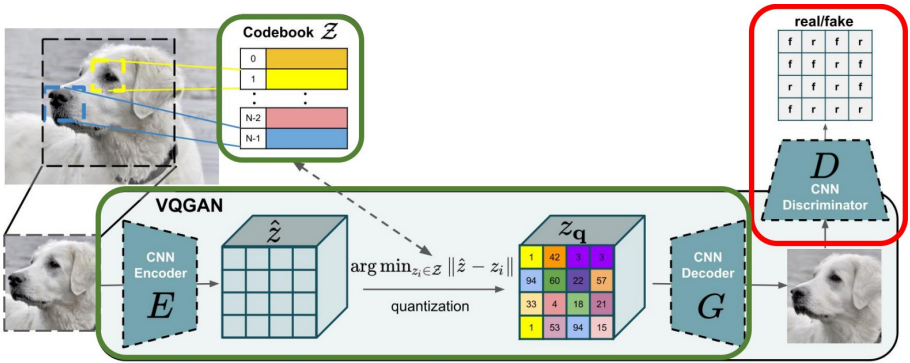
- **Vector Quantization:** Encodes images into discrete latent codes, allowing for efficient representation and generation.
- **GAN Framework:** Uses a discriminator to guide the generator in producing realistic images.
- **Applications:** Image synthesis, super-resolution, and style transfer.

VQ-GAN (cont.)

Architecture: VQ-GAN consists of an encoder, a quantizer, a decoder, and a discriminator.

- **Encoder:** Maps input images to a continuous latent space.
- **Quantizer:** Discretizes the continuous latent space into a finite set of codes.
- **Decoder:** Reconstructs images from the quantized latent codes.
- **Discriminator:** Evaluates the realism of generated images, providing feedback to the generator.

VQ-GAN (cont.)



VQ-VAE + GAN

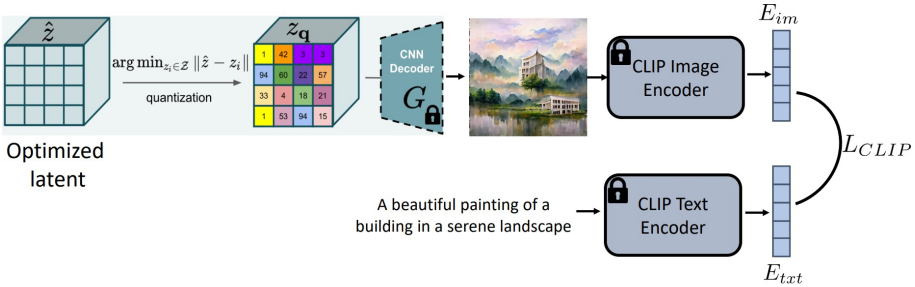
VQ-GAN + CLIP is a powerful combination of two models that enhances image generation and understanding by leveraging the strengths of both vector quantization and contrastive learning.

- **VQ-GAN:** A generative model that uses vector quantization to encode images into discrete latent codes, enabling efficient representation and high-quality image synthesis.
- **CLIP:** A model that learns visual representations by aligning images and text, allowing for zero-shot classification and improved understanding of visual content.

How They Work Together:

- **Image Generation:** VQ-GAN generates images by decoding quantized latent codes, while CLIP provides a text-based conditioning mechanism that guides the generation process.
- **Text Conditioning:** CLIP's text embeddings can be used to condition the VQ-GAN generator, allowing it to produce images that match specific textual descriptions.
- **Loss Function:** The similarity between generated image embeddings and text embeddings from CLIP can be used as a loss function to optimize the VQ-GAN generator, ensuring that the generated images align with the given text prompts.

VQ-GAN + CLIP (cont.)



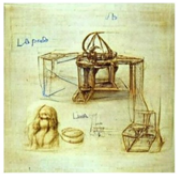
$$L_{CLIP} = 1 - Cos(E_{im}, E_{txt})$$

[VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance, Crowson et al. ICCV 13 2021]

Applications:

- **Text-to-Image Synthesis:** VQ-GAN + CLIP can generate images based on textual descriptions, enabling creative applications like art generation and design.
- **Image Editing:** The combination allows for editing images based on text prompts, such as changing the style or content of an image while preserving its structure.
- **Zero-Shot Learning:** CLIP's zero-shot capabilities can be leveraged to classify generated images into categories without additional training, enhancing the versatility of the model.

VQ-GAN + CLIP (cont.)



"A sketch of 3D printer by da Vinci"

"An autogyro flying, artstation"

"A futuristic city in synthwave style"

"A colored pencil drawing of a waterfall"

"A painting of a city in a deep valley"

"Baba Yaga's house, fantasy art"

Using CLIP for generative tasks

Generation

A beautiful painting of a building in a serene landscape

Editing

"A cake made of ice"

VQ-GAN + CLIP Editing

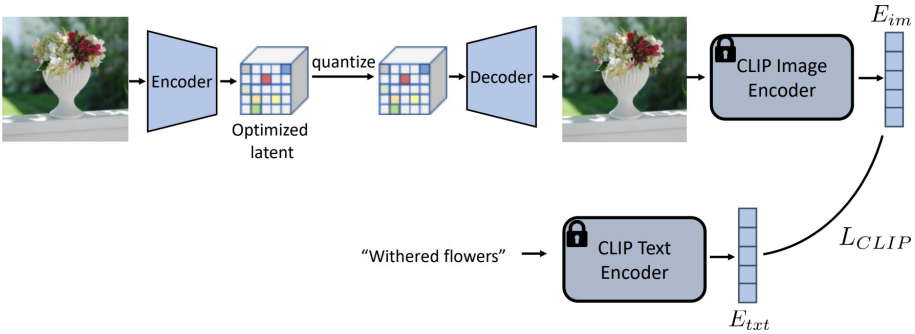
VQ-GAN + CLIP Editing combines the strengths of VQ-GAN for image generation and CLIP for text-guided editing, enabling high-quality image synthesis and manipulation.

- **VQ-GAN:** Generates images from discrete latent codes, allowing for efficient representation.
- **CLIP:** Guides the editing process by understanding text prompts, enabling targeted modifications to generated images.
- **Applications:** Image synthesis, style transfer, and content-aware editing.

How They Work Together:

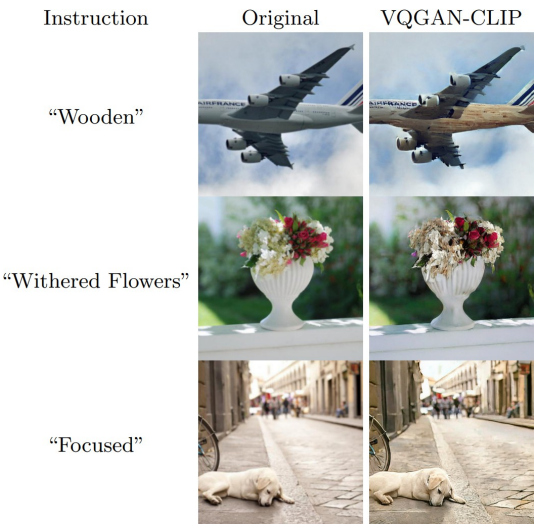
- **Image Generation:** VQ-GAN generates images based on quantized latent codes, while CLIP provides text-based guidance for editing.
- **Text Conditioning:** CLIP's text embeddings condition the VQ-GAN generator, allowing it to produce images that match specific textual descriptions.
- **Loss Function:** The similarity between generated image embeddings and text embeddings from CLIP is used as a loss function to optimize the VQ-GAN generator, ensuring that the generated images align with the given text prompts.

VQ-GAN + CLIP Editing (cont.)



[VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance, Crowson et al. ICCV 13 2021]

VQ-GAN + CLIP Editing (cont.)



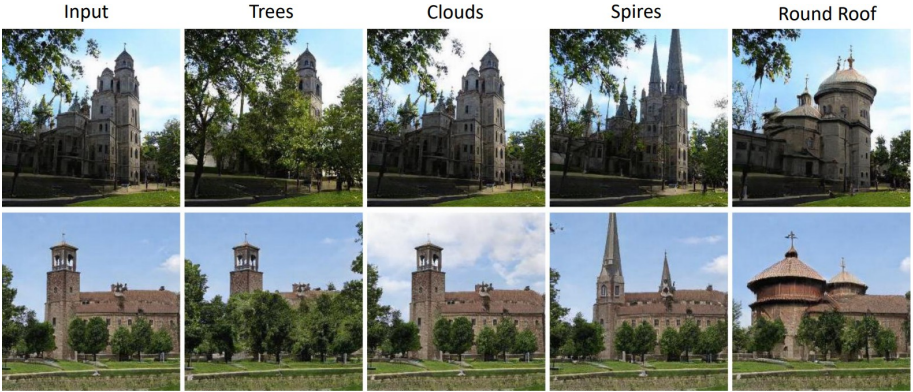
StyleCLIP is a method that combines the power of CLIP (Contrastive Language-Image Pretraining) with StyleGAN to generate images based on text prompts, allowing for fine-grained control over image attributes.

- **CLIP Embeddings:** Uses CLIP to encode text prompts into embeddings that capture semantic information.
- **StyleGAN Manipulation:** Applies these embeddings to manipulate the latent space of StyleGAN, enabling the generation of images that match the desired attributes described in the text.
- **Applications:** Useful for tasks like image editing, attribute manipulation, and creative content generation.
- **Benefits:** Provides a powerful way to generate diverse images based on textual descriptions.
- **Costs:** Requires careful tuning of the StyleGAN latent space and may produce artifacts if not managed properly.

StyleCLIP (cont.)

StyleCLIP (cont.)

StyleCLIP (cont.)



Text2Live is a method that generates live video content from text descriptions, enabling the creation of dynamic and engaging video sequences.

- **Text-to-Video Generation:** Converts textual descriptions into video sequences, allowing for the synthesis of new video content.
- **Dynamic Content Creation:** Generates videos that can change over time based on the input text, providing a rich visual experience.
- **Applications:** Useful in various domains such as entertainment, education, and marketing, where dynamic video content is required.

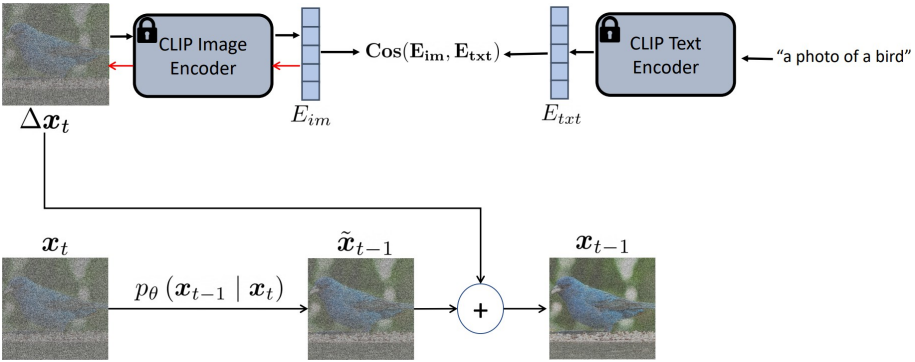
Text2Live (cont.)

[Text2LIVE: Text-Driven Layered Image and Video Editing, Bar-Tal Ofri-Amar and Fridman et al. ECCV 2022]

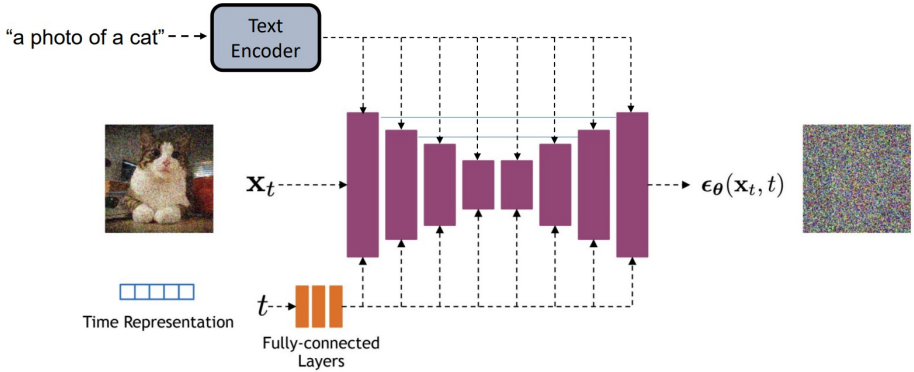
Conditional image generation with CLIP Using Diffusion Models combines the power of diffusion models for image synthesis with CLIP's ability to understand and condition on text prompts, enabling the generation of images that align with specific textual descriptions.

- **Diffusion Models:** These models generate images by iteratively refining noise into coherent images, allowing for high-quality synthesis.
- **CLIP:** Provides text embeddings that guide the diffusion process, ensuring that the generated images match the desired attributes described in the text.
- **Applications:** Image synthesis, style transfer, and content-aware editing based on textual prompts.

Conditional image generation with CLIP Using Diffusion (cont.)



[Diffusion Models Beat GANs on Image Synthesis, Dhariwal and Nichol et al. NeurIPS 2021]

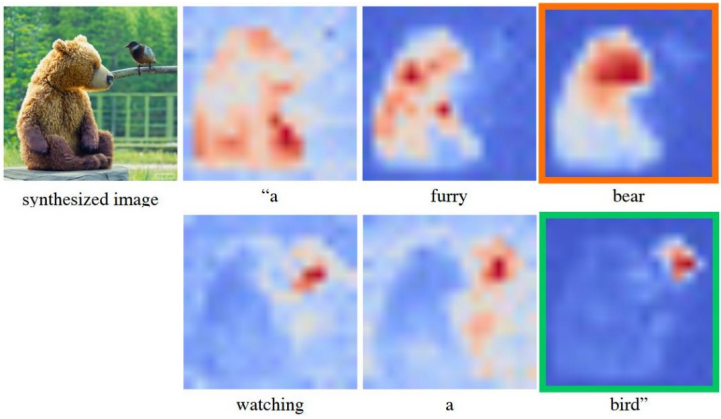

"a photo of a cat"
Text Encoder
$\mathbf{x}_t$
$\epsilon_\theta(\mathbf{x}_t, t)$
Time Representation
t
Fully-connected Layers

Text conditioning in Diffusion Models

Text conditioning in Diffusion Models enhances the generation of images by incorporating textual descriptions, allowing for more controlled and context-aware image synthesis.

- **Diffusion Models:** These models generate images by iteratively refining noise into coherent images, enabling high-quality synthesis.
- **Text Conditioning:** Uses text embeddings to guide the diffusion process, ensuring that the generated images align with specific textual descriptions.
- **Applications:** Image synthesis, style transfer, and content-aware editing based on textual prompts.

Text conditioning in Diffusion Models (cont.)



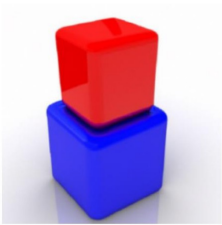
Average attention maps across all timestamps

[Prompt-to-Prompt Image Editing with Cross Attention Control, Hertz and Mokady et al., ICRL 2022]

Text conditioning in Diffusion Models (cont.)

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and purple party hat"



"a red cube on top of a blue cube"

"a high-quality oil painting of a psychedelic hamster dragon"

Video Learning & Generation: **Summary**

Key Takeaways

- Video understanding evolves from single-frame analysis to capturing global context using convolution, recurrence, and attention mechanisms.
- Temporal modeling is essential for tasks like action recognition and future prediction.
- Self-supervised learning reduces the need for extensive manual annotation.
- CLIP connects vision and language; GANs and diffusion models enable generative video synthesis.

Video Learning & Generation: **References**

References

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri,
"Learning Spatiotemporal Features with 3D ConvNets,"
in *ICCV*, 2015.
arxiv.org/abs/1412.0767

[2] J. Carreira and A. Zisserman,
"Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,"
in *CVPR*, 2017.
arxiv.org/abs/1705.07750

[3] X. Wang, R. Girshick, A. Gupta, and K. He,
"SlowFast Networks for Video Recognition,"
in *ICCV*, 2019.
arxiv.org/abs/1812.03982

References (cont.)

[4] G. Bertasius, H. Wang, and L. Torresani,
"Is Space-Time Attention All You Need for Video Understanding?"
in *NeurIPS*, 2021.
arxiv.org/abs/2102.05095

[5] H. Sun et al.,
"Ego4D: Around the World in 3,000 Hours of Egocentric Video,"
in *CVPR*, 2022.
ego4d-data.org

[6] A. Radford et al.,
"Learning Transferable Visual Models From Natural Language Supervision,"
in *ICML*, 2021.
arxiv.org/abs/2103.00020

References (cont.)

[7] P. Esser, R. Rombach, and B. Ommer,
"Taming Transformers for High-Resolution Image Synthesis,"
in *CVPR*, 2021.
arxiv.org/abs/2012.09841

[8] J. Ho, W. Chan, C. Saharia, et al.,
"Video Diffusion Models,"
in *NeurIPS*, 2022.
arxiv.org/abs/2204.03458

[9] L. Yang et al.,
"VideoCrafter: Open Diffusion Models for High-Quality Video Generation,"
2023.
arxiv.org/abs/2309.08485

References (cont.)

[10] S. Khandelwal, A. Jain, S. Choudhury, et al.,
"TrajDiffuser: Diffusion for Trajectory Prediction,"
in *NeurIPS*, 2023.
arxiv.org/abs/2306.08694

[11] F.-F. Li, Y. Li, and R. Gao,
"Stanford CS231n: Deep Learning for Computer Vision,"
cs231n.stanford.edu

[12] A. Shocher, S. Bagon, M. Galun, and T. Dekel,
"WAIC DL4CV: Deep Learning for Computer Vision: Fundamentals and Applications,"
WAIC-DL4CV

[13] J. Johnson,
"UMich EECS 498.008/598.008: Deep Learning for Computer Vision,"
UMich DL4CV

Credits

Dr. Prashant Aparajeya

Computer Vision Scientist — Director(AISimply Ltd)

p.aparajeya@aisimply.uk

This project benefited from external collaboration, and we acknowledge their contribution with gratitude.