Lecture 10: Videos

June 6\textsuperscript{st}, 2021

Tali Dekel
Videos

Videos are all around us
Span an enormous space of spatial and temporal signals
Challenges in Videos: size of video

Size of video >> size of image

Computational constrains \(\rightarrow\) short, low-res clips

3 x H x W

4D tensor:
\[ T \times 3 \times H \times W \]

time

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

Reduce spatial and temporal resolution

\(~30\) frames per second (fps)

5fps, half the spatial resolution

Slide inspiration: Justin Johnson, EECS 498-007
Challenges in Videos: size of video

Computational constrains $\rightarrow$ short, low-res clips

Input video

Walking
Running
Cycling
Jumping

Original video (long, high FPS)

Training: Short, low FPS

Test time: inference on different short clips, average predictions

Slide inspiration: Justin Johnson, EECS 498-007
Challenges in Videos: Videos Datasets

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

“ImageNet”-equivalent dataset for videos?

Massive human labelling efforts

UCF101
YouTube videos
13320 videos, 101 action categories

Kinetics
YouTube videos
650,000 video clips, 600 human action classes

Sports-1M
YouTube videos
1,133,157 videos, 487 sports labels

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

Deep Learning-based Models for Videos
• How to reduce computation cost without sacrificing accuracy?
• What architecture to best capture temporal patterns?

Self-Supervision in Videos
• Which types of pretext tasks can we define to capture temporal information?
• Applications

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Single-Frame Baseline

- Train 2D CNN to classify video frames independently
Models for Videos: Single-Frame Baseline

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

Often a surprisingly strong baseline!

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Late Fusion

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

![Diagram of late fusion model]

Frame features: \(D \times H' \times W'\)

Flatten + concatenate and feed to FC layers

Frame features: \(D \times H' \times W'\)

“Biking”

2D CNN

MLP (FC)

MLP (FC)

Input video frames

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Late Fusion w/ pooling

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

"Biking"

MLP (FC)  MLP (FC)

Pooled feature: D

Average Pool over space and time

Concatenated features: $T \times D \times H' \times W'$

2D CNN  2D CNN  2D CNN  2D CNN  2D CNN

Input video frames

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Late Fusion w/ pooling

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

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

Cons: late fusion is late... hard to represent low level motion between frames
Models for Videos: Early Fusion

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

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

Input video frames

Reshaped input: 3T x H x W
Input: T x 3 x H x W

“Biking”
MLP (FC)
MLP (FC)

2D CNN

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Early Fusion

Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel

Input: $T \times 3 \times H \times W$

Weights: $C \times T \times 3 \times h \times w$

Output: $C \times H' \times W'$
Models for Videos: Early Fusion

Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel

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

Large motion occurred

Input: $T \times 3 \times H \times W$

Weights: $C \times T \times 3 \times h \times w$

Output: $C \times H' \times W'$

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Early Fusion

Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel

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

Input: $T \times 3 \times H \times W$

Weights: $C \times T \times 3 \times h \times w$

Output: $C \times H' \times W'$
Models for Videos: Early Fusion

Pros: Allow the network to learn local motion characteristics

Cons:
• Not temporal shift-invariant
• Only have one layer of temporal processing

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Slow Fusion a.k.a 3D Convs

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

Filters are sliding in both space and time

Input video frames

Input: $T \times 3 \times H \times W$

Reshaped input: $3T \times H \times W$

"Biking"

MLP (FC)

MLP (FC)

3D CNN

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Slow Fusion a.k.a 3D Convs

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

Input: $T \times 3 \times H \times W$

Weights: $C \times t \times 3 \times h \times w$

Output: $C \times T' \times H' \times W'$

Temporal shift-invariant!
Models for Videos: Slow Fusion a.k.a 3D Convs

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

Input: $T \times 3 \times H \times W$

Weights: $C \times t \times 3 \times h \times w$

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

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Models for Videos: Multi-scale

How can we reduce computational cost while maintaining accuracy?

- Reduce video resolution → lower performance
- Reduce network’s capacity → lower performance

- Context stream \textit{(low res)}: process low res video frames (H/2, W/2)
- Fovea stream \textit{(high res)}: process a (H/2, W/2) crop from the original resolution

Reduce the input dimensionality by half

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Action classification -- Sports-1M

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

• Ground truth
• Correct prediction
• Incorrect prediction

Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014
Action classification -- Sports-1M

Sports-1M Top-5 Accuracy

86
84
82
80
78
76
74
72

Single Frame | Early Fusion | Late Fusion | 3D CNN
77.7        | 76.8         | 78.7        | 80.2

Single frame: a shockly powerful baseline

This is from 2014...
Models for Videos: C3D (Convolutional 3D)

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

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

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

Tran et al, “Learning Spatiotemporal Features with 3D Convolutional Networks”, ICCV 2015
Non-deep learning video classification

Motion is the most informative cue for action recognition \(\rightarrow\) design hand crafted motion features:

Aggregate local motion features to compute a global representation of the video \(\rightarrow\) linear SVM for action recognition

MODEL MOTION EXPLICITLY

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

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

Mean accuracy on UCF-101

- Slow fusion (3D CNN)
- Hand-crafted motion
Explicitly modeling motion in deep-based models

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

Optical flow provides **local motion cues**

Color wheel
Saturation = mag.
Color = angle

Optical flow between two frames

Frame $t$

Frame $t+1$
Two Stream Networks: modeling motion explicitly

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

"Biking"

FC layers
fusion

"Single Frame" baseline

Spatial stream ConvNet

Temporal stream ConvNet

Precomputed flow fields between consecutive frames

Using "Early Fusion" baseline

Frame t

Multi-frame optical flow

Accuracy on UCF-101

65.4 73 83.7 86.9 88

3D CNN  Spatial only  Temporal only  Two-stream (fuse by average)  Two-stream (fuse by SVM)

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

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

“Biking”

FC layers

fusion

“Single Frame” baseline

Spatial stream ConvNet

Temporal stream ConvNet

using “Early Fusion” baseline

Frame t

Multi-frame optical flow

Precomputed flow fields between consecutive frames

Accuracy on UCF-101

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D CNN</td>
<td>65.4</td>
</tr>
<tr>
<td>Spatial only</td>
<td>73</td>
</tr>
<tr>
<td>Temporal only</td>
<td>83.7</td>
</tr>
<tr>
<td>Two-stream (fuse by average)</td>
<td>86.9</td>
</tr>
<tr>
<td>Two-stream (fuse by SVM)</td>
<td>88</td>
</tr>
</tbody>
</table>

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

**Inflating 2D networks to 3D (I3D)**
Take an existing 2D CNN model $\rightarrow$ convert it to a 3D CNN model
*Transfer the weights from 2D and 3D*
Carreira and Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”, CVPR 2017

**Long range temporal processing**
Use LSTMs and RNNs to model long range temporal information

**Long range temporal processing**
Self attention, non-local networks, Transformers
Self-Supervision in Videos

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

Video: https://ajabri.github.io/timecycle/
Self-Supervision in Videos: frame ordering

**Training data:** shuffled video frames, original video frames

**Pretext task:** predict if the frames are in the correct temporal order (binary classification task)

Self-Supervision in Videos: frame ordering

Generating positive and negative examples

Triplet Siamese network for sequence verification

Self-Supervision in Videos: frame ordering

Transfer learning: fine-tune spatial stream for video classification

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Initialization</th>
<th>Mean Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF101</td>
<td>Random</td>
<td>38.6</td>
</tr>
<tr>
<td></td>
<td>(Ours) Tuple verification</td>
<td><strong>50.2</strong></td>
</tr>
<tr>
<td>HMDB51</td>
<td>Random</td>
<td>13.3</td>
</tr>
<tr>
<td></td>
<td>UCF Supervised</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>(Ours) Tuple verification</td>
<td><strong>18.1</strong></td>
</tr>
</tbody>
</table>
Self-Supervision in Videos: Colorization of Moving Objects

**Ultimate goal:** Tracking

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

**Training data:** grayscale videos + original color videos

Reference frame

Grayscale video

Vondrick et al., Tracking Emerges by Colorizing Videos, ECCV 2018
Self-Supervision in Videos: Colorization of Moving Objects

Video colorization by learning to copy color from a reference frame

Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018
Self-Supervision in Videos: Colorization of Moving Objects

Video colorization by learning to copy color from a reference frame

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

- Cross-entropy over color distribution

Grayscale Video

- Reference Frame
- Target Frame

CNN

Embeddings

- Reference Colors
  - $c_i$

Predicted Colors

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

Similarity in the embedding space

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

Vondrick et. al, Tracking Emerges by Colorizing Videos, ECCV 2018
Self-Supervision in Videos: Colorization of Moving Objects

Video colorization by learning to copy color from a reference frame

\[
\min_{\theta} \sum_j \mathcal{L}(y_j, c_j)
\]

cross-entropy over color distribution

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

\[y_j = \sum_i A_{ij} c_i\]

linear combination of the reference colors

\[A_{ij} = \frac{\exp(f_i^T f_j)}{\sum_k \exp(f_k^T f_j)}\]

Similarity in the embedding space

Vondrick et al., Tracking Emerges by Colorizing Videos, ECCV 2018
**Self-Supervision in Videos: Colorization of Moving Objects**

Video colorization by learning to copy color from a reference frame.

A linear combination of the reference colors:

\[ y_j = \sum_i A_{ij} c_i \]

where \( A \) is a similarity matrix between reference and target (rows sum to one).

---

**Inputs**

- Reference
- Held-out video

**Predicted Segmentations**

Reference colors applied to the videos.
Self-Supervision in Videos: Learning correspondence

**Ultimate goal:** Correspondence

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

\[ \text{Similarity} \]

\[ \text{CNN} \rightarrow \text{Similarity} \]

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

Tracking → Similarity

Slide credit: https://ajabri.github.io/timecycle/
Self-Supervision in Videos: Learning correspondence

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

**How to obtain supervision?**

**Supervision:** Cycle-Consistency in Time

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019
Self-Supervision in Videos: Learning correspondence

**Supervision:** Cycle-Consistency in Time

**Challenge:** Occlusions

Skip-cycles: skipping occlusions

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019
Self-Supervision in Videos: Learning correspondence

Differentiable tracker: densely match features in learned feature space

\[ A_{ij} = \frac{\exp(f_i^T f_j)}{\sum_k \exp(f_k^T f_j)} \]

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019
Self-Supervision in Videos: Learning correspondence

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

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019
Self-Supervision in Videos: Learning correspondence

Wang and Efros, Learning Correspondence from the Cycle-consistency of Time, CVPR 2019
Self-Supervision in Videos: Temporal cycle consistency

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

Jabri et. al, Space time correspondence as Contrastive Random Walk, NeurIPS 2020
Self-Supervision in Videos: Learning the Speediness in Videos

Ultimate goal: Watch video content faster by adaptively speeding up the video
“Speediness” in Videos

Slower

Normal speed

Faster

Joint work with: Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, Bill Freeman, Miki Rubinstein and Michal Irani, CVPR 2020
Self-Supervision in Videos: Learning the Speediness in Videos

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

“Learning and Using the Arrow of Time”, Wei at. al, CVPR 2018
Self-Supervision in Videos: Learning the Speediness in Videos

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

Self-supervised training on Kinetics

Input segment (30 frames)

Self-supervised training on Kinetics

Normal speed or Sped Up

Learning properties of natural motion, avoid “easy cheats” → very challenging!
Self-Supervision in Videos: Learning the Speediness in Videos

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

**Training data:** sped up video segments + original video segments

---

**Input**

- N
- T

**3D Conv Network**

*Based on S3D-G*

**Space-time Features**

(1024 Channels)

**Pooling**

(spatial max, temporal average)

**1x1 Conv**

1 x 1024

---

**Normal Speed**

or

**Sped Up**

---

*“Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification”, Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy, ECCV’18.*
Self-Supervision in Videos: Learning the Speediness in Videos

**Inference:** sliding window $\rightarrow$ prediction for every frame
Self-Supervision in Videos: Learning the Speediness in Videos

From “Speediness” to Speedup factor:
Low speediness $\rightarrow$ speedup more
High speediness $\rightarrow$ speedup less
Learning the Speediness in Videos: Adaptive Video Speedup

Uniform Speedup 2x

Adaptive Speedup 2x (ours)
Learning the Speediness in Videos: Transfer Learning

Pre-trained SpeedNet

Self Supervised Action Recognition

<table>
<thead>
<tr>
<th>Method</th>
<th>Initialization</th>
<th>Architecture</th>
<th>Supervised accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>UCF101</td>
</tr>
<tr>
<td>Random init</td>
<td></td>
<td>S3D-G</td>
<td>73.8</td>
</tr>
<tr>
<td>ImageNet inflated</td>
<td></td>
<td>S3D-G</td>
<td>86.6</td>
</tr>
<tr>
<td>Kinetics supervised</td>
<td></td>
<td>S3D-G</td>
<td>96.8</td>
</tr>
<tr>
<td>CubicPuzzle [19]</td>
<td></td>
<td>3D-ResNet18</td>
<td>65.8</td>
</tr>
<tr>
<td>Order [40]</td>
<td></td>
<td>R(2+1)D</td>
<td>72.4</td>
</tr>
<tr>
<td>DPC [13]</td>
<td></td>
<td>3D-ResNet34</td>
<td>75.7</td>
</tr>
<tr>
<td>AoT [38]</td>
<td></td>
<td>T-CAM</td>
<td>79.4</td>
</tr>
<tr>
<td>SpeedNet (Ours)</td>
<td></td>
<td>S3D-G</td>
<td><strong>81.1</strong></td>
</tr>
</tbody>
</table>

Video Retrieval

Query

Retrieved top-3 results
Learning the Speediness in Videos: CAM visualizations

“Memory Eleven” artistic video by Bill Newsinge

Our space-time speediness visualization

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

blue/green = normal speed

yellow/orange = slowed down
Next tutorial: “GPU Fundamentals”

Next class: ”Neural Rendering”