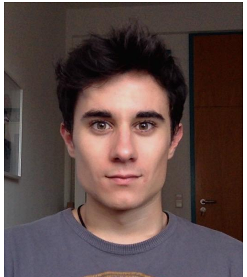
#DLUPC

Day 3 Lectures 1 & 2

Video Analysis

http://bit.ly/dlcv2018



Víctor Campos
victor.campos@bsc.es

PhD Candidate

Barcelona Supercomputing Center

telecom BCN

BSC

Outline

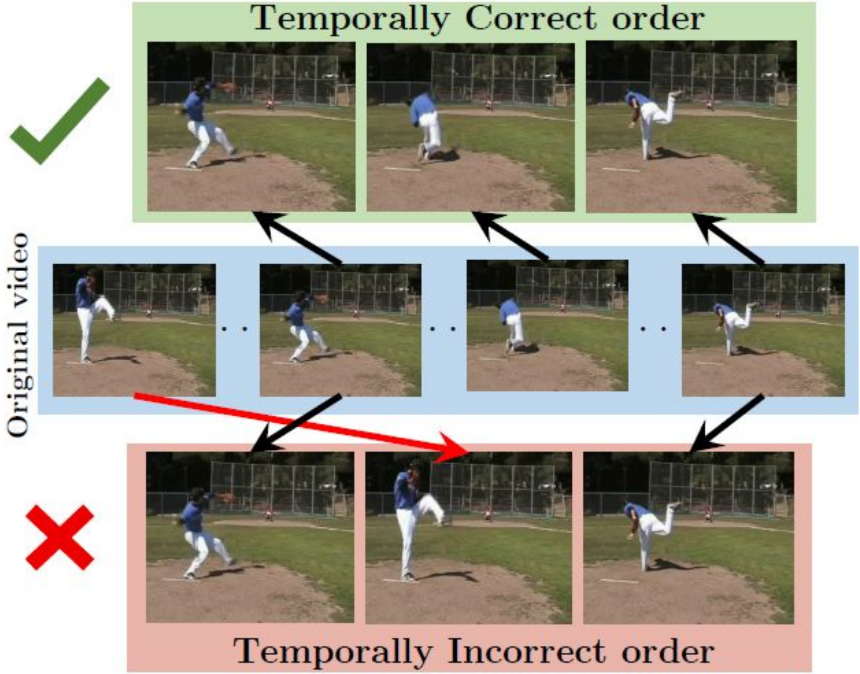
1. **Self-supervision from videos**
2. Architectures for video analysis
3. Exploiting redundancy in videos
4. Tips and tricks for applying deep learning to video

Self-supervision: motivation

- Neural Networks generally need tons of annotated data, but generating those annotations is expensive
- Can we create some tasks for which we can get free annotations?
  - Yes! And videos are very convenient for this
- We want to
  - Frame the problem as a supervised learning task…
  - … but collecting annotations in an unsupervised manner

Self-supervision: examples

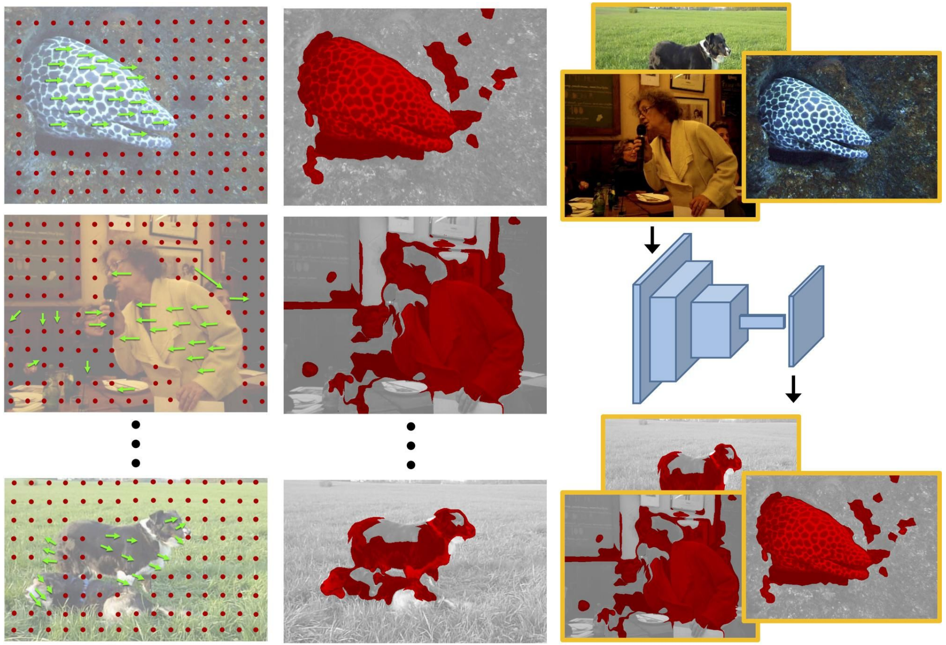
Temporal coherence



Misra et al., Shuffle and learn: unsupervised learning using temporal order verification. ECCV 2016

Self-supervision: examples

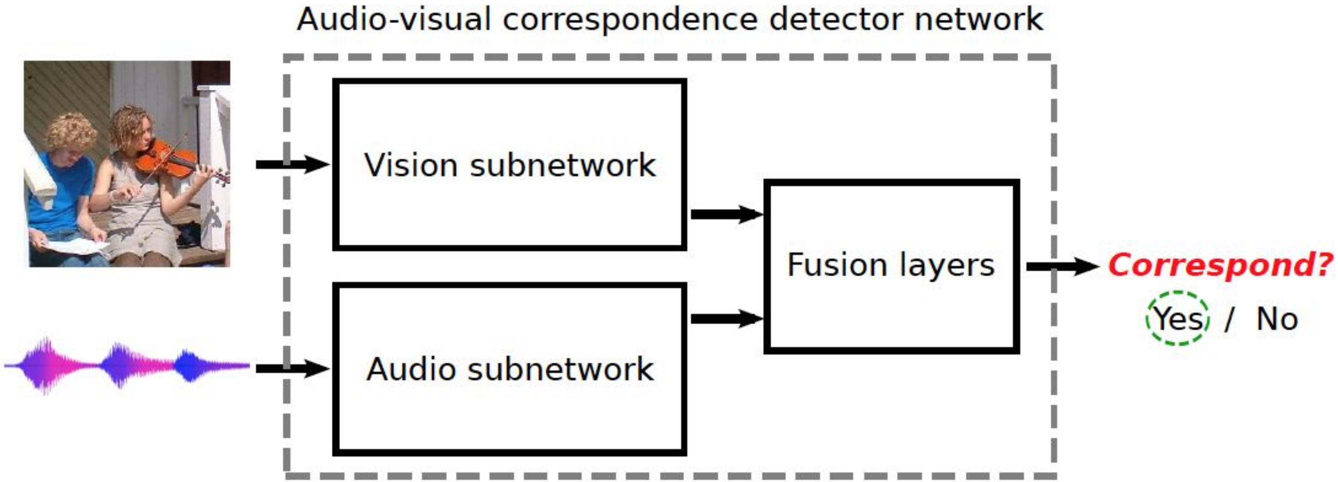
Motion as a proxy for foreground segmentation



Pathak et al., Learning features by watching objects move. CVPR 2017

Self-supervision: examples

Correspondence between images and audio



Arandjelović & Zisserman. Look, Listen and Learn. ICCV 2017.

Outline

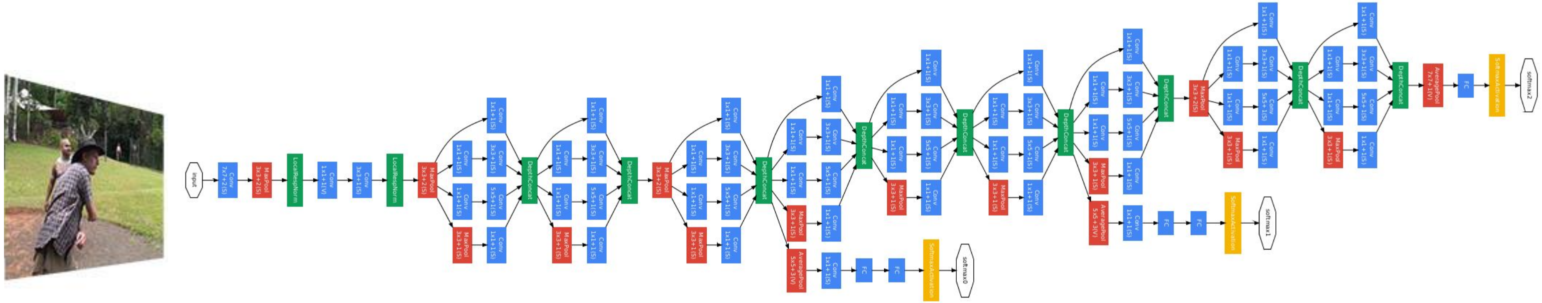
1. Self-supervision from videos
2. **Architectures for video analysis**
3. Exploiting redundancy in videos
4. Tips and tricks for applying deep learning to video

What is a video?

- Formally, a video is a 3D signal
  - Spatial coordinates: *x*, *y*
  - Temporal coordinate: *t*
- If we fix *t*, we obtain an image. We can understand videos as sequences of images (a.k.a. frames)

How do we work with images?

- Convolutional Neural Networks (CNN) provide state of the art performance on image analysis tasks



- How can we extend CNNs to image sequences?

CNNs for sequences of images

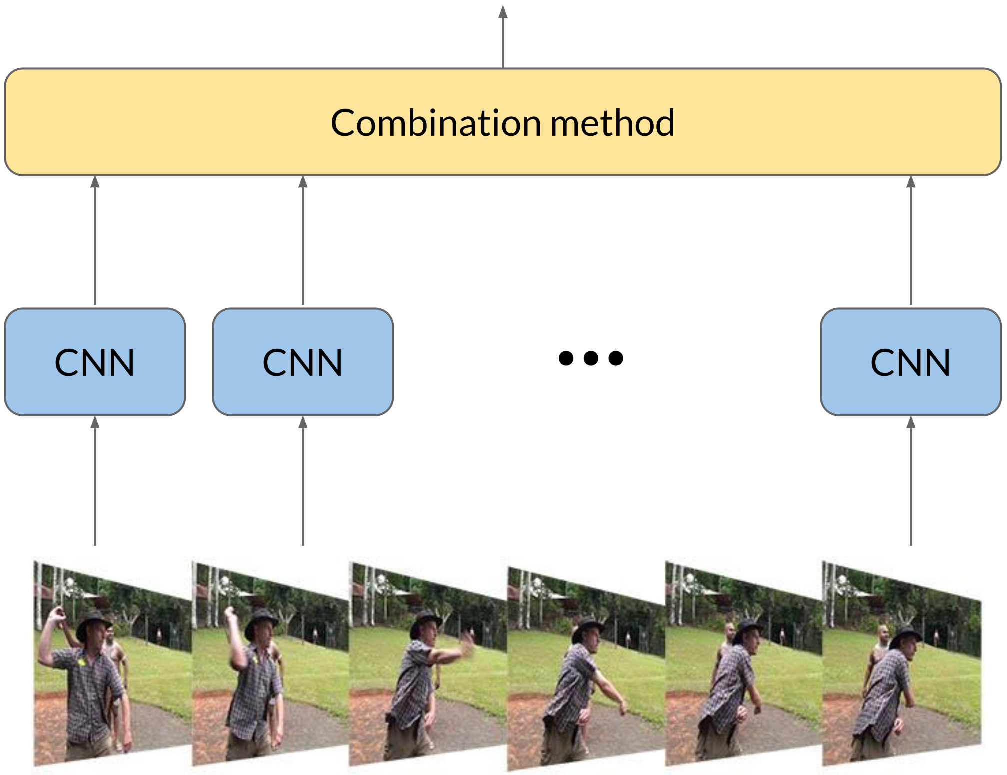
Several approaches have been proposed

1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. Two-stream CNN

Each method leverages the temporal information in a different way

f  $\longrightarrow \hat{y}$

Single frame models

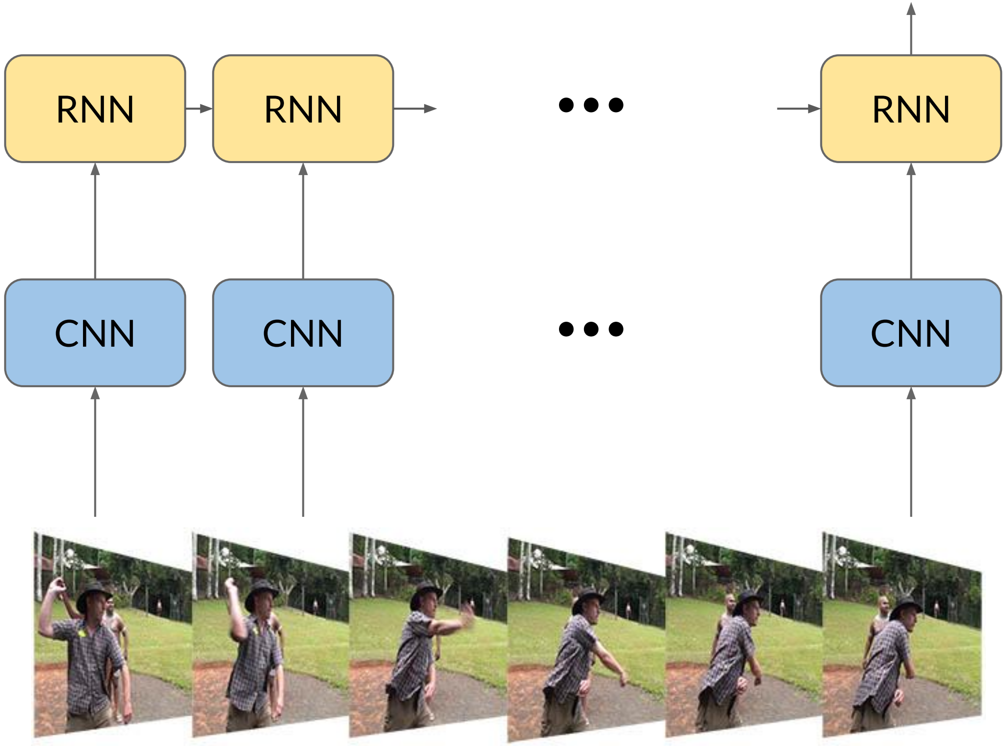


Combination is commonly implemented as a small NN on top of a pooling operation (e.g. max, sum, average).

Drawback: pooling is not aware of the temporal order!

Ng et al., Beyond short snippets: Deep networks for video classification, CVPR 2015

CNN + RNN



Recurrent Neural Networks are well suited for processing sequences.

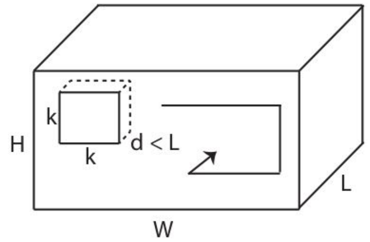
Drawback: RNNs are sequential and cannot be parallelized.

Donahue et al., Long-term Recurrent Convolutional Networks for Visual Recognition and Description, CVPR 2015

3D Convolutions: C3D

We can add an extra dimension to standard CNNs:

- An image is a HxWxD tensor: MxNxD’ conv filters
- A video is a TxHxWxD tensor: KxMxNxD’ conv filters



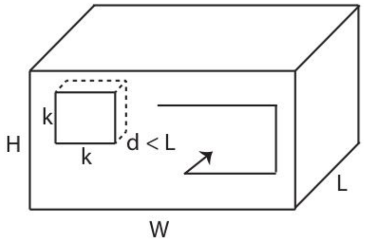
The video needs to be split into chunks (also known as *clips*) with a number of frames that fits the receptive field of the C3D. Usually clips have 16 frames.

Drawbacks:

- How can we handle longer videos?
- How can we capture longer temporal dependencies?
- How can we use pre-trained networks?

Tran et al., Learning spatiotemporal features with 3D convolutional networks, ICCV 2015

From 2D CNNs to 3D CNNs

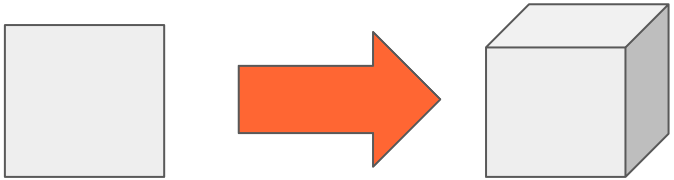


We can add an extra dimension to standard CNNs:

- An image is a HxWxD tensor: MxNxD' conv filters
- A video is a TxHxWxD tensor: KxMxNxD' conv filters

We can convert an MxNxD' conv filter into a KxMxNxD' filter by replicating it K times in the time axis and scaling its values by 1/K.

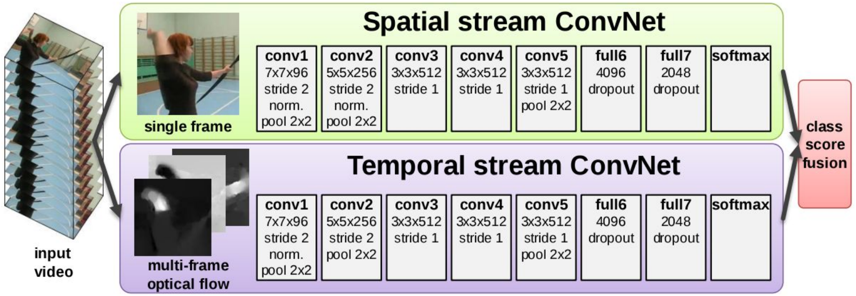
- This allows to leverage networks pre-trained on ImageNet and alleviate the computational burden associated to training from scratch



Feichtenhofer et al., Spatiotemporal Residual Networks for Video Action Recognition, NIPS 2016
Carreira et al., Quo vadis, action recognition? A new model and the kinetics dataset, CVPR 2017

Two-stream CNNs

Single frame models do not take into account motion in videos. Proposal: extract optical flow for a stack of frames and use it as an input to a CNN.



Drawback: not scalable due to computational requirements and memory footprint.

Simonyan & Zisserman, Two-stream convolutional networks for action recognition in videos, ICCV 2015

Outline

1. Self-supervision from videos
2. Architectures for video analysis
3. **Exploiting redundancy in videos**
4. Tips and tricks for applying deep learning to video

Problem definition

- So far, we considered video-level predictions

$f\Big($  $\Big) \longrightarrow \hat{y}$

- What about frame-level predictions?

$f\Big($  $\Big) \longrightarrow (\hat{y}_1, \ldots, \hat{y}_N)$

- What about applications which require low latency?

Minimizing latency

Not all methods are suited for real-time applications

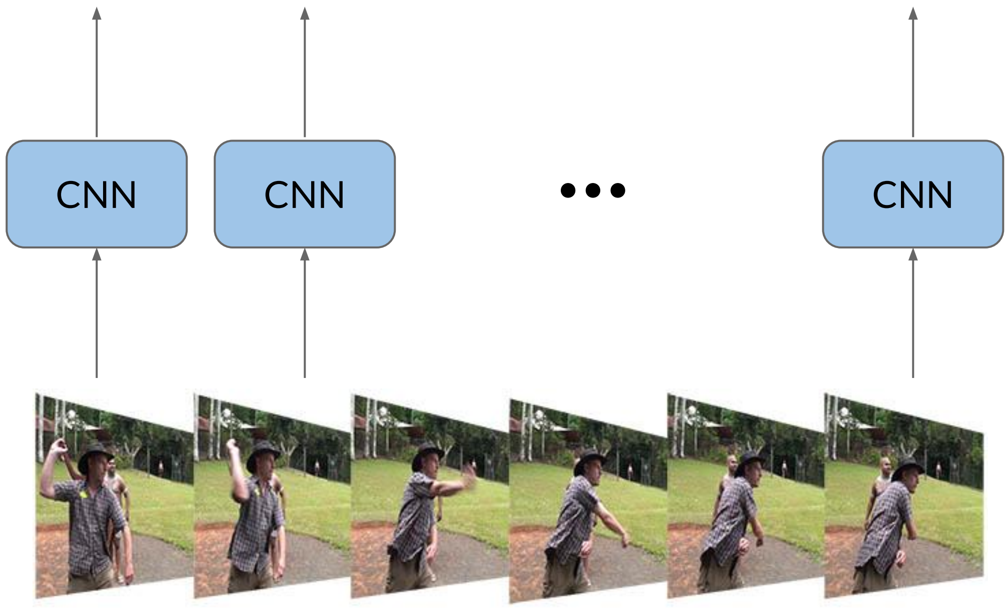
1. Single frame models
2. CNN + RNN
3. 3D convolutions
4. Two-stream CNN

Minimizing latency

Not all methods are suited for real-time applications

1. Single frame models
2. CNN + RNN
3. ~~3D convolutions~~
4. ~~Two stream CNN~~

Single frame models



Single frame models: redundancy

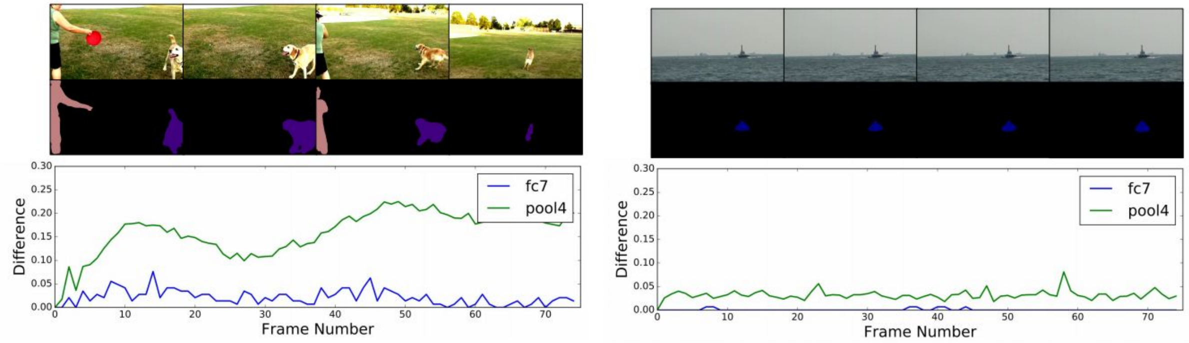
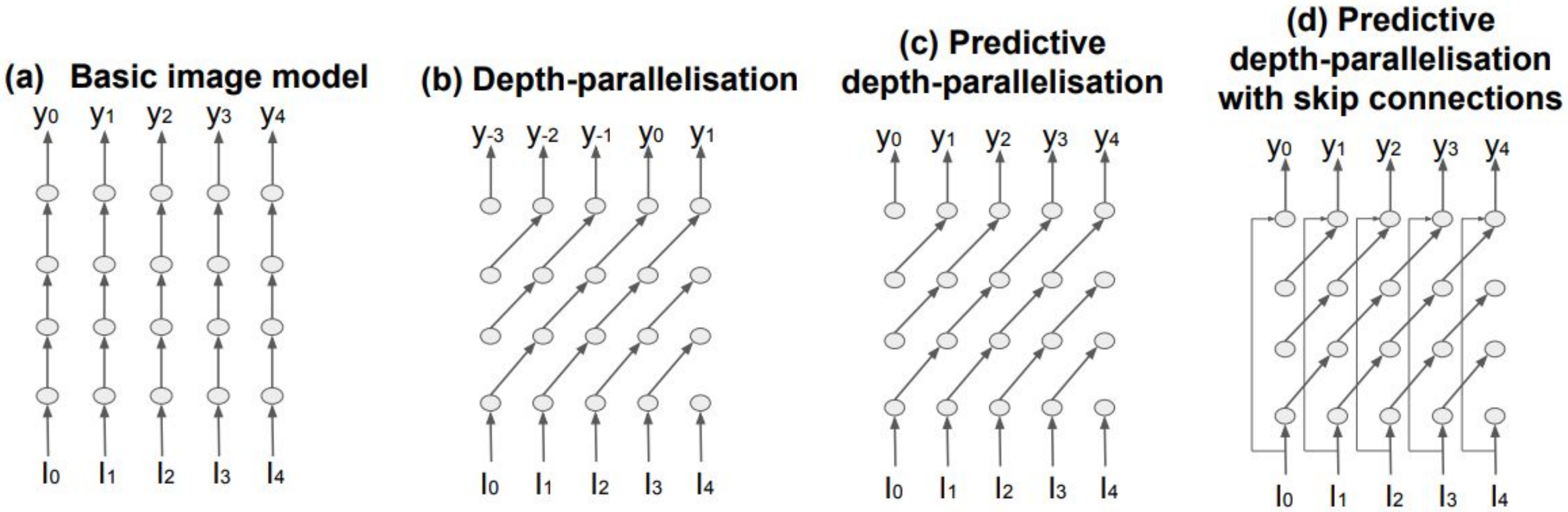


Fig. 2: The proportional difference between adjacent frames of semantic predictions from a mid-level layer (`pool4`, green) and the deepest layer (`fc7`, blue) are shown for the first 75 frames of two videos. We see that for a video with lots of motion (left) the difference values are large while for a relatively static video (right) the difference values are small. In both cases, the differences of the deeper `fc7` are smaller than the differences of the shallower `pool4`. The "velocity" of deep features is slow relative to shallow features and most of all the input. At the same time, the differences between shallow and deep layers are dependent since the features are compositional.

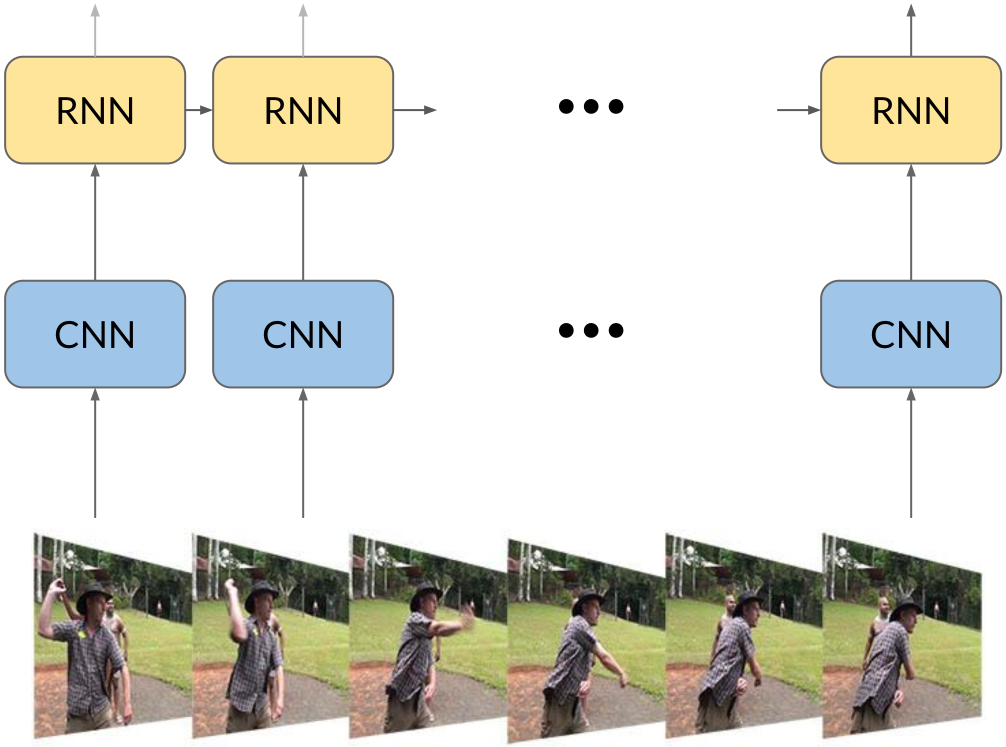
Shellhamer et al., Clockwork Convnets for Video Semantic Segmentation, ECCV 2016

Single frame models: exploiting redundancy

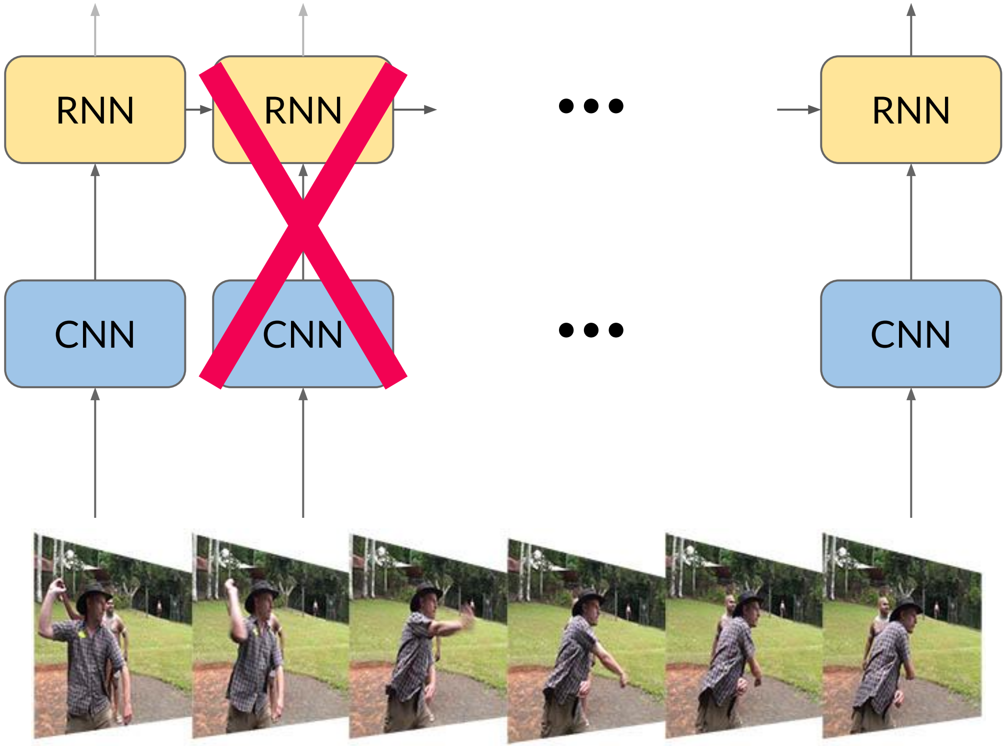


Carreira et al., Massively Parallel Video Networks, arXiv 2018

CNN+RNN: redundancy



CNN+RNN: exploiting redundancy



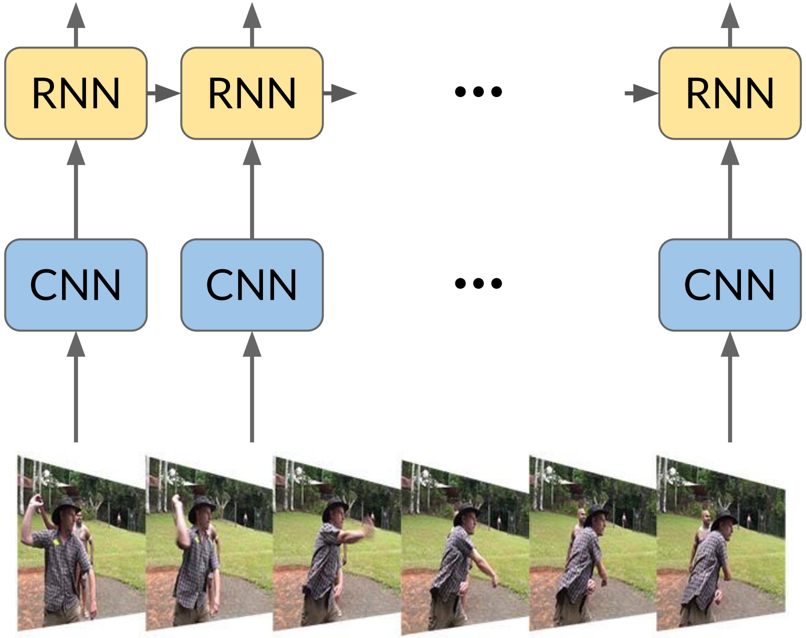
After processing a frame, let the RNN decide how many future frames can be skipped

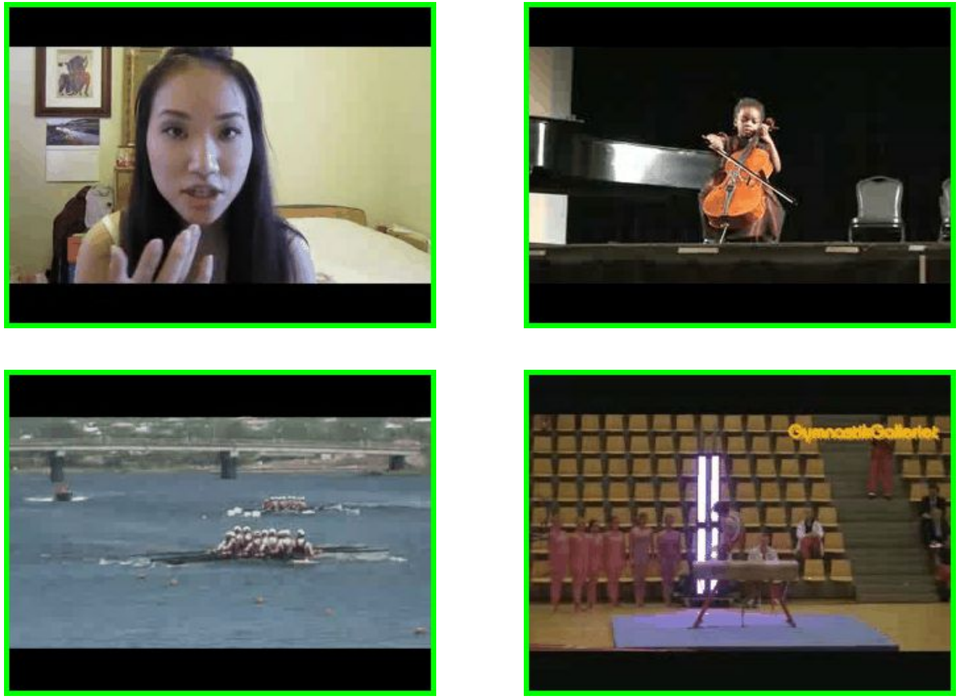
In skipped frames, simply copy the output and state from the previous time step

There is no ground truth for which frames can be skipped. The RNN **learns** it by itself during training!

Campos et al., Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks, ICLR 2018

CNN+RNN: exploiting redundancy





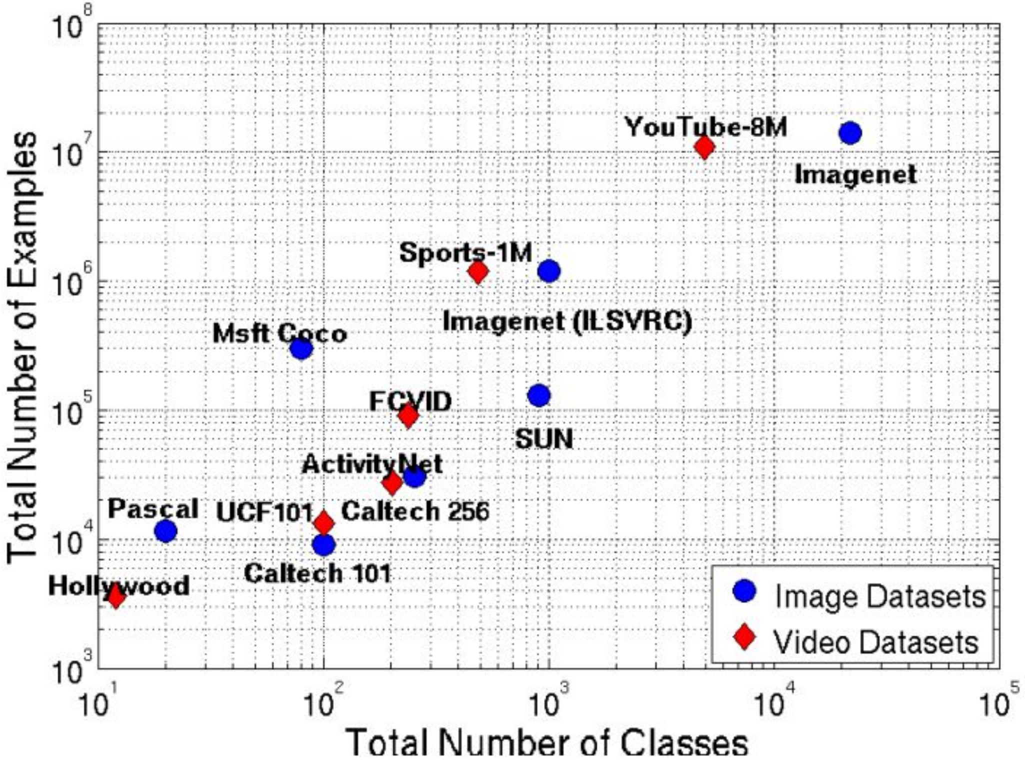
Used Unused

Campos et al., Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks, ICLR 2018

Outline

1. Self-supervision from videos
2. Architectures for video analysis
3. Exploiting redundancy in videos
4. **Tips and tricks for applying deep learning to video**

Activity Recognition: Datasets



Abu-El-Haija et al., Youtube-8m: A large-scale video classification benchmark, arxiv 2016

Computational burden

- The reference dataset for image classification, ImageNet, has ~1.3M images
  - Training a state of the art CNN can take up to 2 weeks on a single GPU
- Now imagine that we have an 'ImageNet' of 1.3M videos
  - Assuming videos of 30s at 24fps, we have 936M frames
  - This is 720x ImageNet!
- Videos exhibit a large redundancy in time
  - We can reduce the frame rate without losing too much information

Memory issues

- Current GPUs can fit batches of 32~64 images when training state of the art CNNs
  - This means 32~64 video frames at once
- Memory footprint can be reduced in different ways if a pre-trained CNN model is used
  - Freezing some of the lower layers, reducing the memory impact of backprop
  - Extracting frame-level features and training a model on top of it (e.g. RNN on top of CNN features). This is equivalent to freezing the whole architecture, but the CNN part needs to be computed only once.

I/O bottleneck

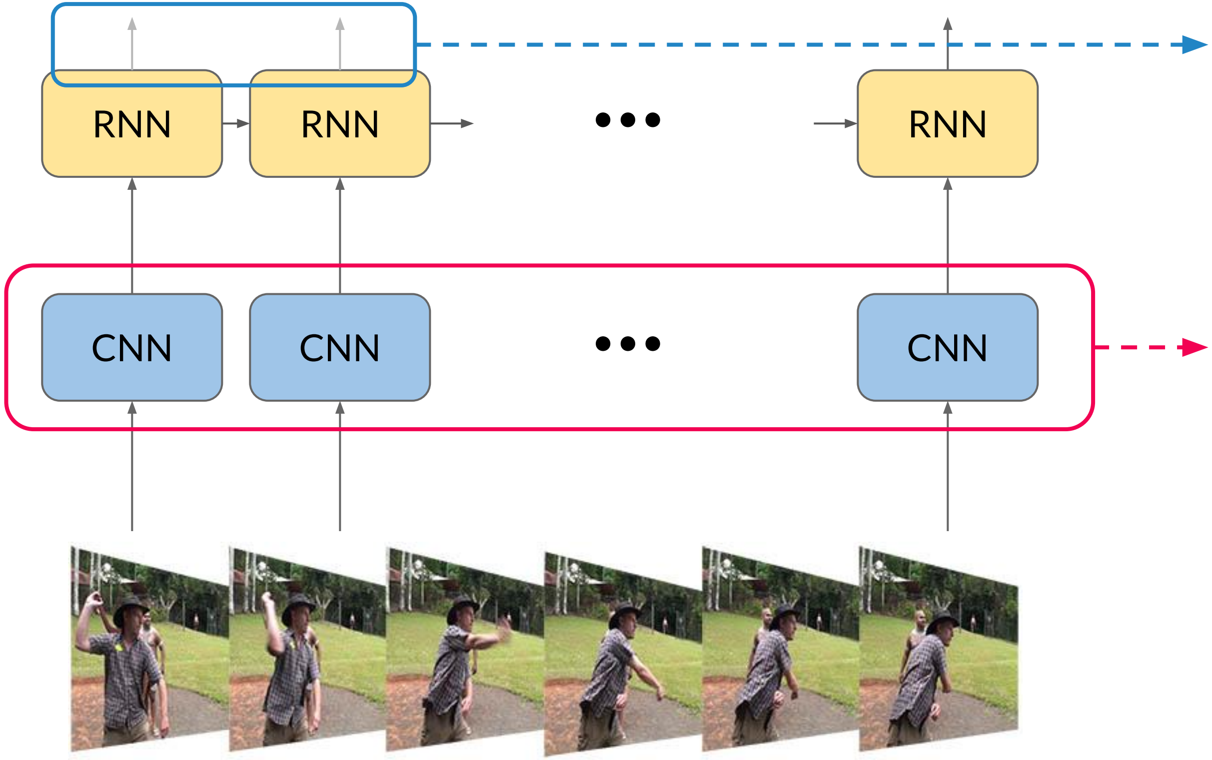
- In practice, applying deep learning to video analysis requires from multi-GPU or distributed settings
- In such settings it is very important to avoid *starving* the GPUs or we will not obtain any speedup
    - The next batch needs to be loaded and preprocessed to keep the GPU as busy as possible
    - Using asynchronous data loading pipelines is a key factor
    - Loading individual files is slow due to the introduced overhead, so using other formats such as TFRecord/HDF5/LMDB is highly recommended

Outline

1. Self-supervision from videos
2. Architectures for video analysis
3. Exploiting redundancy in videos
4. Tips and tricks for applying deep learning to video


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BÏYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
CHALTU
NUHUN
SPASSIBO
SNACHALHUYA
YUSPAGARÄTAM
SUKSAMA
WABEEJA
MAITEKA
HUI
EKHMET
DHANYABAAD
ANIHA
MERSI
SPASIBO
DENKAUJA
NENACHALHYA
UNALCHÉESH
ATTO
GUI
HATUR
EKOJU
SIKOMO
MAKETAI
MAAKE
GRAZIE
MEHRBANI
KOMAPSUMNIDA
SANCO
MERASTAWHY
GAEJTHO
LAH
PALDIES
TAVTAPUCH
MEDAWAGSE
GOZAIMASHITA
EFCHARISTO
AGUYJE
FAKAAUE
BAIIKA
JUSPAXAR
BOLZÏN
MERCI
MINMONCHAR

CNN+RNN: redundancy



Frame-level predictions are optional and depend on the task

~99% of the computational burden is here