Temporal Neural Networks

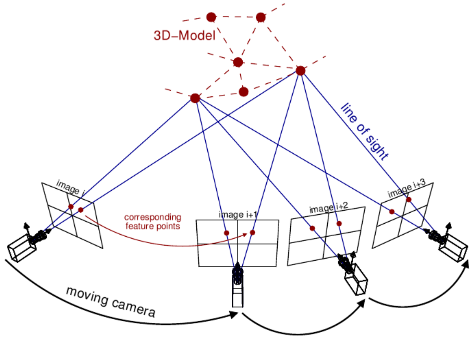
Lecture 12

Automatic Image Analysis

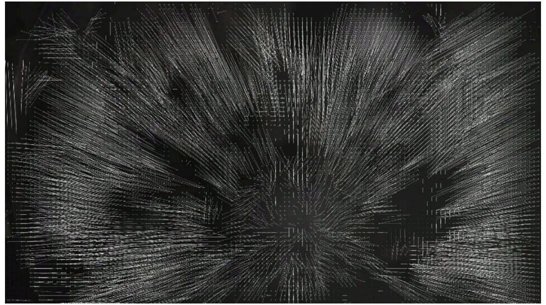
June 29, 2021

Technische Universität Berlin

Why should we analyze Videos?

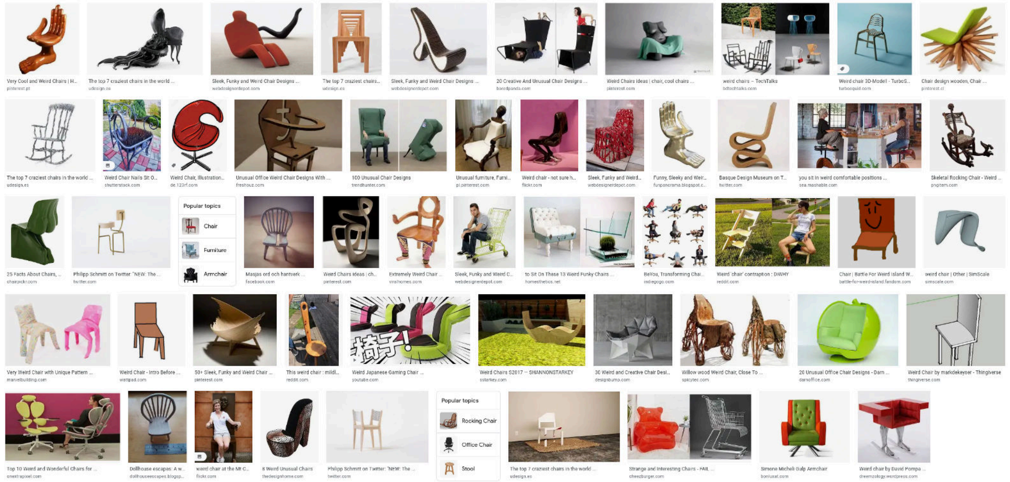


-
- Image from http://theia-sfm.org/



- Images from https://de.wikipedia.org/wiki/Optischer_Fluss and https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html

- Image from Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Carreira & Zissermann, NeurIPS 2014
-



- Google image search for 'weird chairs'.
-

- Action classification, Action detection
- Video captioning
- Object localization (position + orientation + dynamics)
- Forecasting

- UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild, Soomro et al., CRCV 2012

- 13320 videos (YouTube)
- 101 action categories

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

- 1,133,157 videos
- 487 sports classes

- 80 atomic visual actions
- 430 15-minute movie clips
- 1.62M action labels (bounding boxes in space and time)

- Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding, Gunnar et al., ECCV 2016
- AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions, Gu et al., CVPR 2018

- A Short Note on the Kinetics-700-2020 Human Action Dataset, Smaira et al., 2020

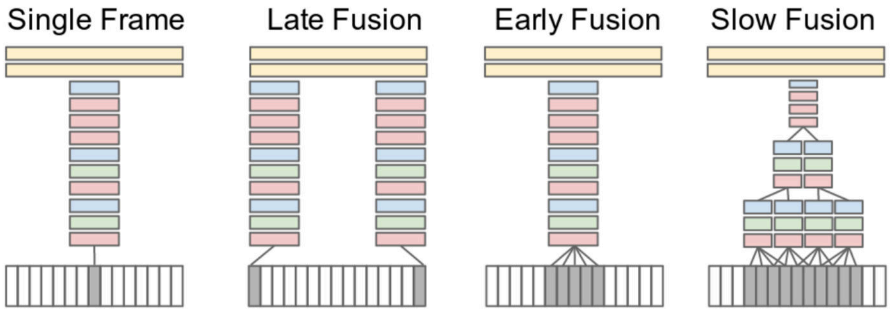
- 650000 video clips
- 400/600/700 action classes

What are the challenges?

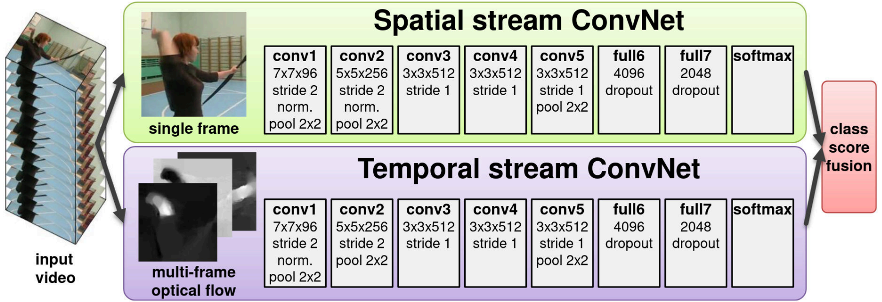
What are the challenges?

- More data, higher redundancy
- Lower quality (resolution, motion)
- Higher variance

How could we analyze Videos?



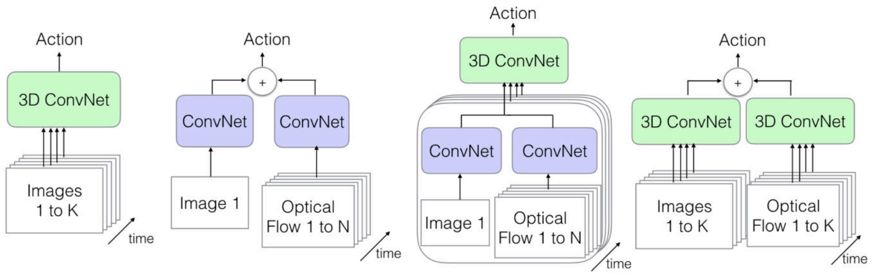
- Information along the time domain can be integrated on different levels of abstraction.
- Image from Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al, CVPR 2014



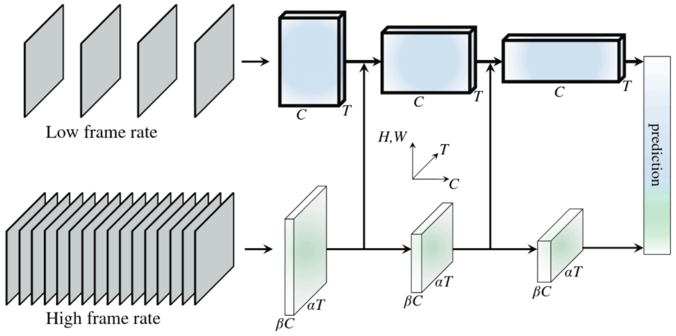
- Inspired by the two-stream hypothesis (dorsal stream: where, ventral stream what) https://en.wikipedia.org/wiki/Two-streams_hypothesis
- Image from Two-Stream Convolutional Networks for Action Recognition in Videos, Simonyan & Zissermann, NeurIPS 2014

- Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Carreira & Zissermann, CVPR 2017

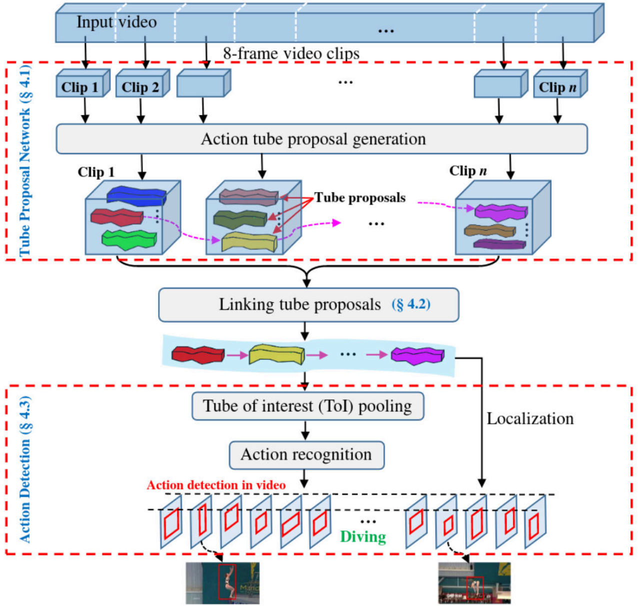
- Build 3D Convolutional Neural Networks based on well known architectures
- Initialize weights with networks pre-trained on ImageNet
- Replicate weights as if network is applied to boring video
  (sequence of duplicates of single frame)
- Striding and pooling have to be adjusted for time domain



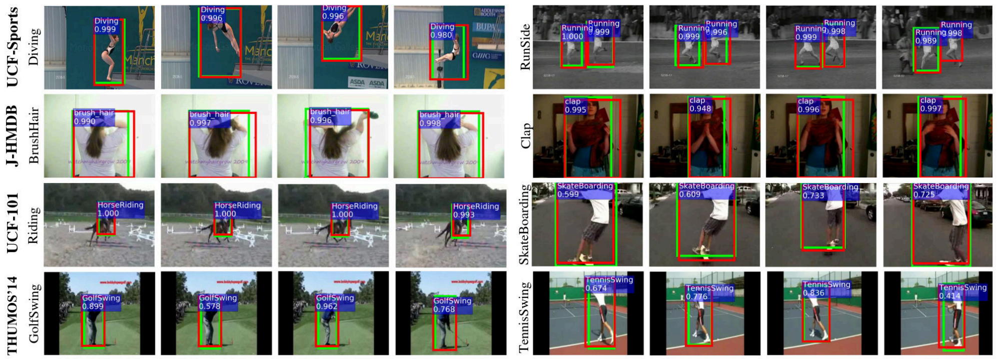
- Even with 3d convolutional networks a second steam based on optical flow improves the results.
- Image from Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Carreira & Zissermann, CVPR 2017



- Inspired by the retinal ganglion cells.
- 80% of computation for low frame rate but high spatial resolution
- 20% of computation for high temporal resolution but less spatial detail and lower dimensionality (channels)
- SlowFast Networks for Video Recognition, Feichtenhofer et al., ICCV 2019

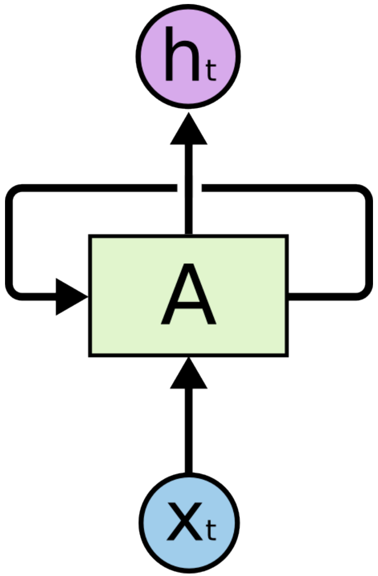


- Extending Faster R-CNN to the time domain for action localization.
- Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos, Hou et al., ICCV 2017

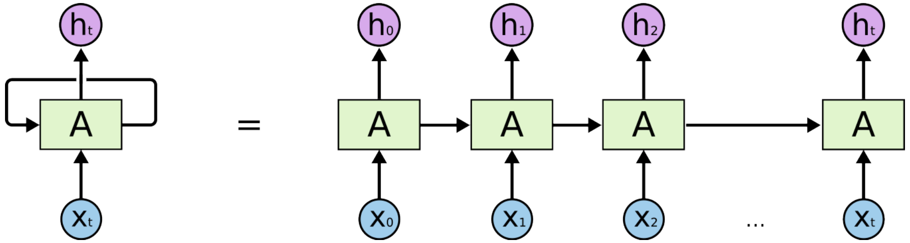


- Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos, Hou et al., ICCV 2017

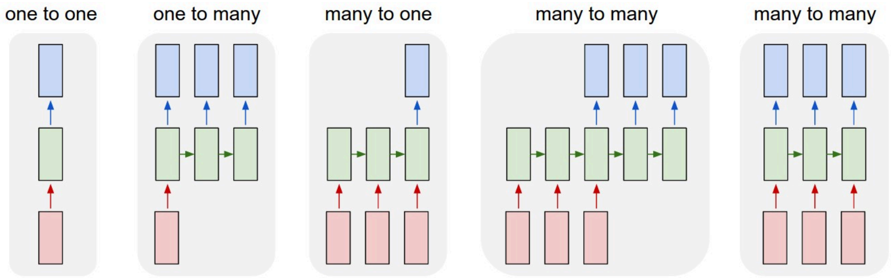
- So far we modeled time/sequences with feed forward networks
- What if we want to have long input sequences?
- What if the interpretation of the next input is dependent on the previous input?



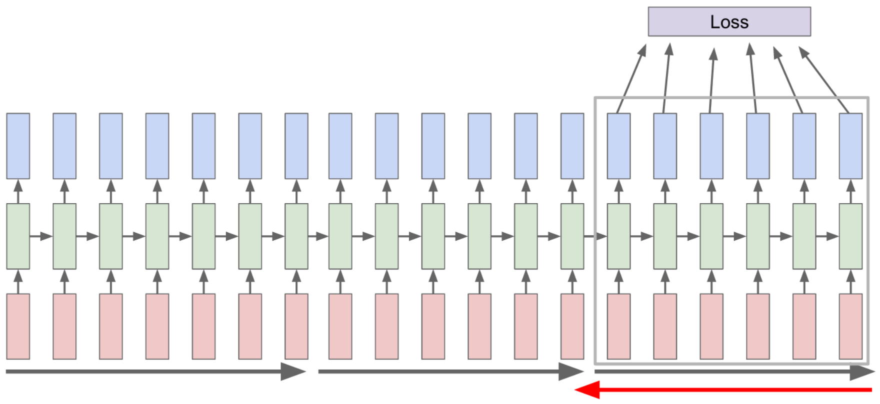
- A recurrent neural network is a network with a loop.
- Image from Understanding LSTM Networks, Chris Olah
  https://colah.github.io/posts/2015-08-Understanding-LSTMs/



-
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/



- Image from The Unreasonable Effectiveness of Recurrent Neural Networks, Andrej Karpathy
  https://karpathy.github.io/2015/05/21/rnn-effectiveness/



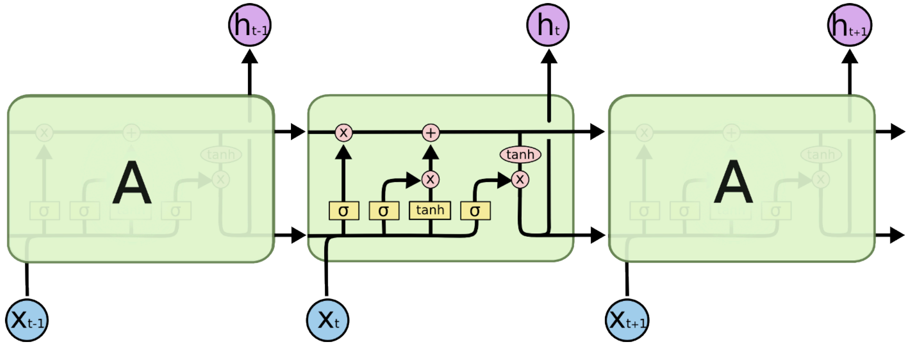
-
- Image from Stanford CS231n Lecture 10, Fei-Fei Li
  http://cs231n.stanford.edu/slides/2021/lecture_10.pdf

RNNs are cool because,

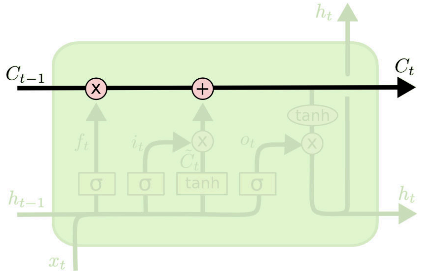
- they can process any length input,
- for processing input at t they can use information from $t - k$,
- model size does not increase with sequence length,

but

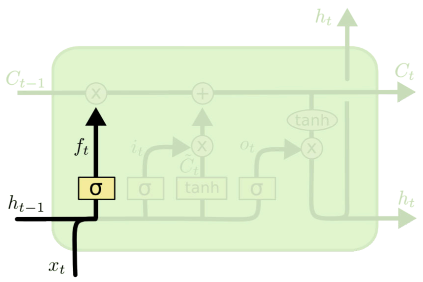
- What information should be saved in the state? For how long?
- Recursive term in gradient: vanishing/exploding gradients



- Long Short-Term Memory, Hochreiter & Schmidhuber, 1997
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/

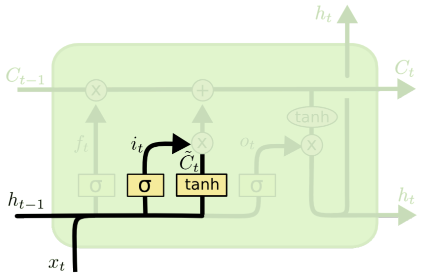


- LSTMs have a cell state, that allows to store information.
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/



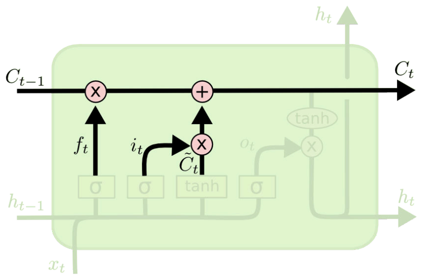
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

- The forget gate allows to delete content from the cell state.
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/



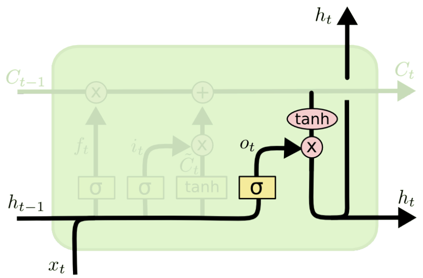
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

- The input gate decides which parts of a new candidate state are written to the cell state.
- Image from Understanding LSTM Networks, Chris Olah
  https://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

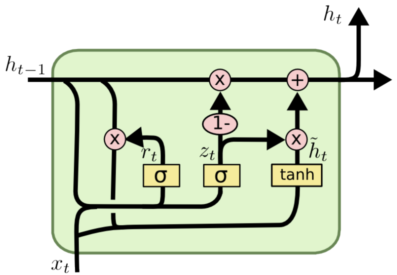
- These parts of the candidate state are than added to the cell state.
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

- The output gate decides which parts of the cell state are going to be the output state.
- Image from Understanding LSTM Networks, Chris Olah `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`



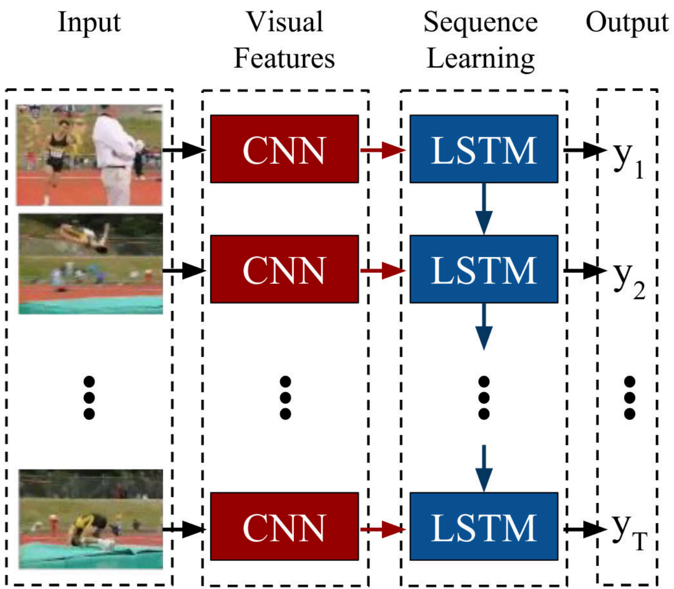
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

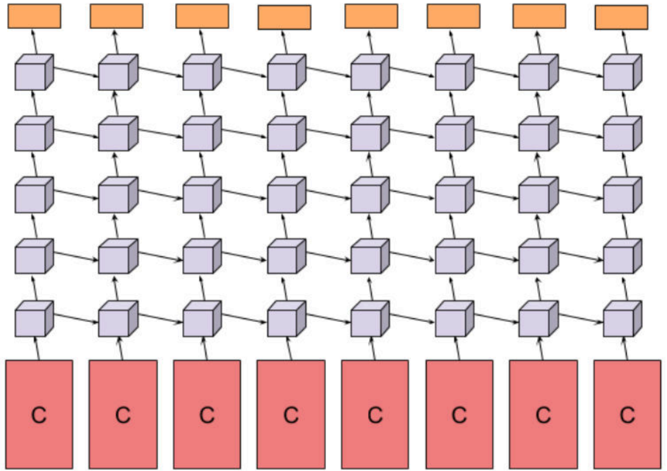
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

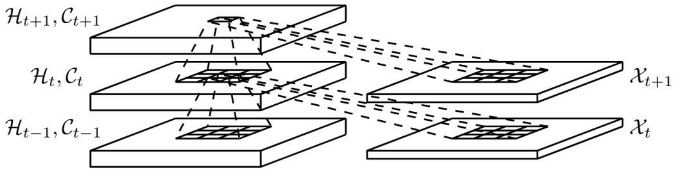
- Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, Cho et al., 2014
- GRUs combine the cell state and output and merge input and forget gate.
- Image from Understanding LSTM Networks, Chris Olah https://colah.github.io/posts/2015-08-Understanding-LSTMs/



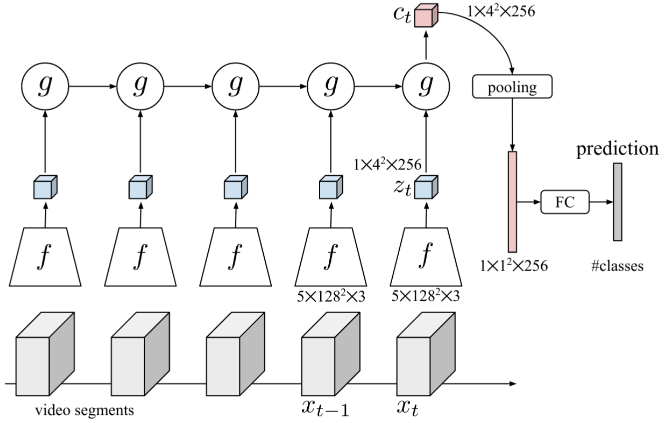
- Image from Long-term Recurrent Convolutional Networks forVisual Recognition and Description, Donahue et al., CVPR 2015



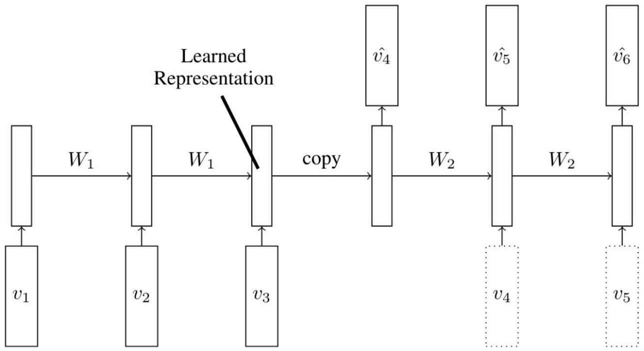
- Beyond Short Snippets: Deep Networks for Video Classification, Joe Yue-Hei Ng et al., CVPR 2015



-
- Image from Convolutional LSTM Network: A Machine LearningApproach for Precipitation Nowcasting, Shi et al., 2015



-
- Video Representation Learning by Dense Predictive Coding, Han et al., ICCV 2019



- Similar as in static images can be used for representation learning, video synthesis, style transfer, ...
- Image from Unsupervised Learning of Video Representations using LSTMs, Srivastava et al., 2015

- Image from https://commons.wikimedia.org/wiki/File:Coin_Toss_(3635981474).jpg
-

- Probabilistic modeling
- Adversarial training

- Image from https://commons.wikimedia.org/wiki/File:Coin_Toss_(3635981474).jpg
-