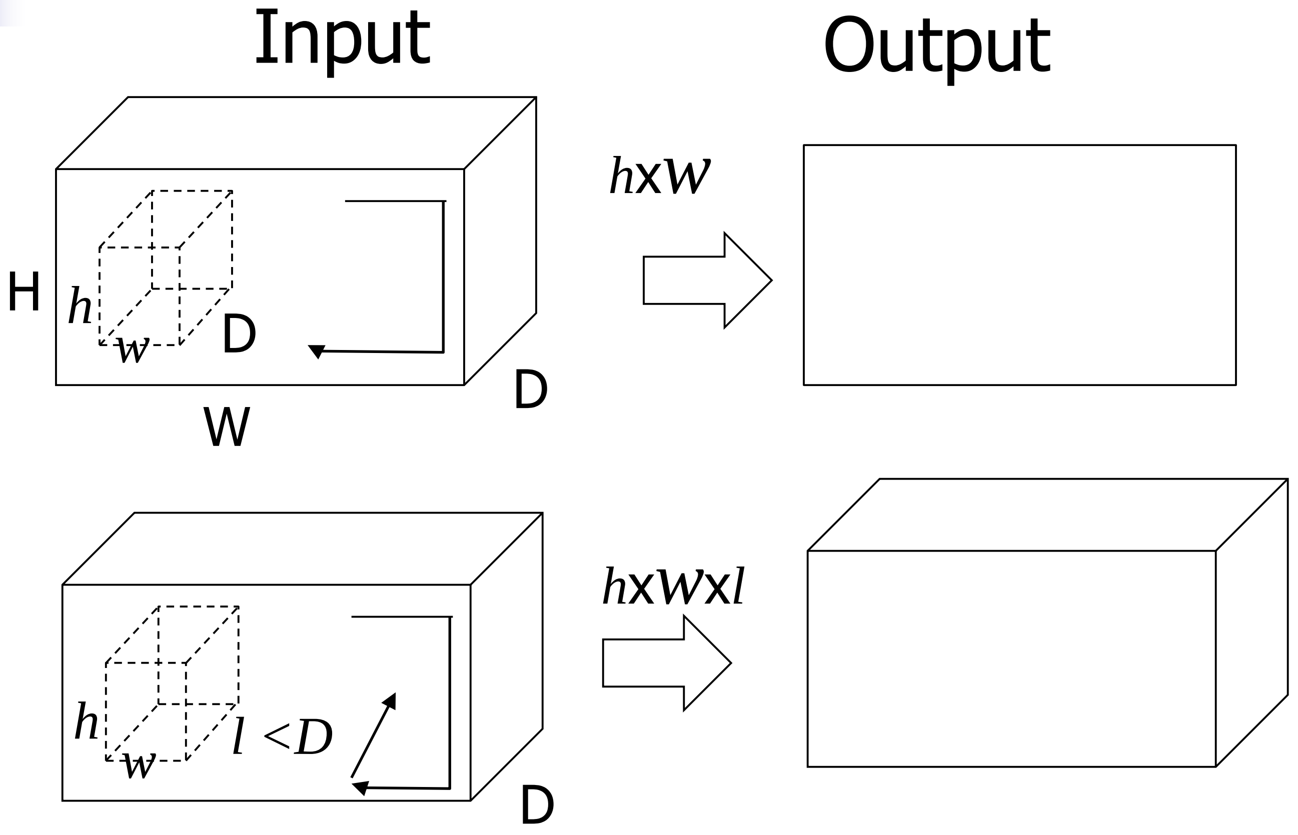
Deep Visual Learning from Videos

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Spatio-temporal base models

- 3-D Convnet
- Recurrent Neural Network
- Long Short Term Memory (LSTM) Network

3-D Convolution Layer



2-D Convolution Layer

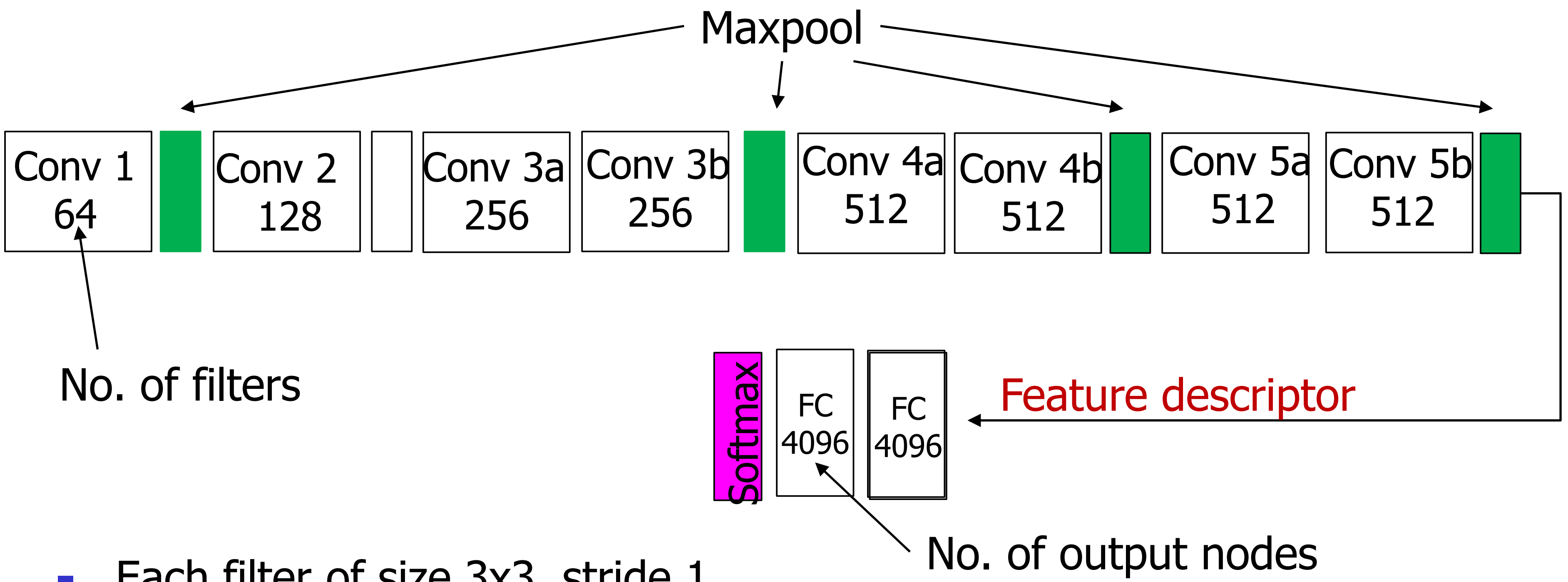
3-D Convolution Layer

Tran et al, Learning Spatiotemporal Features with 3D Convolutional Networks, 2015 IEEE International Conference on Computer Vision, 4489-4497

3-D Convnet

- Process a group of pictures at a time
- Extension of 2-D convnet
  - Kernel moves along width, height and depth
  - Each output channel is a 3-D image
  - 4-D kernel specification for channel aggregation
    - Similar to depth aggregation in 2-D convnet

C3D



- Each filter of size 3x3, stride 1
- Maxpool mask: the first one: 1x2x2 and others: 2x2x2
- Feature descriptor may be normalized for using with other classifier such as SVM

Tran et al, Learning Spatiotemporal Features with 3D Convolutional Networks, 2015 IEEE International Conference on Computer Vision, 4489-4497

Determining Sports Categories

- Dataset: Sports 1M
  - 1.1 Million Video data, 487 Categories
- Training
  - Random extraction of 5 clips of 2 seconds.
  - Frame resized 128x171
  - Randomly crop clips to 16x112x112
  - Augment data with horizontal flipping of frames with 50% prob.
  - Use of SGD for optimization, minibatch size: 30 and learning rate: .0003
    - with 1.9 M iterations.

Accuracy:
Top-1 Video hit: 60%
Top 5 video hit: 84%

Tran et al, Learning Spatiotemporal Features with 3D Convolutional Networks, 2015 IEEE International Conference on Computer Vision, 4489-4497

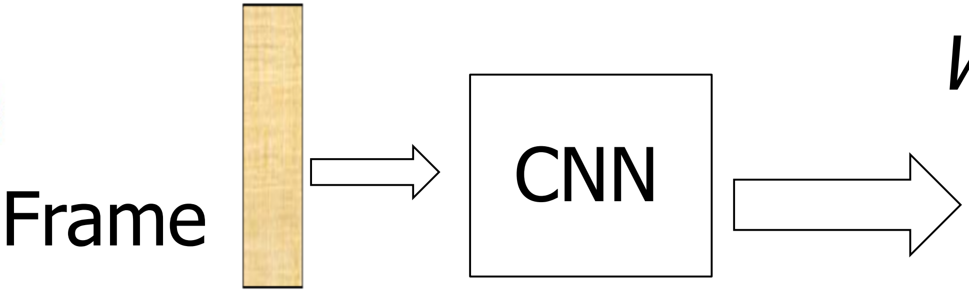
Recurrent Neural Networks

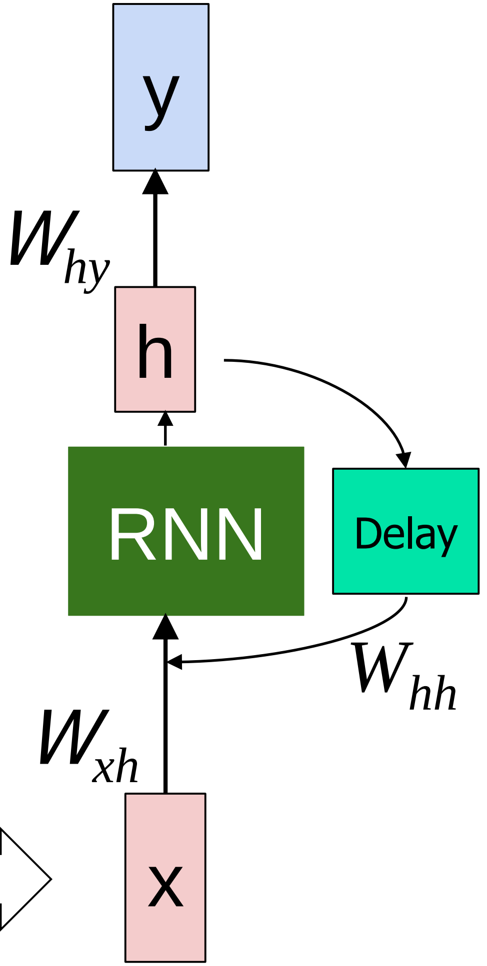
- Processes a sequence of vectors **x** by applying a recurrence formula at every time step.
  - usually for prediction of a vector at some time steps
  - Representing spatio-temporal context

$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

$$y_t = W_{hy} h_t$$

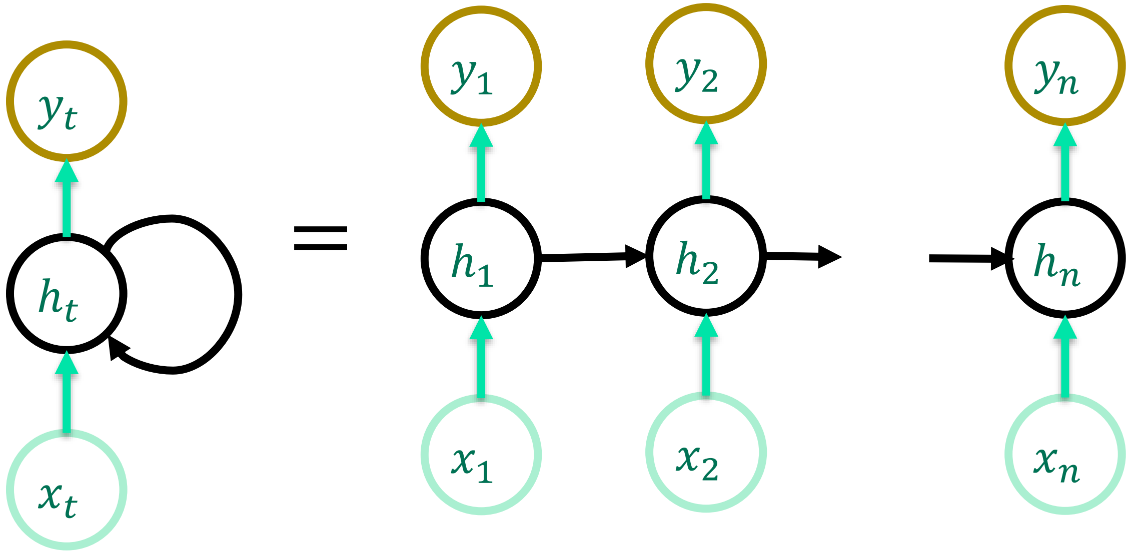




RNN for sequence modelling

Capture the dynamics of sequences via recurrent connections.

Recurrence → Memory



Equivalent to unfolded feedforward network of infinite number of layers

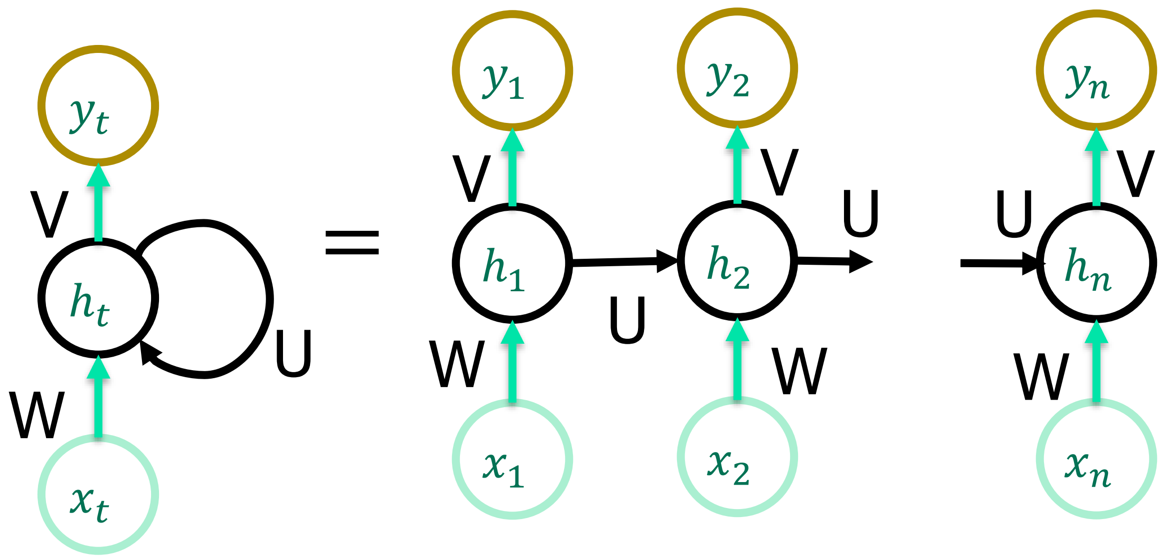
Unrolled in Time

Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

RNN for sequence modelling

Capture the dynamics of sequences via recurrent connections.

Recurrence → Memory



Unrolled in Time

$$\boxed{h_t = \mathrm{g}(\mathrm{U}h_{t-1} + \mathrm{W}\, x_t)}$$

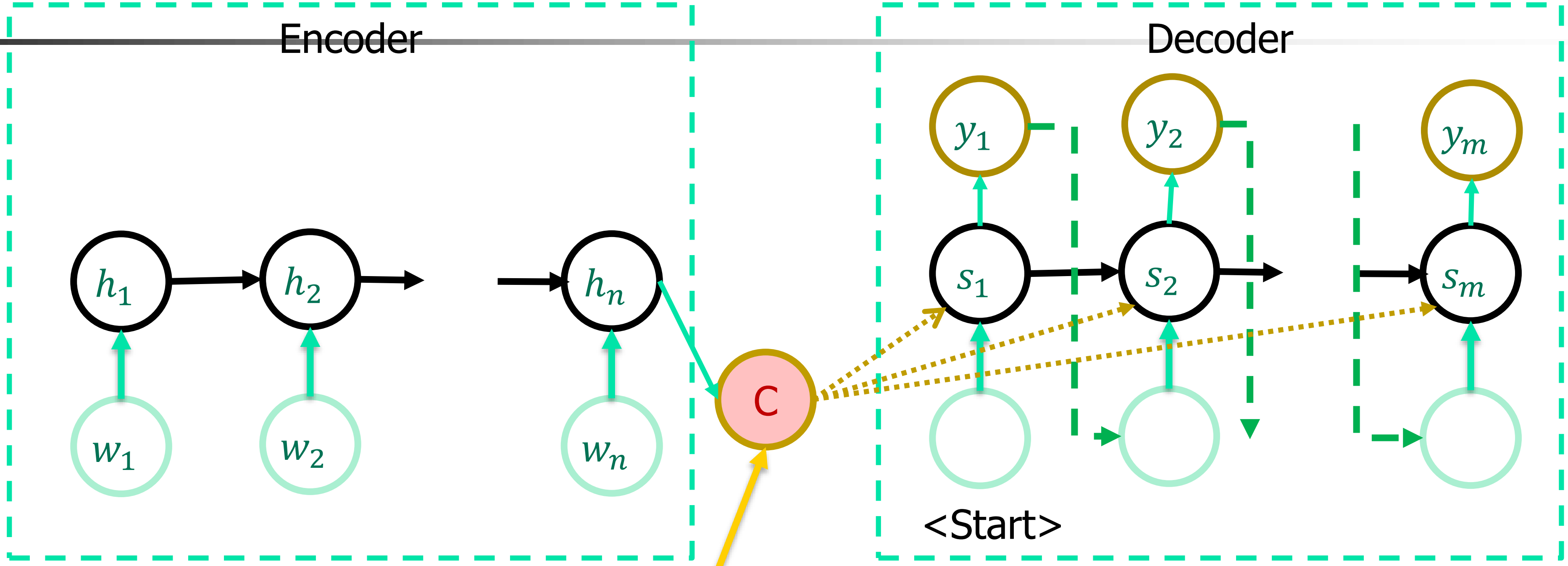
$$y_t = \mathrm{softmax}(\mathrm{V}h_t)$$

$$y_t = \mathrm{f}(\mathrm{V}h_t)$$

Recurrence → Memory
Memory fades

Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

Encoder Decoder with RNNs



$$s_t = g(\hat{y}_{t-1}, s_{t-1}, c)$$

C captures the entire information about the Input that is shared with the decoder

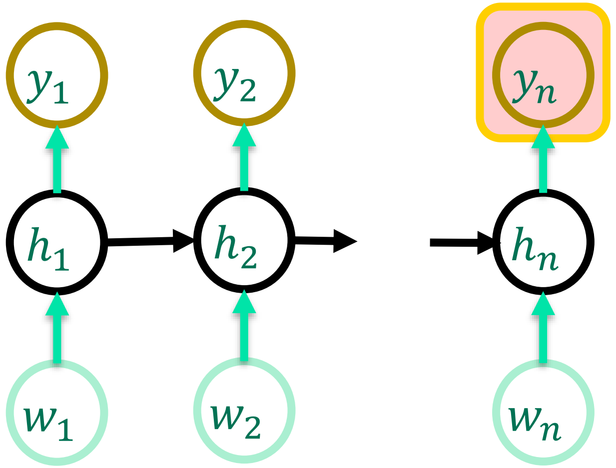
Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

Managing Context in RNNs: LSTMs

- Recurrence → Memory
- Memory fades
  - However, **long-distance information is often critical** to many language applications.

The flights the airline ***was*** *cancelling* ***were*** *full*.

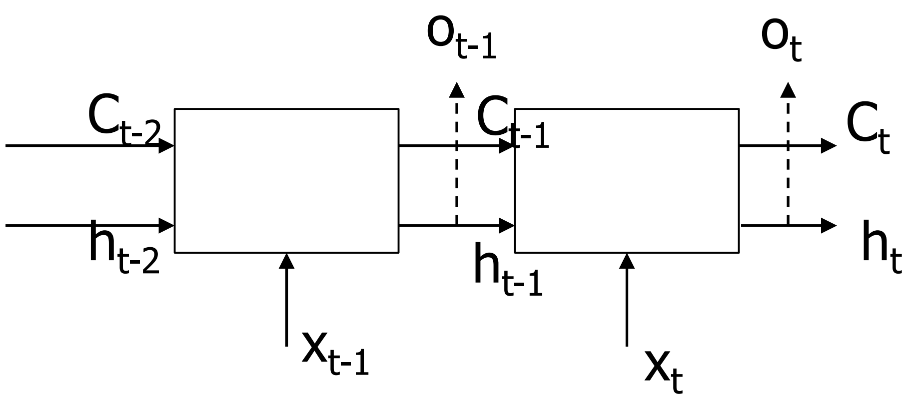
Fixed length vector



Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

Long short term memory (LSTM) networks

- RNN unable to learn long term dependency.
- LSTM proposed to avoid this problem.
- An LSTM node or cell maintains two states
- C_t : Long term memory state
- h_t : Short term memory state



Long short-term Memory (LSTM)

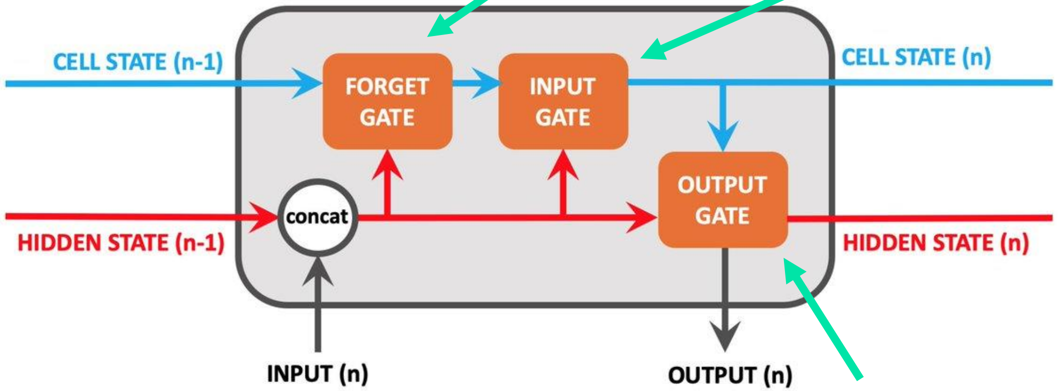
Gating for controlling what is “remembered”

- Hidden State: holds previous information (Short-term memory)
- Cell State: memory of the network (Long-term memory)

Memory cells capture long range dependencies

What should be forget in cell state

How much new data should be taken



How much does memory affect output

Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

LSTMs: learnable gates

Split the hidden layer into two vectors **c** and **h** and have three learnable gates

New cell content $g_t = \tanh(U_g h_{t-1} + W_g x_t)$

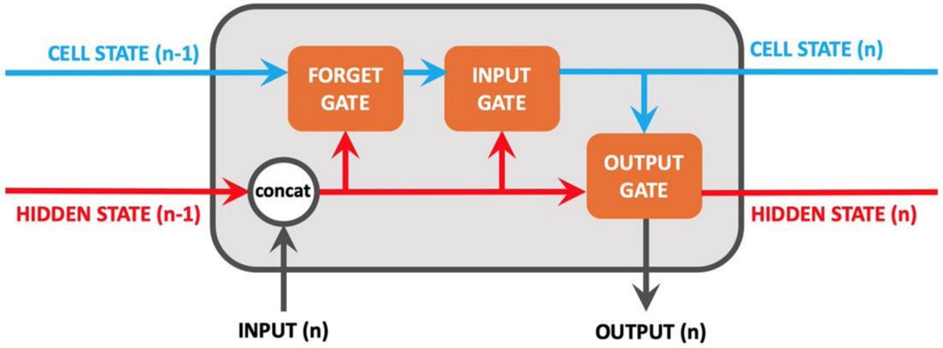
Input $i_t = \sigma(U_i h_{t-1} + W_i x_t)$

Forget $f_t = \sigma(U_f h_{t-1} + W_f x_t)$

Output $o_t = \sigma(U_o h_{t-1} + W_o x_t)$

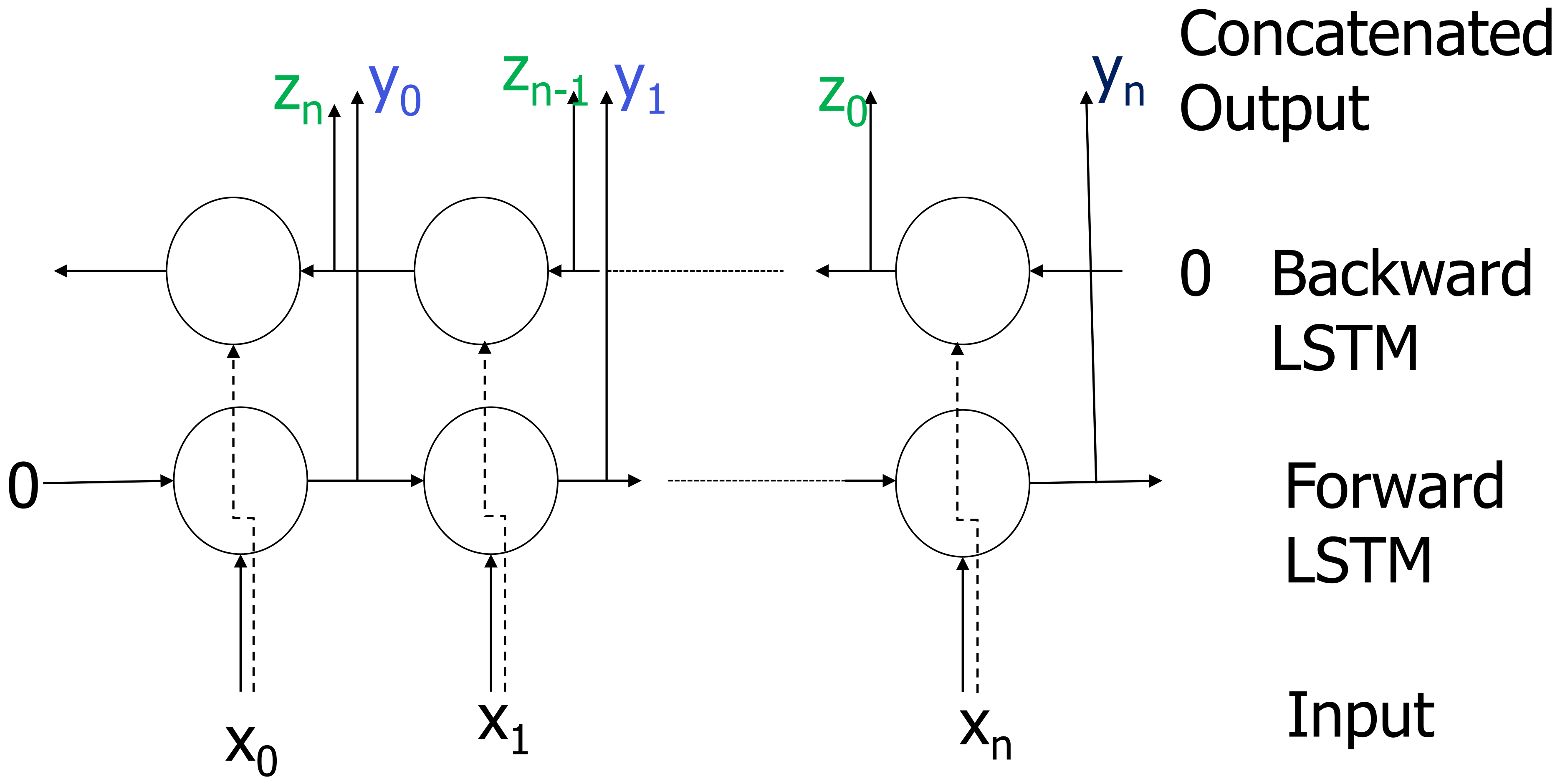
Cell state $c_t = f_t \odot c_{t-1} + i_t \odot g_t$

Hidden state $h_t = o_t \odot \tanh(c_t)$



Courtesy: Prof. Sudeshna Sarkar, CSE, IITKGP

Bi-directional LSTM



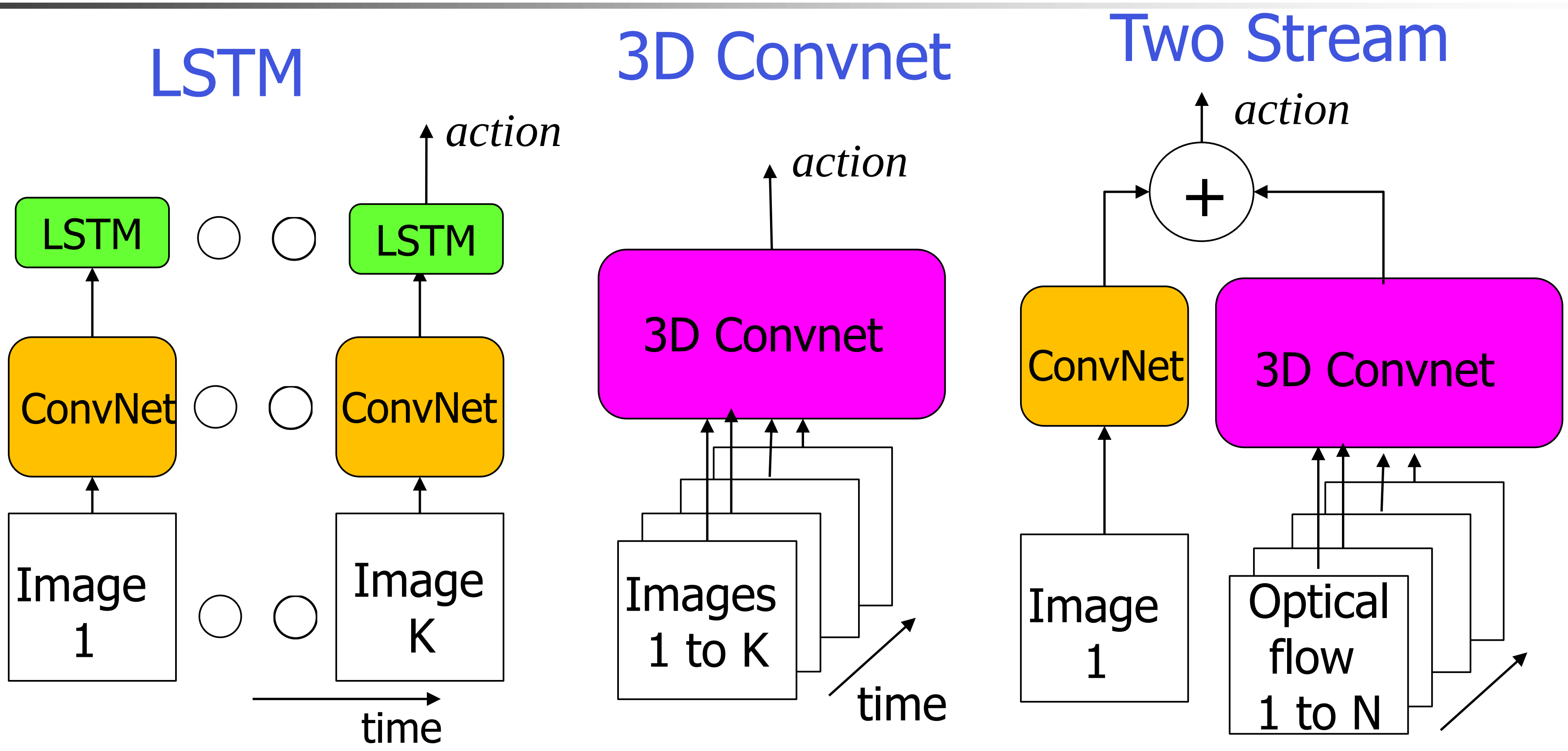
For convenience long-term and short-term states not shown explicitly.

Human Action recognition: An example of video processing

- To recognize human action from a video clip.
- Dataset used to train the model
  - Kinetics Human Action Video Dataset
  - https://github.com/cvdfoundation/kinetics-dataset
    - 650,000 video clips that cover 400/600/700 human action classes, depending on the dataset version
    - include human-object interactions such as playing instruments, as well as human-human interactions such as shaking hands.
    - manually annotated with a single action class lasting around 10 seconds.

Carreira and Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, IEEE Conference on Computer Vision and Pattern Recognition,4724-4733, 2017

Video descriptor architectures



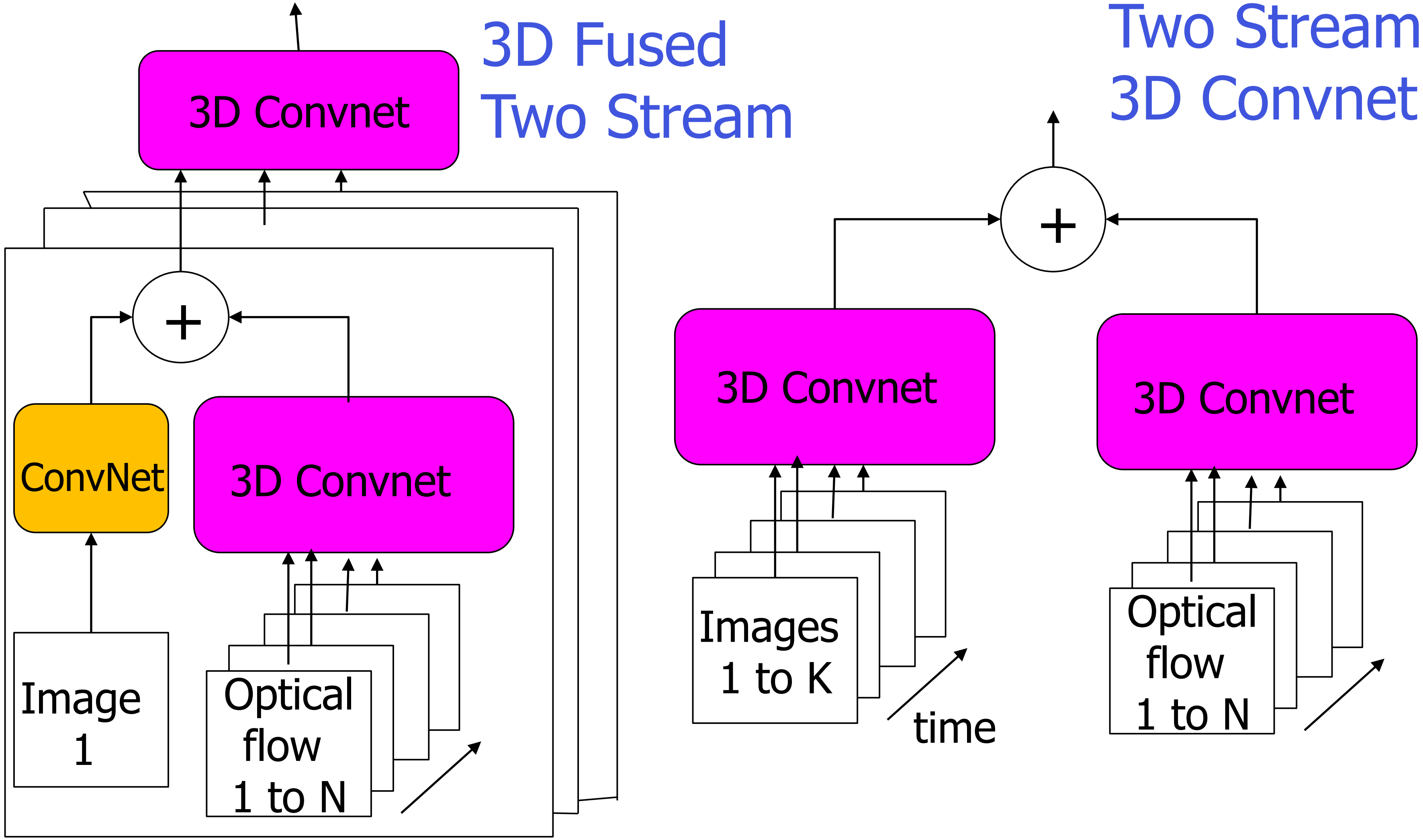
K: Number of frames in a video

N: N frames in a sequence

Carreira and Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, IEEE Conference on Computer Vision and Pattern Recognition,4724-4733, 2017

Carreira and Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, IEEE Conference on Computer Vision and Pattern Recognition,4724-4733, 2017

Video descriptor architectures



Carreira and Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, IEEE Conference on Computer Vision and Pattern Recognition,4724-4733, 2017

Experimental Results

- Data: MiniKinetics (a subset)
  - 213 classes, 120 K clips
  - Training set: 150 – 1000 clips per class
  - Validation Set: 25 clips per class
  - Test Set: 75 clips per set

Architecture	RGB	Optical Flow	RGB + Opt. Flo.
LSTM	69.9	-	-
3D Convnet	60.0	-	-
Two stream	70.1	58.4	72.9
3D Fused Two Stream	71.4	61.0	74.0
Two Stream 3D Convnet	**74.1**	**69.6**	**78.7**

Learning object tracker

- Given a target object X in the starting frame compute its location in subsequent frames.
- Two tasks in tandem
  - Classification of a candidate window in the next frame.
    - Discriminative tracking
  - Regress the bounding box w.r.t. the location of the candidate.
    - Akin to region proposal networks.
- Generic pipeline
  - Learn feature representation
  - Learn classification (and / or) regression model
  - Set mechanism for Updating feature representation periodically

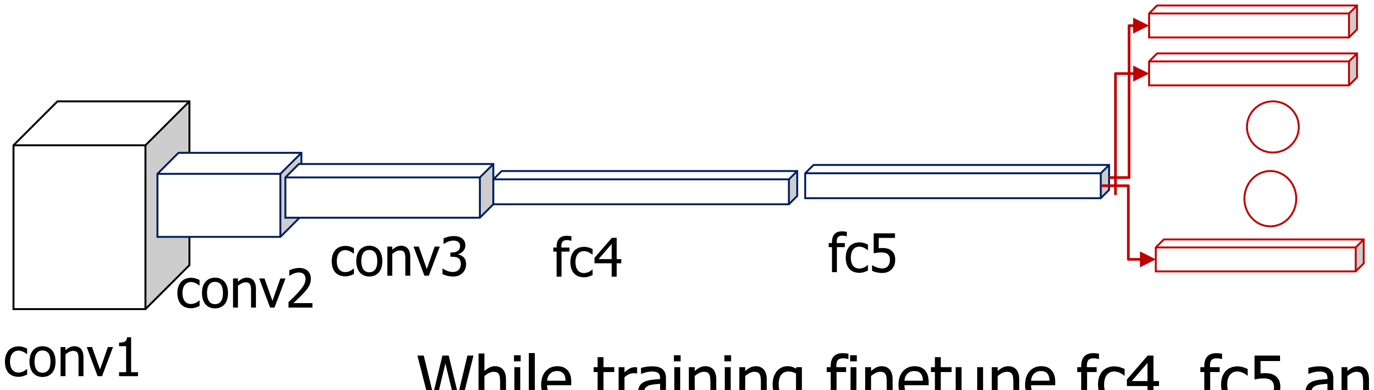
MDNet: Multi-Domain CNN based tracking

- A domain is specific to the tracking of a target.
  - Associated data defines the domain.
  - Multiple targets define multiple domains
- Learn a CNN with shared backbone and domain specific FC layers for classification and regression.
  - Positive (foreground) and negative (background) examples generated around the target
    - Data augmentation by translation and scaling
  - Train the backbone with specific fc layer and minibatches of a domain one after another.

Nam and Han, Learning Multi-Domain Convolutional Neural Networks for Visual Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 4293-4302, 2016

MDNet: Multi-Domain CNN based tracking

- Testing on a new target
  - Fine tune domain specific FC layers (fc6) with Augmented Data
  - Learn regression model in the first frame using Augmented Data and precise target definition
  - Fine tune FC layers periodically



While training finetune fc4, fc5 and specific fc6.

Nam and Han, Learning Multi-Domain Convolutional Neural Networks for Visual Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 4293-4302, 2016

Summary

- 3D image feature descriptor
  - 3D Convnet
- Sequence Modeling
  - RNN, LSTM
- Video descriptor architectures
  - LSTM
  - 3D Convnet
  - Two Stream (RGB + Optical Flow)
  - 3D fused Two stream
  - Two Stream 3D convnet
- Object tracker
  - MDNet

Thank you!