Deep Networks-based Video Classification Methods

A literature overview

Papers for discussion

- **Andrej Karpathy et al.**, Large-scale Video Classification with Convolutional Neural Networks
- **Jeff Donahue et al.**, Long-term Recurrent Convolutional Networks for Visual Recognition and Description
- **Joe Yue-Hei Ng et al.**, Beyond Short Snippets: Deep Networks for Video Classification

Comparison points

PAPER	KARPATHY ET AL.	DONAHUE ET AL.	NG ET AL.
Affiliation	STANFORD UNIVERSITY	UNIVERSITY OF CALIFORNIA, BERKELEY	GOOGLE DEEP MIND
inProceedings	CVPR* 2014	CVPR 2015	CVPR 2015
Architecture	CNN only	CNN + LSTM (Long-term recurrent convolutional nets (LRCNs))	CNN + Feature Pooling/LSTM
Input processing rate	Training: 5-20 fps Testing: 20 frames	Testing: 16 frames	Training: 1fps Testing: 30/120 frames
Accuracy	60.9% on Sports -1M 65.4% on UCF-101	82.92% on UCF-101	73.1% on Sports-1M 88.6% on UCF-101
Code open-sourced?	NO	YES (BVLC Caffe)	NO
Tidbit	Authors created the sports 1M dataset	Can also be used for image/video description	Work emanated from a Google internship
	All the primary authors are currently still Ph.D. students !		

Other video classification codes: Caffe C3D

Other 3D convolution libraries: conv3D (Theano), volumetric convolution (Torch)

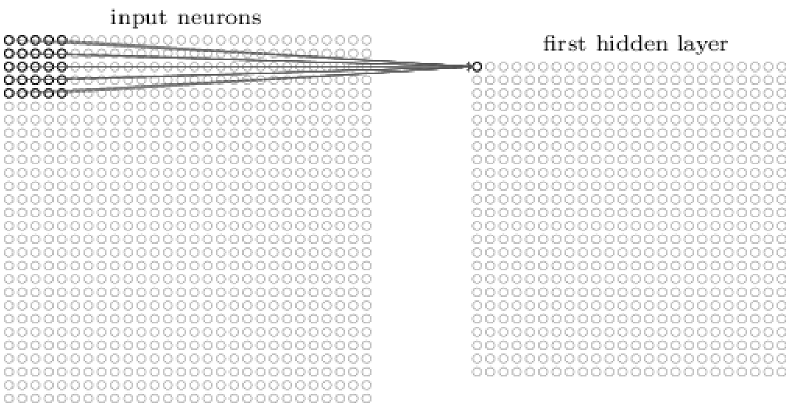
*Computer Vision and Pattern Recognition

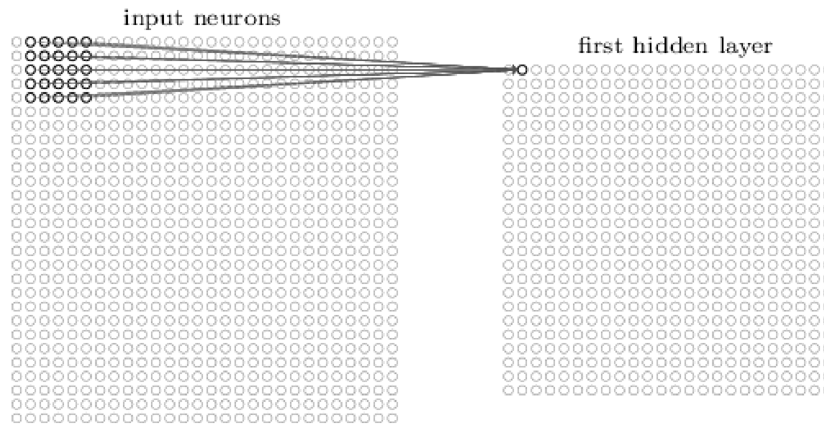
Popular video datasets

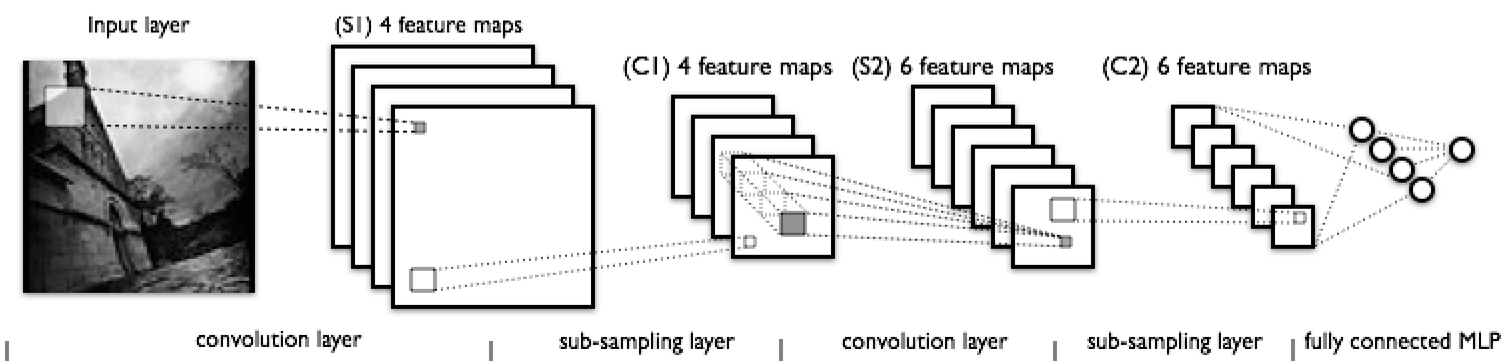
- **UCF 101**: 13320 videos in 101 classes, separated into 5 broad groups: Human-Object interaction, Body-Motion, Human-Human interaction, Playing Instruments and Sports
- **Sports 1M**: 1 million YouTube videos belonging to a taxonomy of 487 classes of sports; 1000-3000 videos per category.
- **CCV**: 9317 videos and 20 categories related to consumer video (wedding dance, basketball, graduation, birthday, etc)
- **UT-interaction**: continuous execution of 6 classes of human-human interaction; 20 video sequences each around 1 minute long – This is similar to the surveillance video we may capture
- **HMDB-51:** 7000 clips in 51 action classes related to human motion.

Popular image datasets: MNIST, CIFAR-10, CIFAR-100, ImageNet

Convolutional Neural Networks (CNNs)

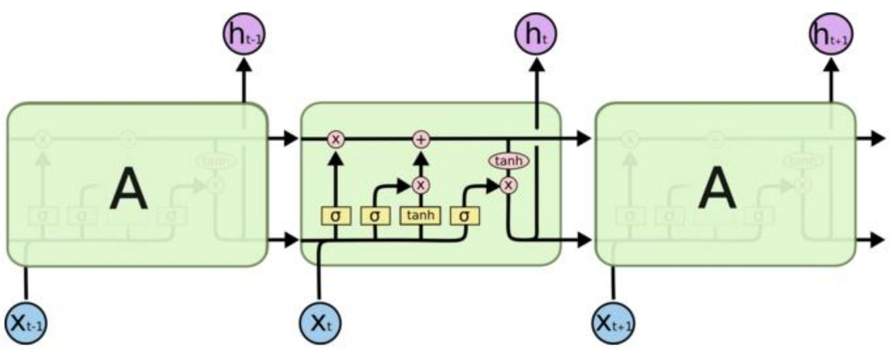


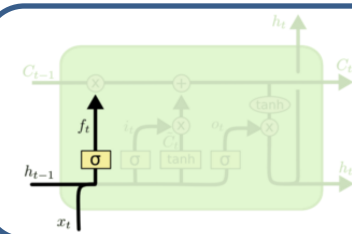




Long Short Term Memory (LSTM)

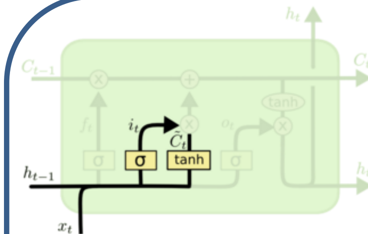
The LSTM is a recurrent neural network that uses memory cells to store, modify, and access internal state, allowing it to better discover long-range temporal relationships





Forget gate layer

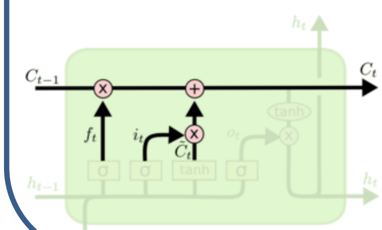
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



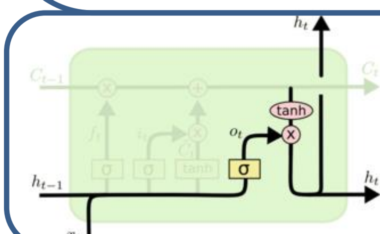
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Input gate layers



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



Output gate layer

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

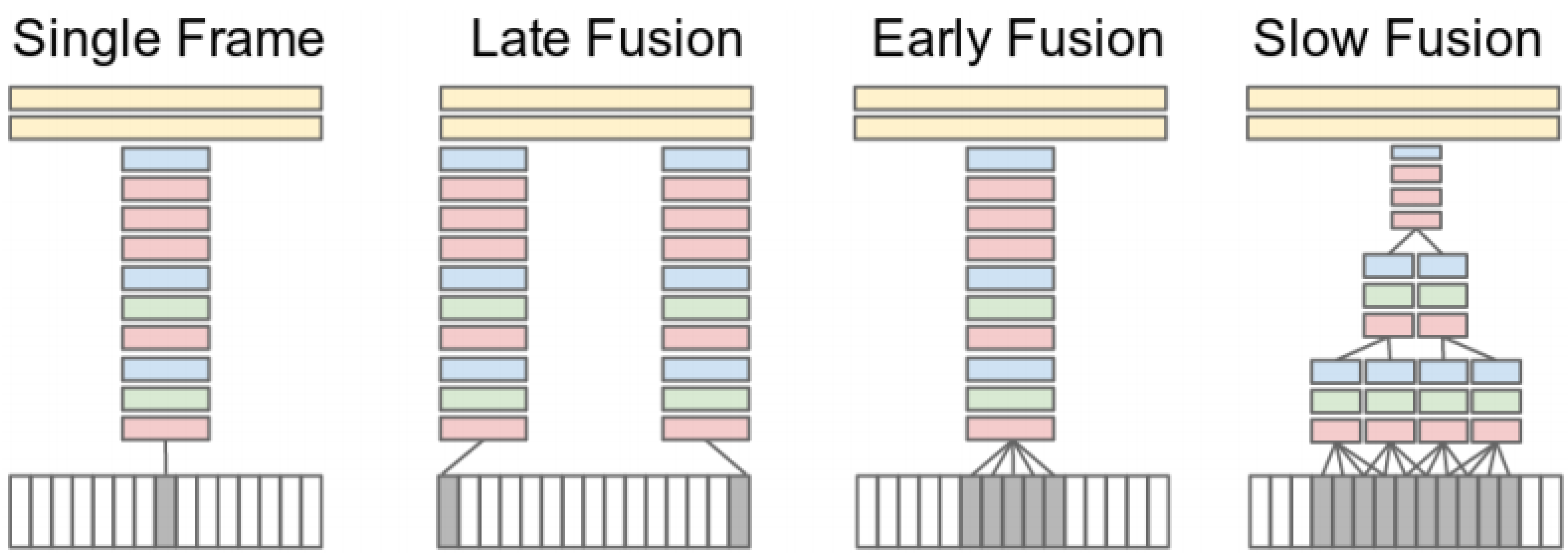
$$h_t = o_t * \tanh\left(C_t\right)$$

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

- Multi-resolution architecture for addressing computational efficiency
    1. Context stream – down-sample the frame at half the original spatial resolution
    2. Fovea stream – sample only the center portion of the video at full resolution

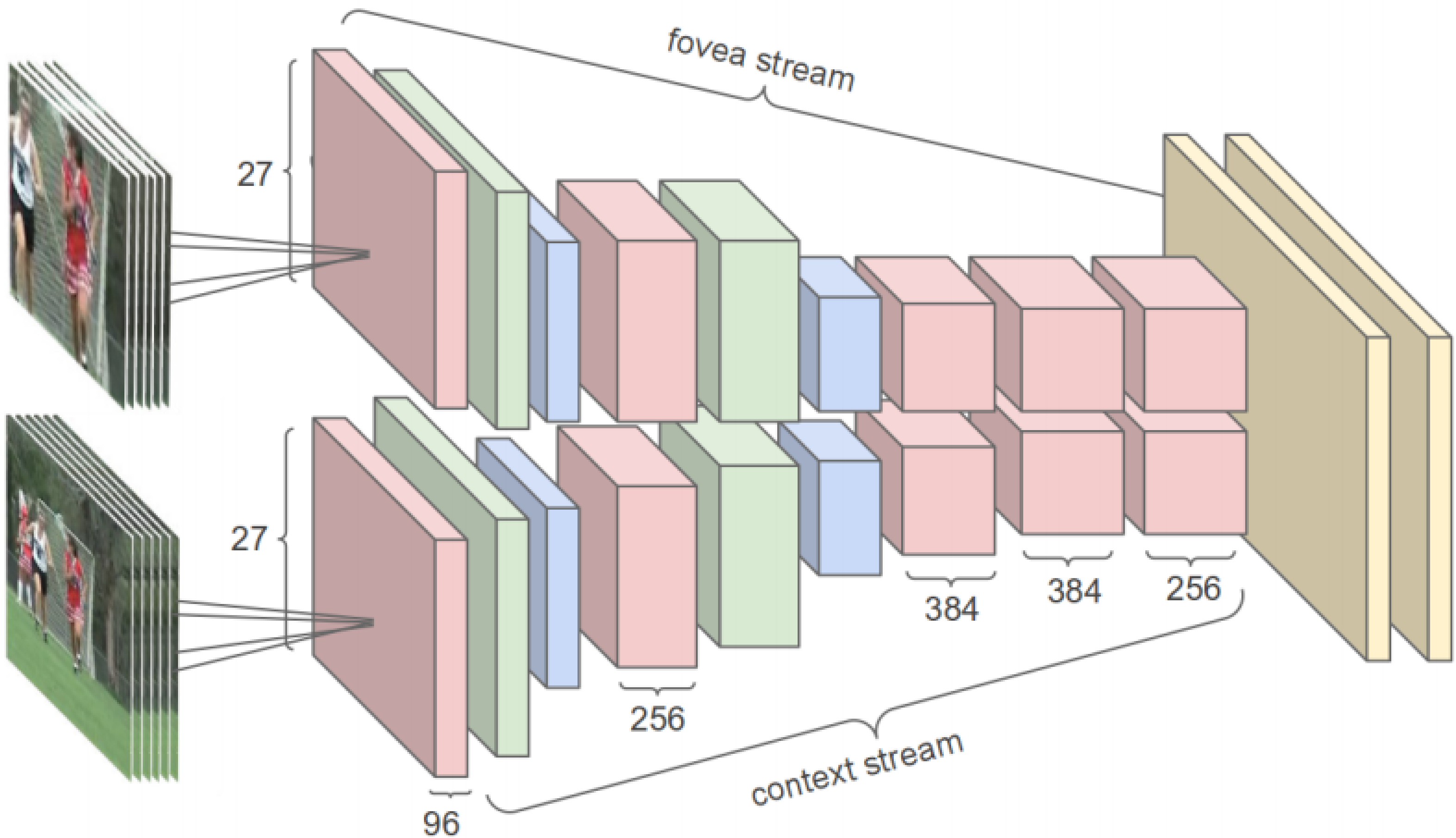
  Apparently, the fovea stream learns grayscale, high-frequency features while the context stream models lower frequencies and colors.
- Temporal information is handled via different time fusion techniques (late, early, slow)
- Video is chopped into 5 clips per second for full-resolution, and 20 clips per second for multi-resolution.
- Input image resolution is 170 x 170. They are randomly flipped horizontally with 50% probability.
- Training is performed on the Sports 1M-dataset (50 random frames per video); testing is done on both the Sports-1M as well as UCF-101 datasets. Training data labeling is automatically done based on the text metadata describing the video.

Methods for fusing information over the temporal domain



Red, green and blue layers indicate convolution, normalization and pooling layers respectively. There are also two yellow fully connected layers.

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



C(96, 11, 3)-N-P-C(256, 5, 1)-N-P-C(384, 3, 1)- C(384, 3, 1)-C(256, 3, 1)-P-FC(4096)-FC(4096)
C(d, f, s) indicates a convolutional layer with d filters of spatial size f ×f, applied to the input with stride s. Pooling is performed across 2x2 regions. Activations via rectified linear units (ReLUs).

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

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	**60.9**	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

Performance on the sports 1M dataset

- Single-frame techniques with hit@5 perform quite well!
- Slow fusion performs the best amongst temporal fusion methods

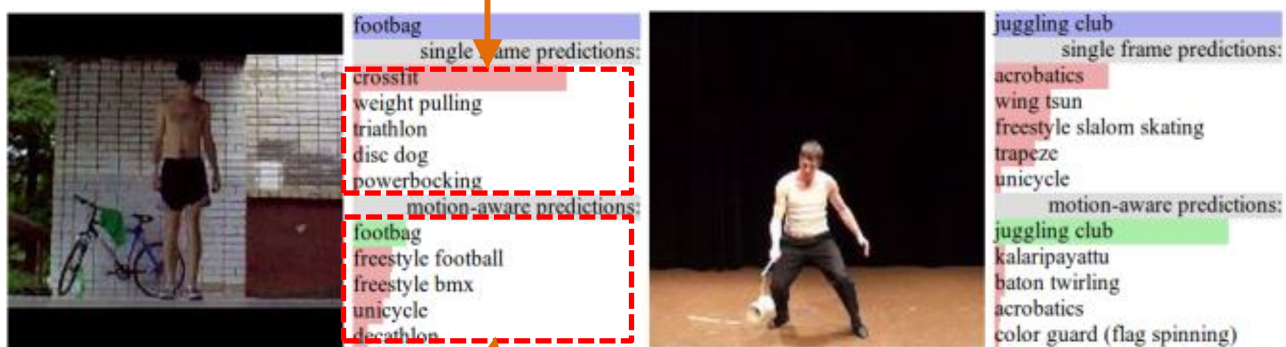
Model	3-fold Accuracy
Soomro et al [22]	43.9%
Feature Histograms + Neural Net	59.0%
Train from scratch	41.3%
Fine-tune top layer	64.1%
Fine-tune top 3 layers	**65.4%**
Fine-tune all layers	62.2%

Performance on the UCF-101 dataset using slow fusion

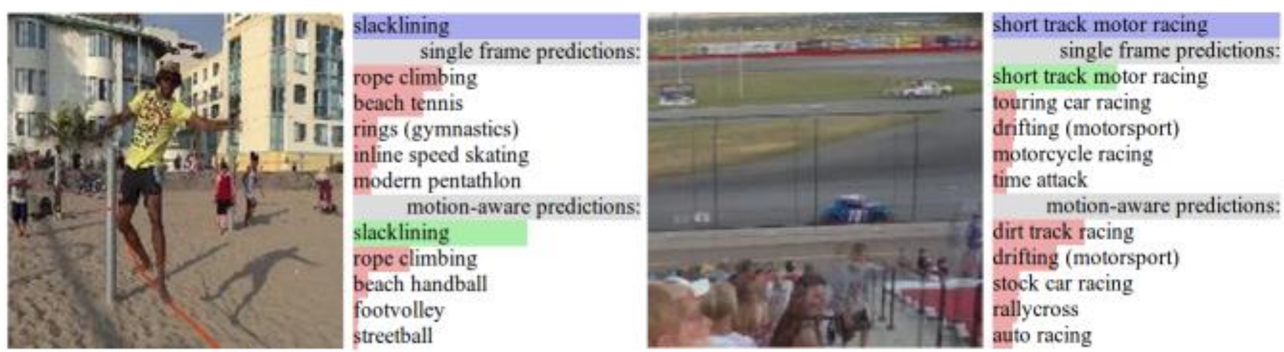
- Train from scratch fails to perform well, likely due to overfitting!
- Taking a balanced approach (fine-tuning the top 3 layers) helps the most in terms of performance
- 3-fold accuracy used for cross-validation

CNN for Single Frame vs Video

Single Frame Results (less accurate)



Motion Aware Results (more accurate)

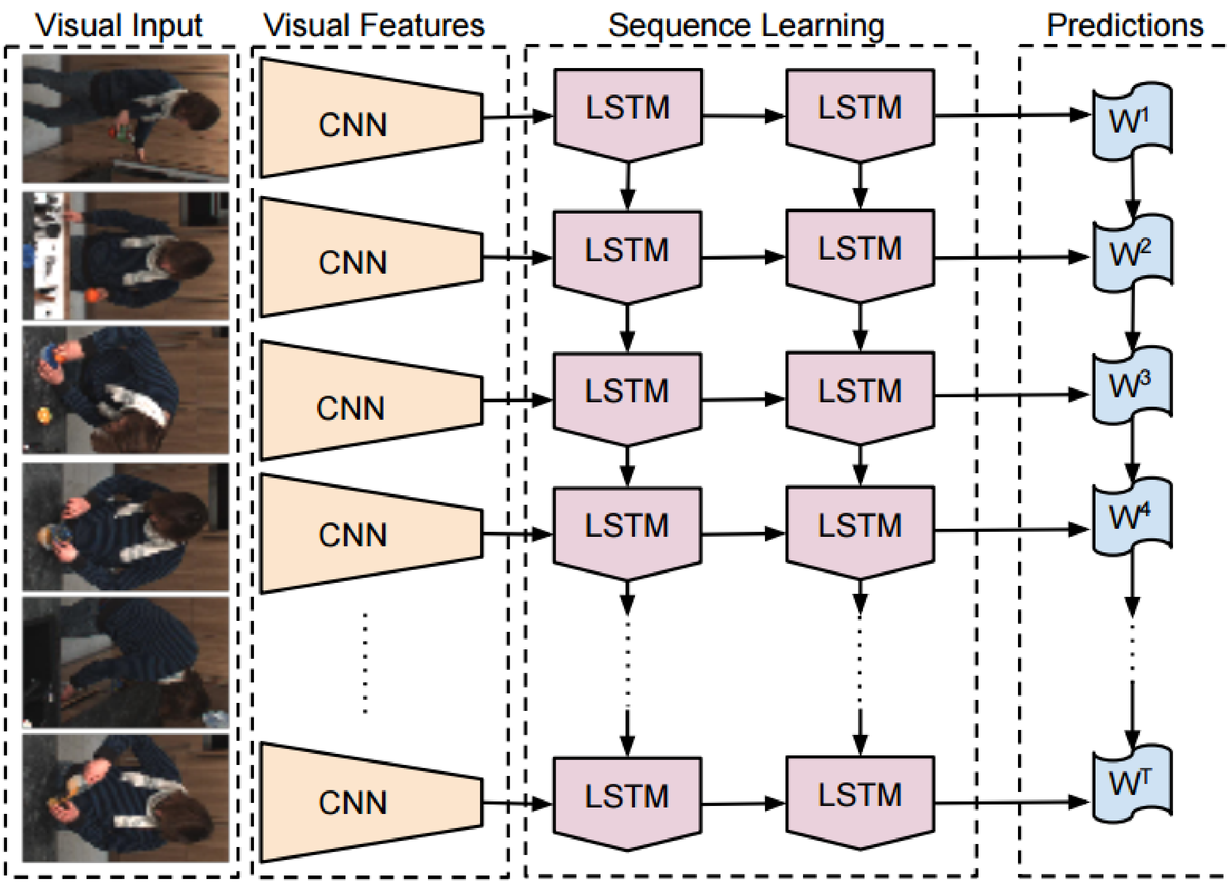


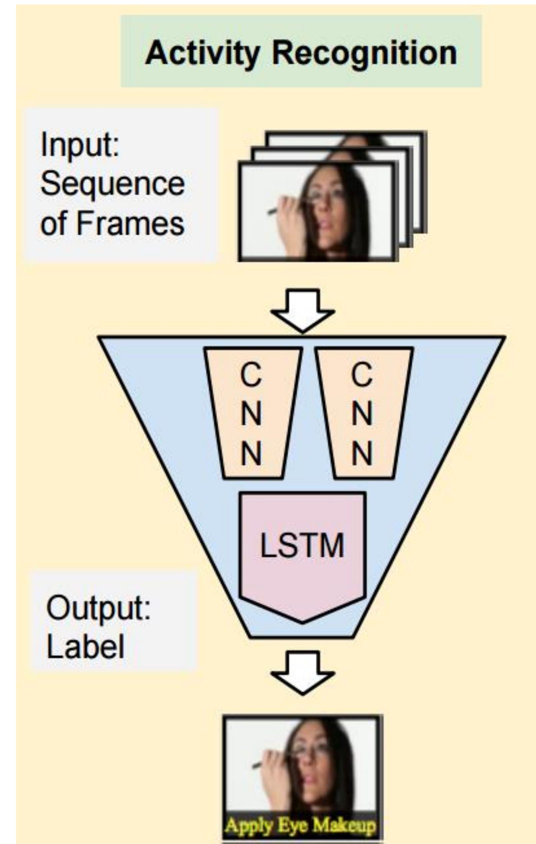
Green indicates correct tag, Top five tags shown in the order of reducing confidence

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

- The CNN used is a hybrid of three historic models, and is pre-trained on the 1.2M image ILSVRC-2012 dataset.
  - T frames are inputs to T CNNS (T=16 in implementation)
- LSTM: A single-layer LSTM with 256 hidden units.
- Two variants of LSTMs tried
  - LRCN – fc6: LSTM is placed after the first fully connected layer
  - LRCN-fc7: LSTM is placed after the second fully connected layer
- Input frame is sized 224x224
- Training is performed with a video of 16 clips. Both training and testing are performed on the UCF-101 dataset.
- Optical flow and RGB inputs are considered

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





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

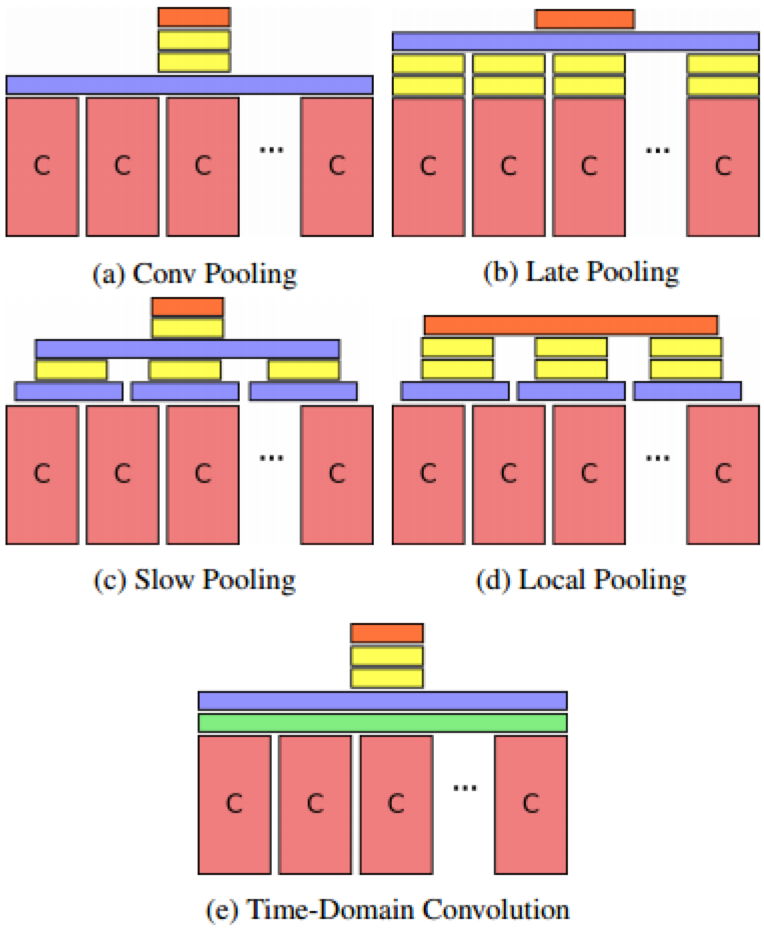
Model	Input Type		Weighted Average	
	RGB	Flow	$^1/_2, ^1/_2$	$^1/_3, ^2/_3$
Single frame	65.40	53.20	–	–
Single frame (ave.)	69.00	72.20	75.71	79.04
LRCN-fc_6	**71.12**	**76.95**	81.97	**82.92**
LRCN-fc_7	70.68	69.36	–	–

- Inputs may be either RGB or optical flow.
- The best performance is about 83%

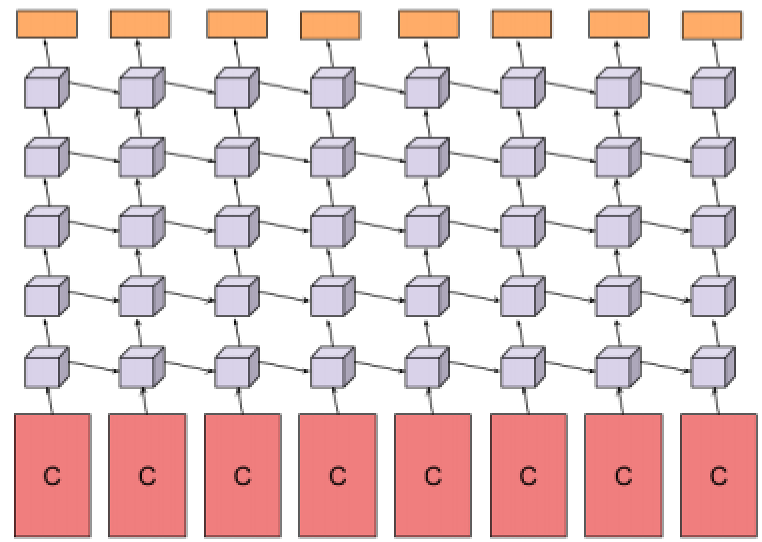
Ng et al., Beyond Short Snippets: Deep Networks for Video Classification

- Only one frame per second
    - Motion information is lost
    - But explicit motion information is available in the form of optical flow
- Two CNN architectures are used to process individual video frames: AlexNet and GoogLeNet.
    - AlexNet: 220x220 image as input, followed by CNNs of size 11, 9 and 5 and two fully connected layers with 4096 ReLUs
    - GoogleNet: 220 × 220 image as input. This image is then passed through multiple Inception modules, each of which applies, in parallel, 1×1, 3×3, 5×5 convolution, and max-pooling operations and concatenates the resulting filters. Finally, the activations are average-pooled and output as a 1000-dimensional vector. This network is 22 layers deep.
- Temporal information is handled via different pooling techniques (convolution, late, slow, local and time-domain convolution) or LSTMs (with 512 memory cells each)
- Training is performed on the Sports 1M-dataset (30-120 frames per video); testing is done on both the Sports-1M (30 frames at 1fps) as well as UCF-101 (30 frames at 6fps) datasets.

Ng et al., Beyond Short Snippets: Deep Networks for Video Classification

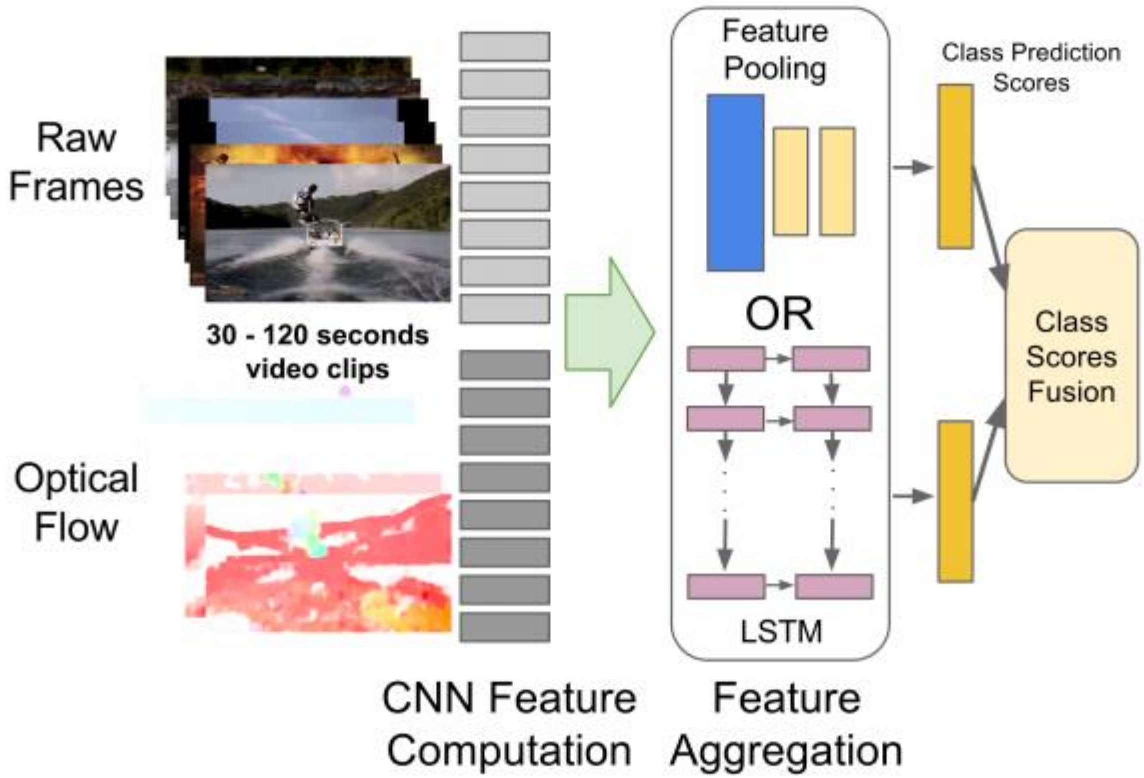


Feature pooling architectures The stacked convolutional layers are denoted by “C”. Blue, green, yellow and orange rectangles represent max-pooling, time-domain convolutional, fully-connected and softmax layers respectively.



CNN followed by 5 LSTM layers

Ng et al., Beyond Short Snippets: Deep Networks for Video Classification



Ng et al., Beyond Short Snippets: Deep Networks for Video Classification

Method	Clip Hit@1	Hit@1	Hit@5
Conv Pooling	**68.7**	**71.1**	**89.3**
Late Pooling	65.1	67.5	87.2
Slow Pooling	67.1	69.7	88.4
Local Pooling	68.1	70.4	88.9
Time-Domain Convolution	64.2	67.2	87.2

Method	Hit@1	Hit@5
AlexNet single frame	63.6	84.7
GoogLeNet single frame	**64.9**	**86.6**
LSTM + AlexNet (fc)	62.7	83.6
LSTM + GoogLeNet (fc)	**67.5**	**87.1**
Conv pooling + AlexNet	70.4	89.0
Conv pooling + GoogLeNet	**71.7**	**90.4**

Method	Hit@1	Hit@5
LSTM on Optical Flow	59.7	81.4
LSTM on Raw Frames	72.1	90.6
LSTM on Raw Frames + LSTM on Optical Flow	73.1	90.5
30 frame Optical Flow	44.5	70.4
Conv Pooling on Raw Frames	71.7	90.4
Conv Pooling on Raw Frames + Conv Pooling on Optical Flow	71.8	90.4

Sports 1M dataset performance:
Pooling, CNN architecture and optical flow comparisons

Method	Frame Rate	3-fold Accuracy (%)
Single Frame Model	N/A	73.3
Conv Pooling (30 frames)	30 fps	80.8
	6 fps	82.0
Conv Pooling (120 frames)	30 fps	82.6
	6 fps	82.6

Method	3-fold Accuracy (%)
Improved Dense Trajectories (IDTF)s [23]	87.9
Slow Fusion CNN [14]	65.4
Single Frame CNN Model (Images) [19]	73.0
Single Frame CNN Model (Optical Flow) [19]	73.9
Two-Stream CNN (Optical Flow + Image Frames, Averaging) [19]	86.9
Two-Stream CNN (Optical Flow + Image Frames, SVM Fusion) [19]	88.0
Our Single Frame Model	73.3
Conv Pooling of Image Frames + Optical Flow (30 Frames)	87.6
Conv Pooling of Image Frames + Optical Flow (120 Frames)	**88.2**
LSTM with 30 Frame Unroll (Optical Flow + Image Frames)	**88.6**

UCF-101 dataset performance:
Optical flow helps improve performance here but not in the sports 1M dataset
Improved performance due to video better centered, less shaky, and better trimmed

Conclusions

- CNNs capture spatial correlation, pooling methods or LSTMs capture temporal correlation. Hence, combine the two for video analytics.
  - We could also have two CNNs, one for stitching in the spatial domain, and another in the temporal domain
- Single frame CNNs can themselves do quite a decent job (around 80% prediction rate with hit@5)
  - But this probably requires the image of object to be centered across the frame and very little noise
- Issue 1: Over-fitting
  - Need to re-train depending on the application?
- Issue 2: Computational time
  - Training time may take a few weeks or even months
  - Reduce no of frames per second/down-sample frames
  - GPUs are essential for training
- Fine tuning higher layers is critical in making the network specific to the application. There is perhaps no need to re-train all the layers.
  - Or we can simply do 'transfer learning' (use logistic regression/random forest-type models on the already learnt features).

Other papers

- **Simoyan et al.,** Two-stream convolutional networks for action recognition in videos, NIPS 2014