Spatiotemporal Pyramid Network for Video Action Recognition

Yunbo Wang

Mingsheng Long

Jianmin Wang

Philip S. Yu

Tsinghua University

China

Paper with the same name to appear in CVPR 2017 https://github.com/thuml/stpyramid

Main idea

Architecture

Experiments

Image Classification to Action Recognition

Cat

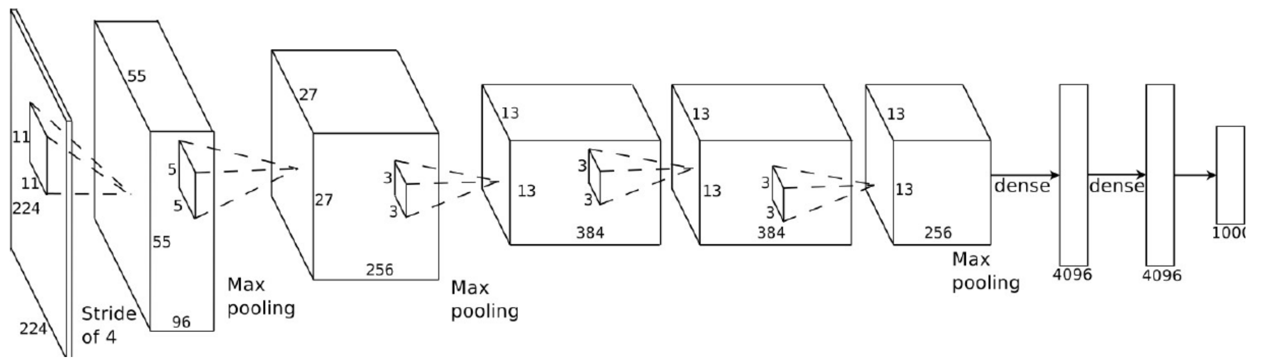
Motion

Basketball

Deep ConvNets

[Krizhevsky et al. 2012]

Input: 227x227x3



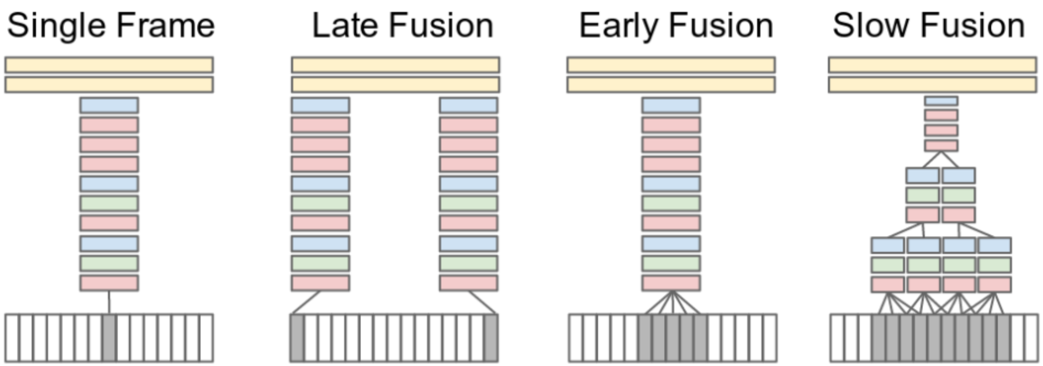
Q: What if the input is now a small chunk of video? E.g. [227x227x3x15]

A: Extend the convolutional filters in time or perform spatiotemporal convolutions!

Spatiotemporal ConvNets – Temporal Fusion

[Karpathy et al. 2014]

Applying 2D CONV on a video volume
(multiple frames as multiple channels)



Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	$\boxed{59.3}$	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**	$\boxed{\mathbf{60.9}}$	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

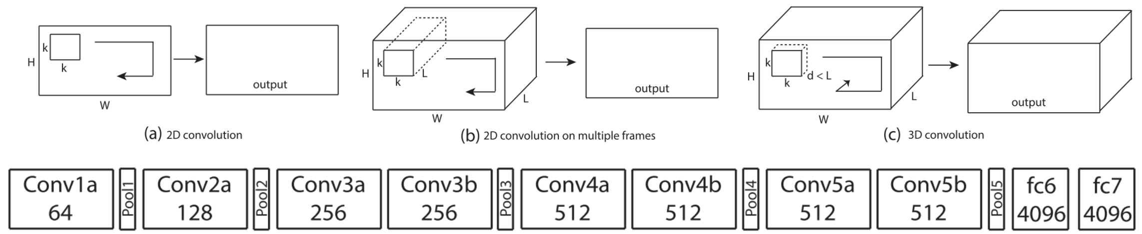
The motion information did not be fully captured…

Spatiotemporal ConvNets – C3D

[Tran et al. 2015]

Applying 3D CONV on a video volume

Accuracy: 85.2%



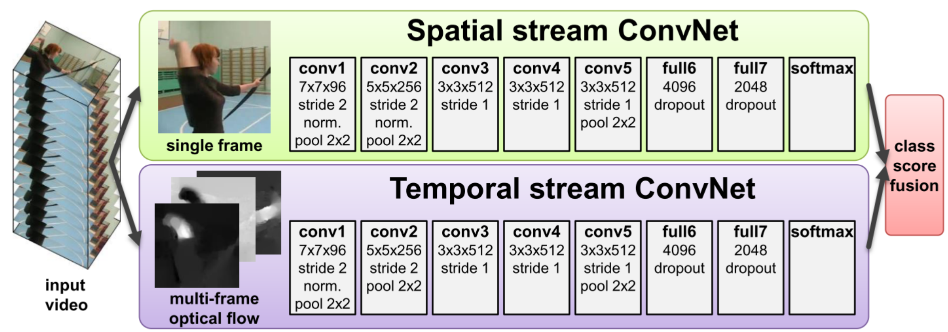
3D VGGNet

Spatiotemporal ConvNets – Optical Flow

[Simonyan and Zisserman. 2014]

Two-stream VGGNet

Accuracy: 88.0% (UCF101)



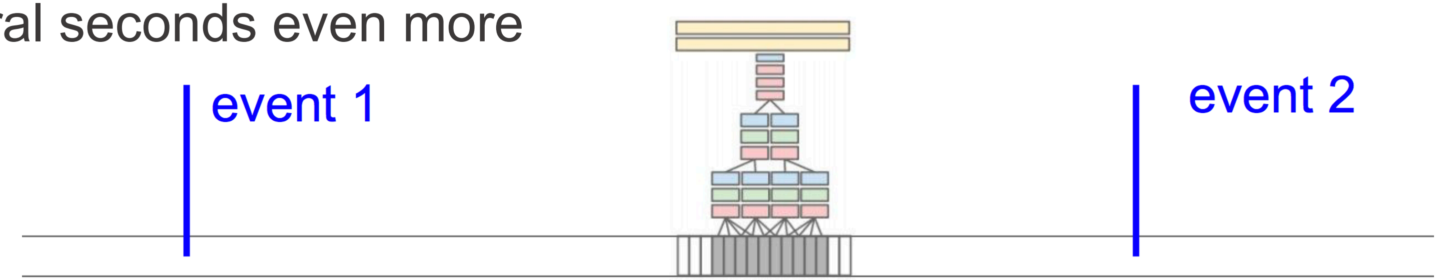
Spatial stream ConvNet	73.0%	40.5%
Temporal stream ConvNet	83.7%	54.6%
Two-stream model (fusion by averaging)	86.9%	58.0%
Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

Two-stream version works much better than either alone.

Motivation 1: Long-Time Dependencies

All above ConvNets used local motion cues to get extra accuracy.
E.g. half a second or less

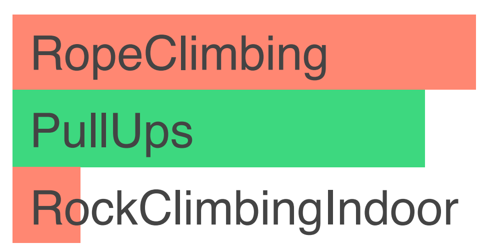
Q: what if the temporal dependencies are much longer?
E.g. several seconds even more



Local motion leads to misclassifications when different actions resemble in short time, though distinguish in the long term.

*E.g. **Pull-ups vs. Rope-climbing***

Classification result produced by
Two-stream ConvNets
[Simonyan and Zisserman, 2014]



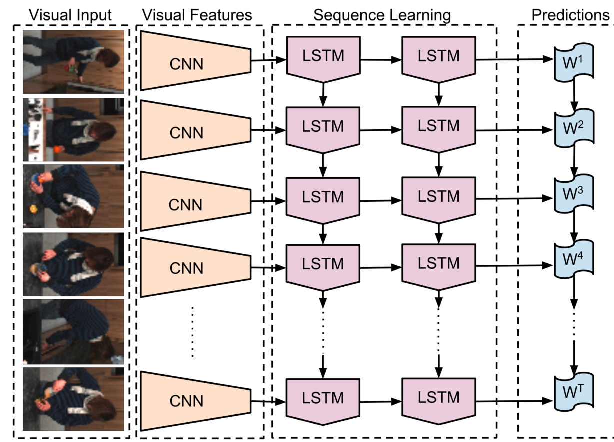
Long-Time Solution – RNNs

[Donahue et al. 2015]

LRCN = ConvNets + LSTM

Long-term temporal extent: RNNs model all video frames in the past.

Accuracy: 82.9%



Learning difficulty in predicting high-dimensional features across states.

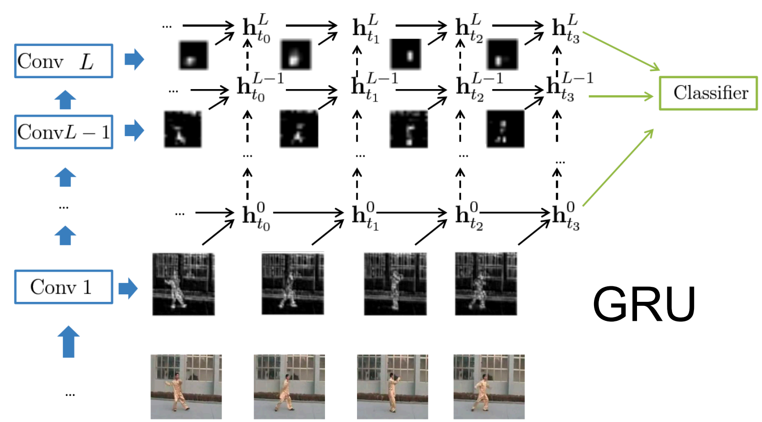
Long-Time Solution – Convolutional RNNs

[Ballas et al. 2016]

ConvNet neurons are recurrent

Only require 2D CONV routines. No need for 3D spatiotemporal CONV.

Accuracy: 80.7%



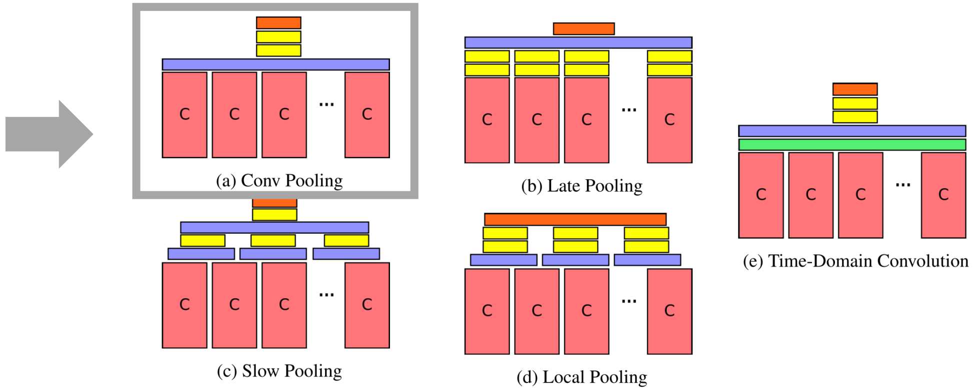
However, convolutional depth is limited by memory usage

Long-Time Solution – Snippets Fusion

Beyond short snippets *[Ng et al. 2015]*

- Explore various pooling methods
- CONV pooling worked best:
  Perform max-pooling over the final CONV layer across frames.

Accuracy: 88.2%



Two-stream fusion *[Feichtenhofer et al. 2016]*

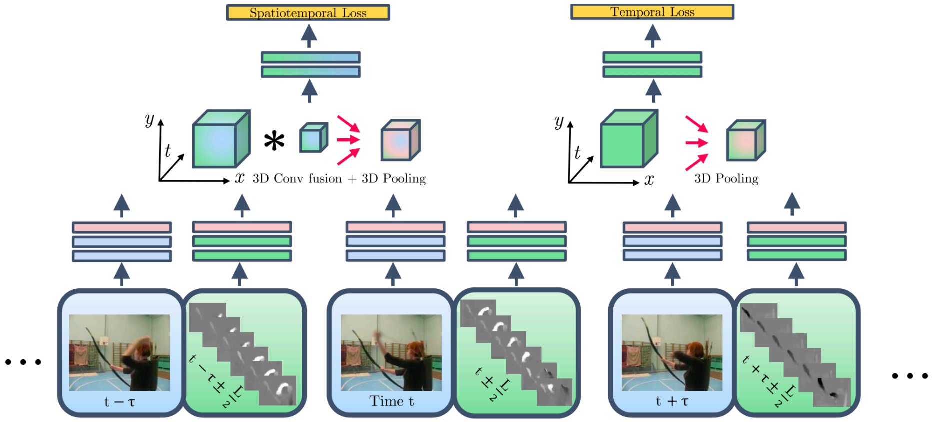
- *Where to fuse networks?*

It is better to fuse them at the last CONV layer

- *How to fuse networks?*

3D CONV fusion and 3D Pooling over spatiotemporal neighborhoods.

Accuracy: 92.5%

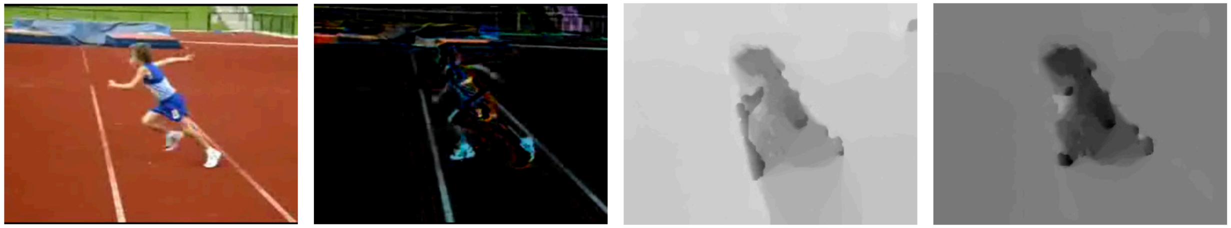


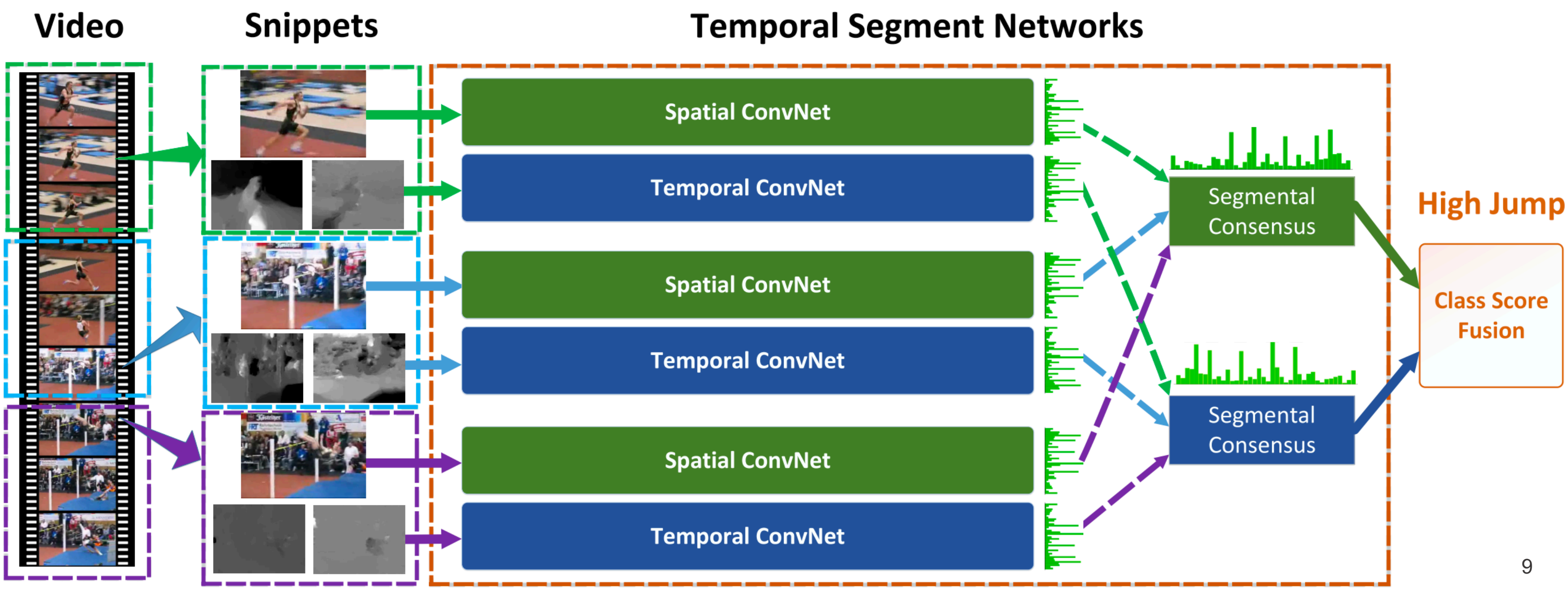
Long-Time Solution – Snippets Fusion

Temporal Segment Networks *[Wang et al. 2016]*

- Segmental consensus: average spatial/temporal features over 3 snippets
- Two new modalities: *RGB difference* and *warped optical flow fields*

Accuracy: 94.0%





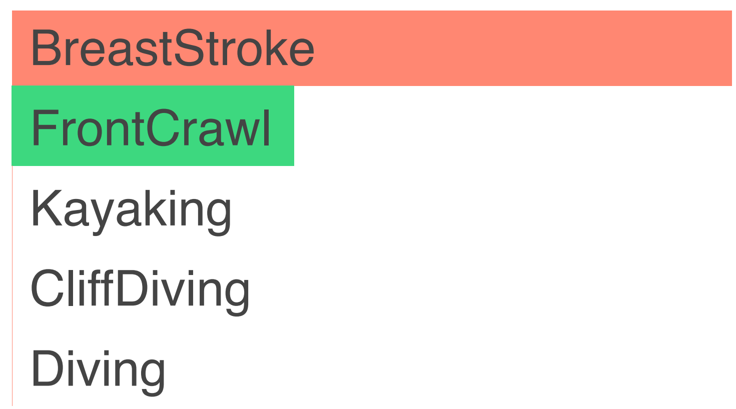
Motivation 2: Visual Interest

Above ImageNet fine-tuned ConvNets are easily fooled by **similar visual scenarios.**

*E.g. **Front Crawl vs. Breast Stroke***

Classification result produced by **Two-stream ConvNets** [Simonyan and Zisserman. 2014]

Ground Truth: FrontCrawl



Visual Interest Solution – Attention

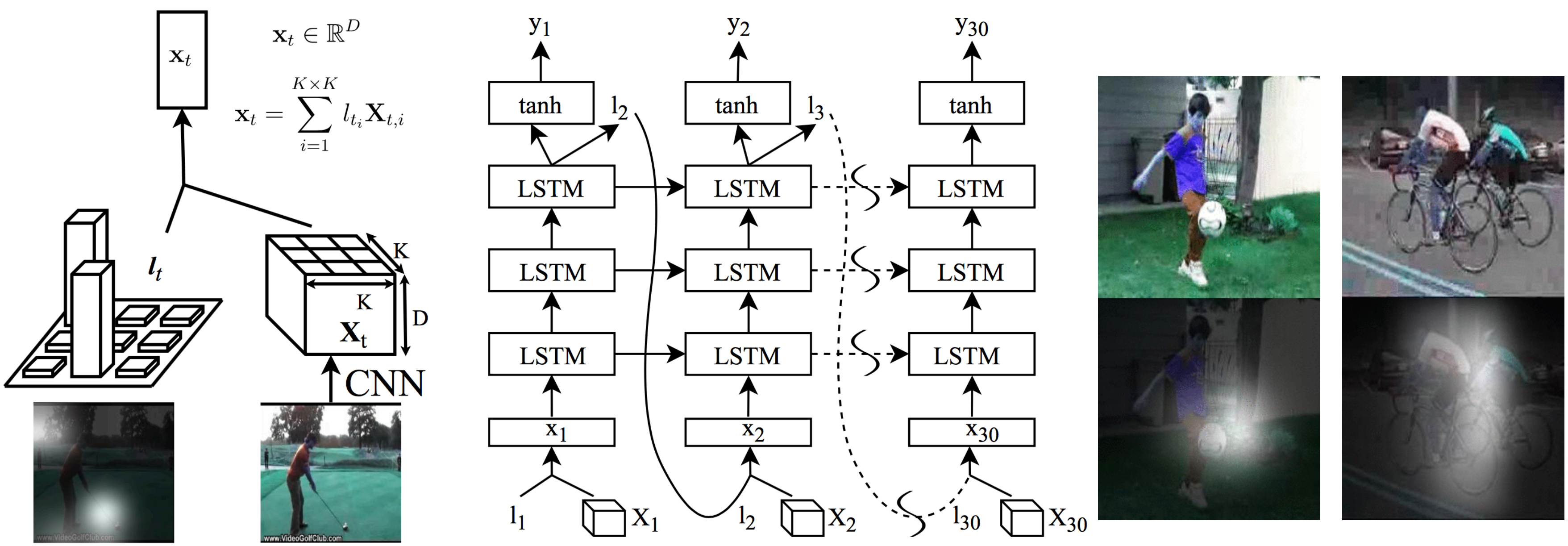
[Sharma et al. 2016]

Attention mechanism:

Pro: Attention mask on the first-layer, giving very intuitive interpretability

Con: The attended features are not discriminative enough for recognition

Accuracy: 85.0%



Main idea

Architecture

Experiments

Spatiotemporal **Pyramid** Networks

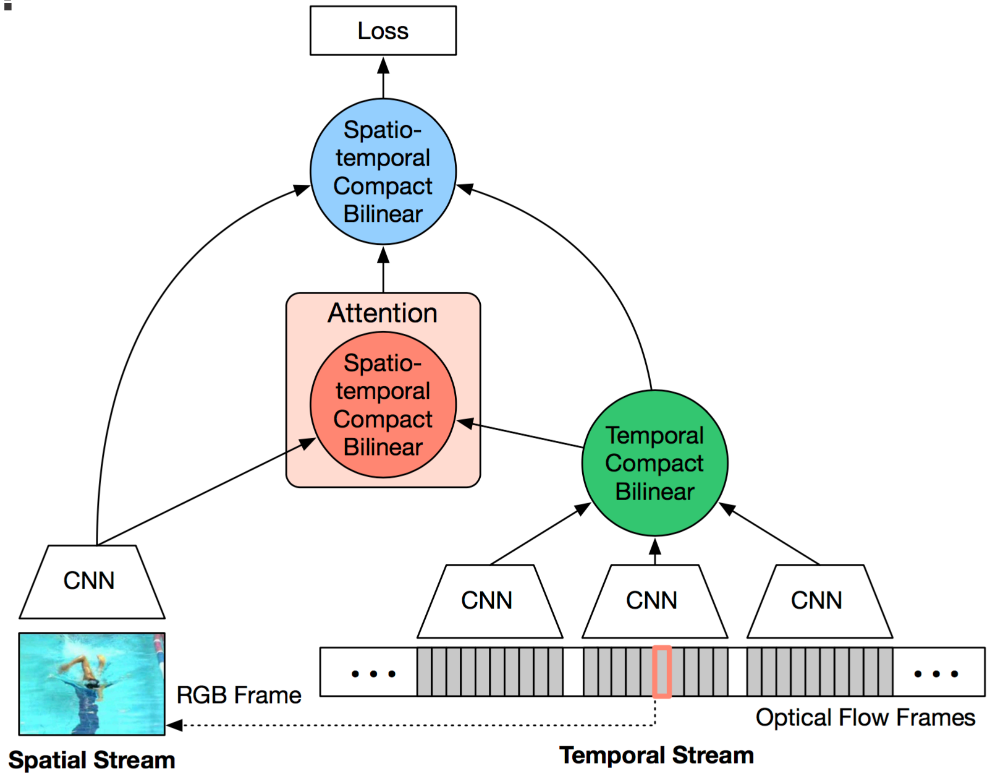
- **What is pyramid?**

  1st fusion level: fuse T ***temporal*** snippets for global motion features

  2nd fusion level: ***attention*** module using global motion as guidance

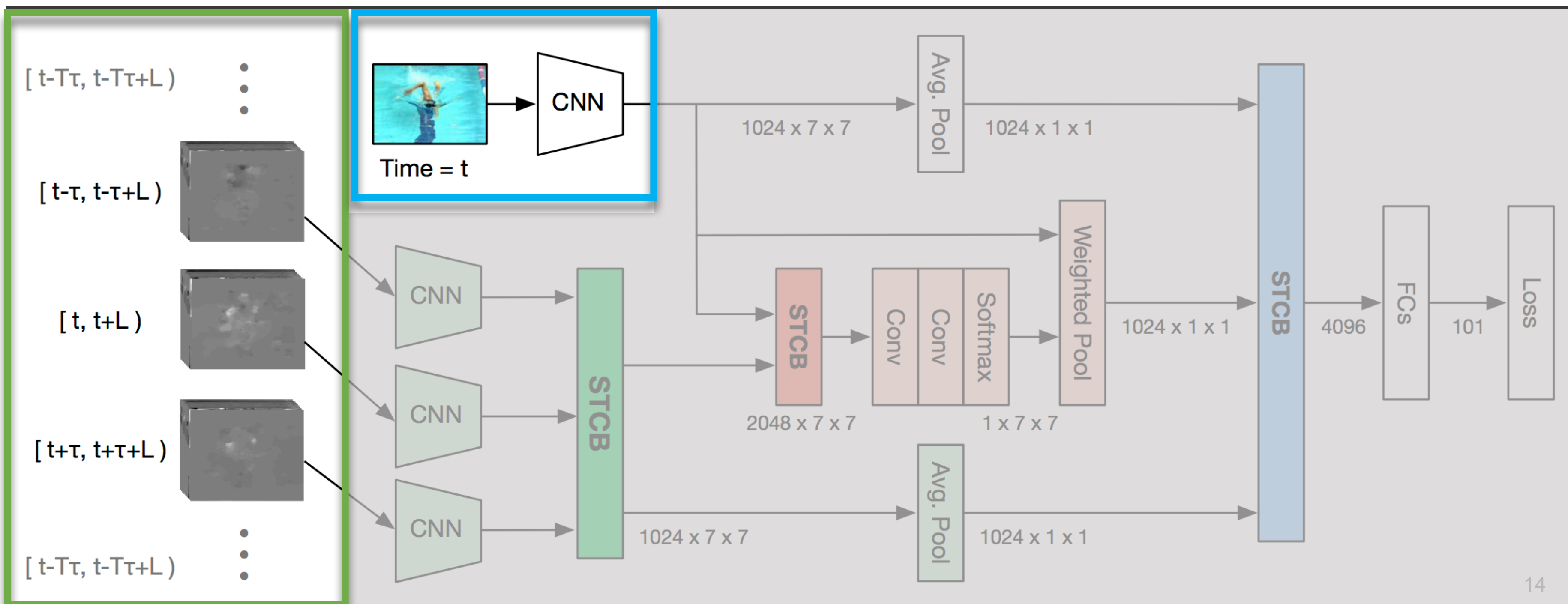
  3rd fusion level: merge ***visual***, ***attention***, ***motion*** features

- **Why pyramid?**



Inputs

- **Spatial:** 1 RGB frame at time t
- **Temporal:** *T* optical flow snippets at an interval of τ around t
- L consecutive frames are covered by each snippet
- L is fixed to 10, τ is randomly selected from 1 to 10, in order to model variable lengths of videos with a fixed number of neurons



Spatiotemporal Compact Bilinear Fusion

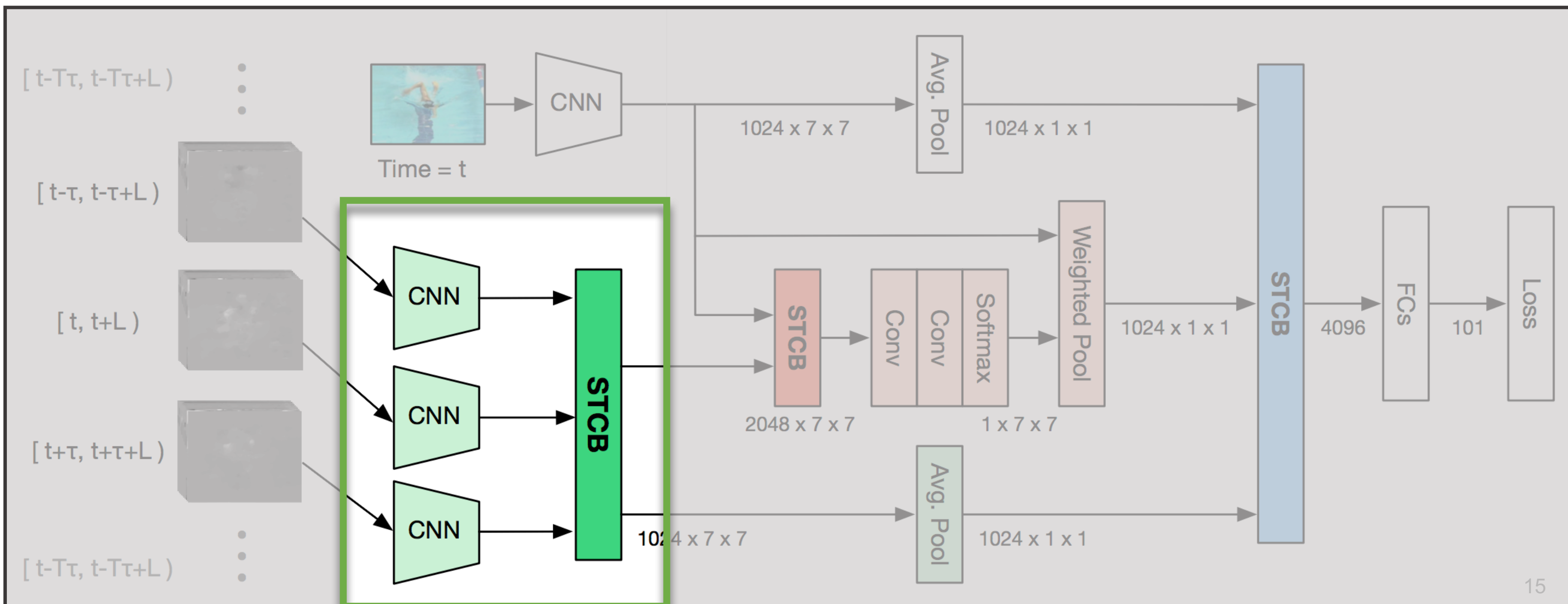
For the long-time dilemma

- Full bilinear features are high dimensional and make subsequent analysis infeasible
- STCB combines single modality (**multi-snippet**) and **multi-modality (spatiotemporal)** features
- STCB preserves the representational ability and efficiently **reduces the output dimension**

Algorithm 1: STCB: Spatiotemporal compact bilinear

Input: Spatial and/or temporal features $\{v_i \in \mathbb{R}^{p_i}\}_{i=1}^m$
Output: Fused features $\Phi(\{v_i\}_{i=1}^m) \in \mathbb{R}^d$

for $i \leftarrow 1$ **to** m **do**
  if h_i, s_i not initialized **then**
    for $j \leftarrow 1$ **to** p_i **do**
      sample $h_i(j)$ from $\{1, \dots, d\}$
      sample $s_i(j)$ from $\{-1, 1\}$
    end
  $v_i' = [0, \dots, 0]$
  for $j \leftarrow 1$ **to** p_i **do**
    $v_i'(h_i(j)) = v_i'(h_i(j)) + s_i(j) \cdot v_i(j)$
  end
  end
  $\Phi(\{v_i\}_{i=1}^m) = \mathrm{FFT}^{-1}(\bigodot_{i=1}^m \mathrm{FFT}(v_i'))$
end

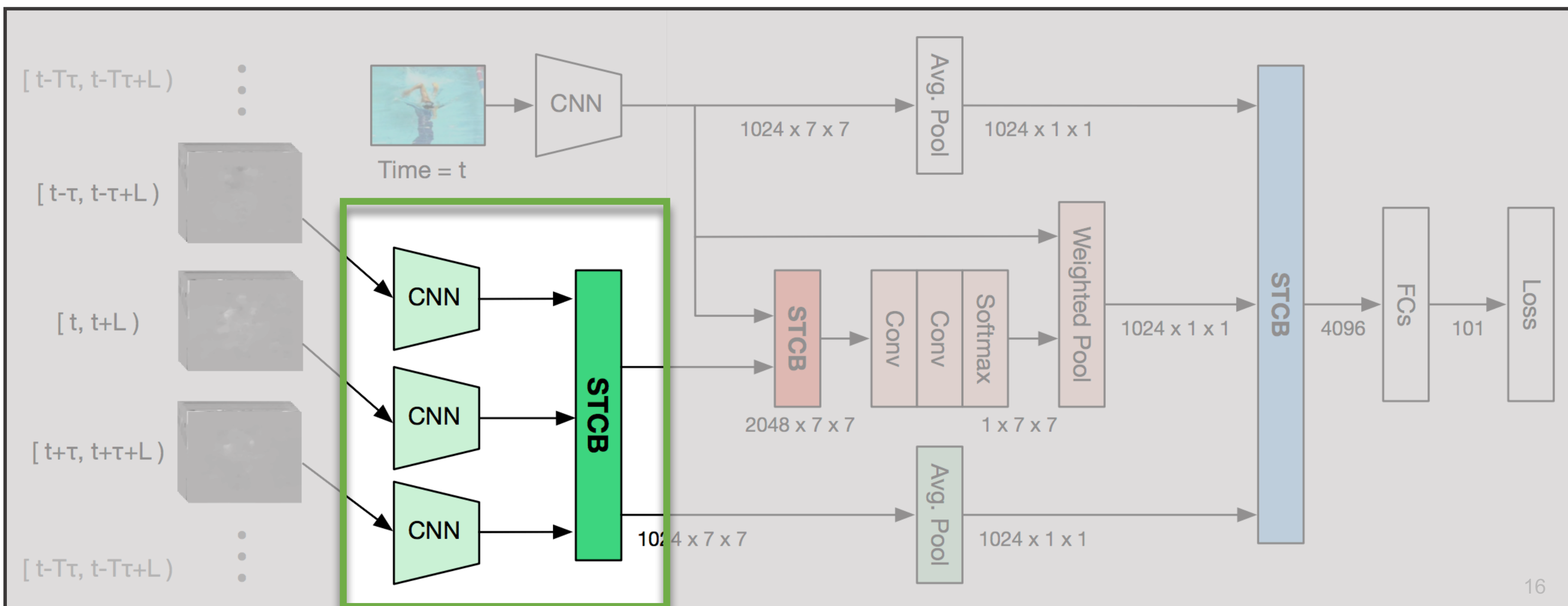


Spatiotemporal Compact Bilinear Fusion

To avoid computing outer-product directly

To project outer-product to lower dimensional space

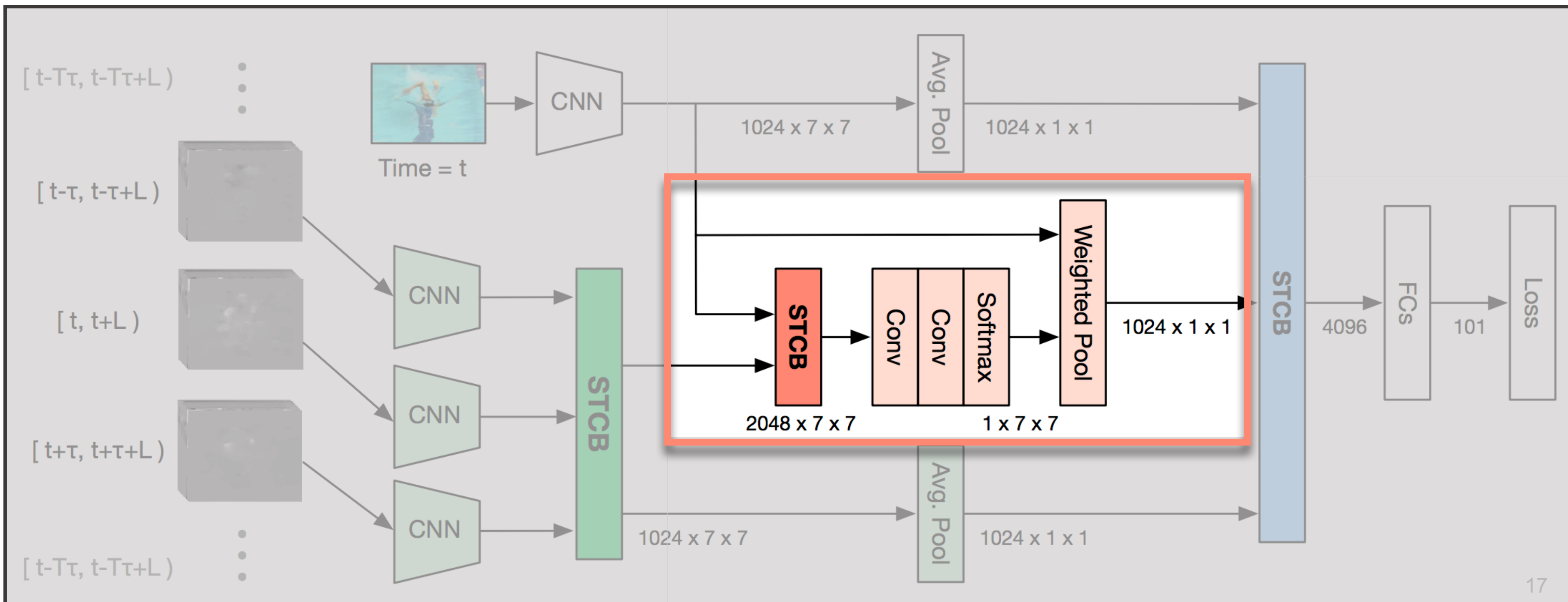
1. Count Sketch: $R^n \rightarrow R^d$
2. Theorem: $\psi\ (x \otimes y) = \psi\ (x) * \psi\ (y)$
3. $\psi\ (x) * \psi\ (y) = FFT^{-1}\ (FFT\ (\psi\ (x)) \odot FFT\ (\psi\ (y)))$



Spatiotemporal Attention

To solve the visual interest problem

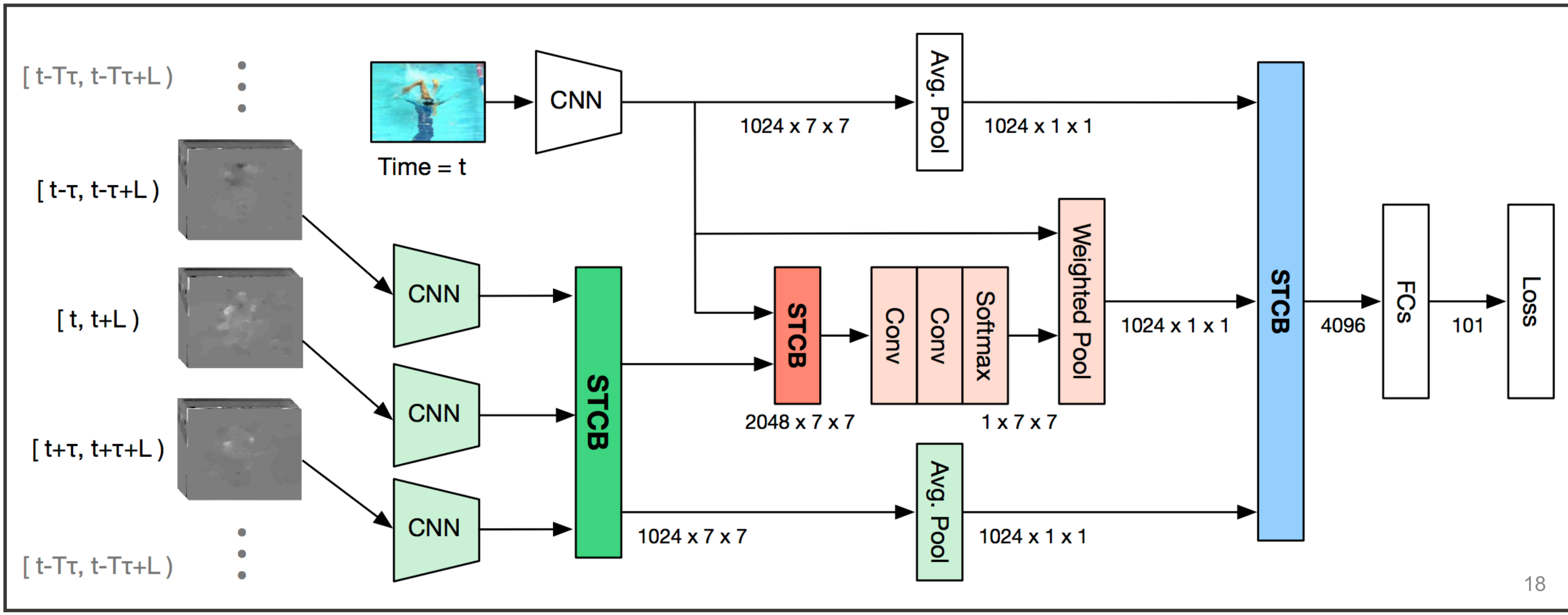
- Plays a role of a more accurate weighted pooling operation
- **Attention guidance:** for each grid location on the image feature maps, we use STCB to merge the spatial and temporal feature vectors
- **Generate attention weights:** CONV*2 → Softmax along each location → Weighted pooling on the spatial feature maps



Final Architecture – Pyramid

A framework **extendible** for almost all deep ConvNets
E.g. VGGNets, BN-Inception, ResNets, etc.

- 1st fusion level: fuse K ***temporal*** snippets for global motion features
- 2nd fusion level: ***attention*** module using global motion as guidance
- 3rd fusion level: merge ***visual***, ***attention***, ***motion*** features



Main idea

Architecture

Experiments

Technical Details

- **BN-Inception turns out to be the top-performing base architecture.** Due to the limited amount of training samples on UCF101, complex network structures are prone to over-fitting.

Model	Spatial	Temporal	Two-Stream [22]
VGG-16	80.5%	85.4%	88.9%
ResNet-50	83.7%	84.9%	90.3%
ResNet-152	84.3%	82.1%	89.8%
BN-Inception	84.5%	87.0%	91.7%

- **Training protocols consistent with *[Wang et al. ECCV 2016]***
- **Cross modality pre-training:** Use ImageNet pre-trained models to initialize the temporal ConvNet
  - ➢ Average weights across the RGB channels in the first CONV layer
  - ➢ Replicate them by the optical flow channel number (e.g. 20)
- **Partial batch normalization:** Freeze the mean and variance of all CONV layers except the first one (as the distribution of optical flow is different from the RGB, its mean and variance need to be re-estimated)
- **Data augmentation:** horizontal flipping, corner cropping, scale-jittering.

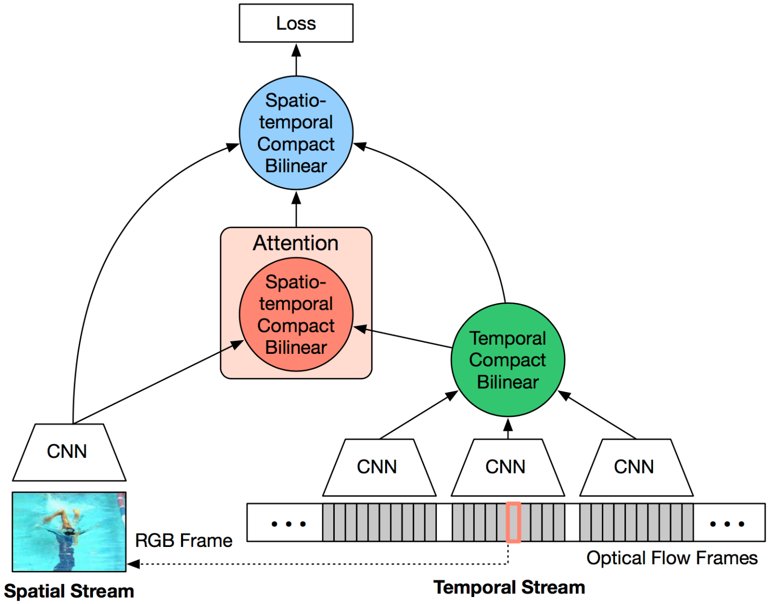
Ablation Study

- Multi-snippets temporal fusion (optical flow only)

Fusion method	1-path	3-path	5-path
Concatenation	87.0%	88.4%	88.5%
Element-wise sum	-	87.9%	87.7%
Compact bilinear	**-**	**89.3%**	**89.2%**

- Attention (spatial features only)

Fusion method	Acc.
Spatial ConvNet (AvgPool)	84.5%
Att. (1-snippet as guidance)	84.3%
Att. (3-snippets concat)	83.9%
Att. (3-snippets STCB)	**86.6%**

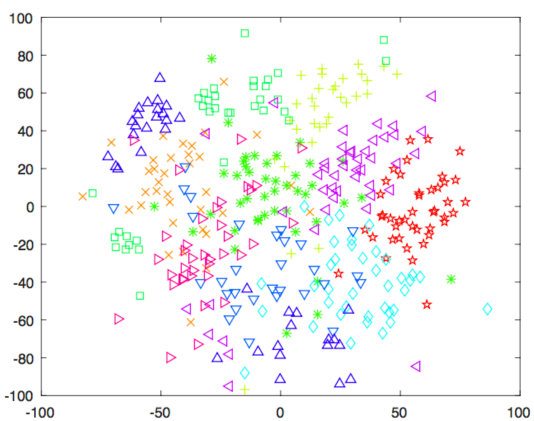


Ablation Study

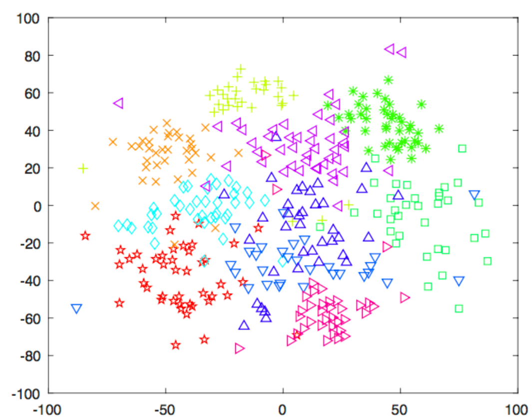
- Now we stack these fusion methods one by one

Model	A	B	C	D
Two-stream STCB	0	1	1	1
Multi-snippets fusion	0	0	1	1
Attention	0	0	0	1
Accuracy	91.7%	93.2%	93.6%	**94.6%**

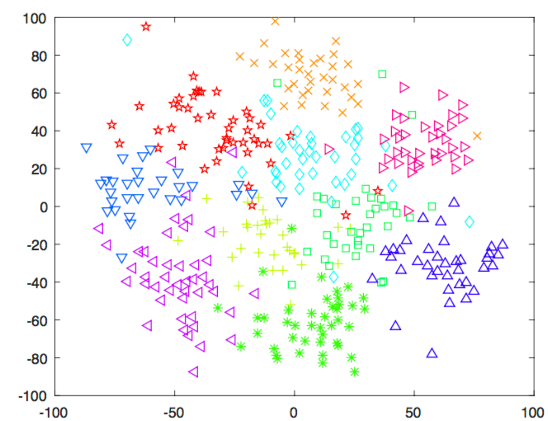
- t-SNE of 10 classes randomly selected from UCF101



Model B



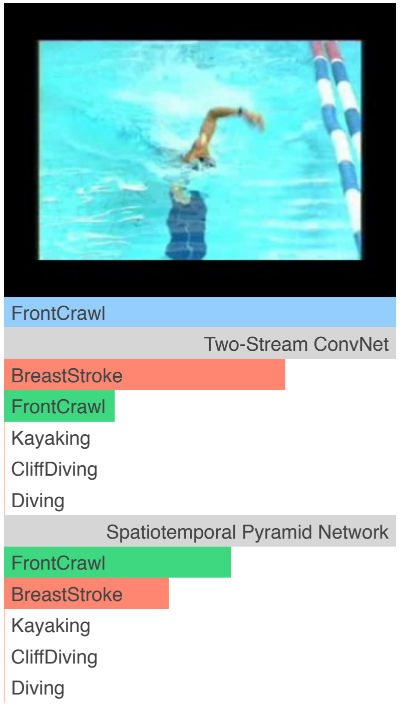
Model C

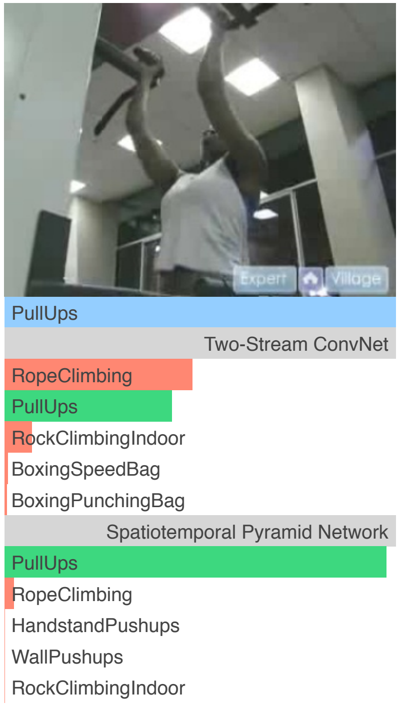


Pyramid (Model D)

Final Results

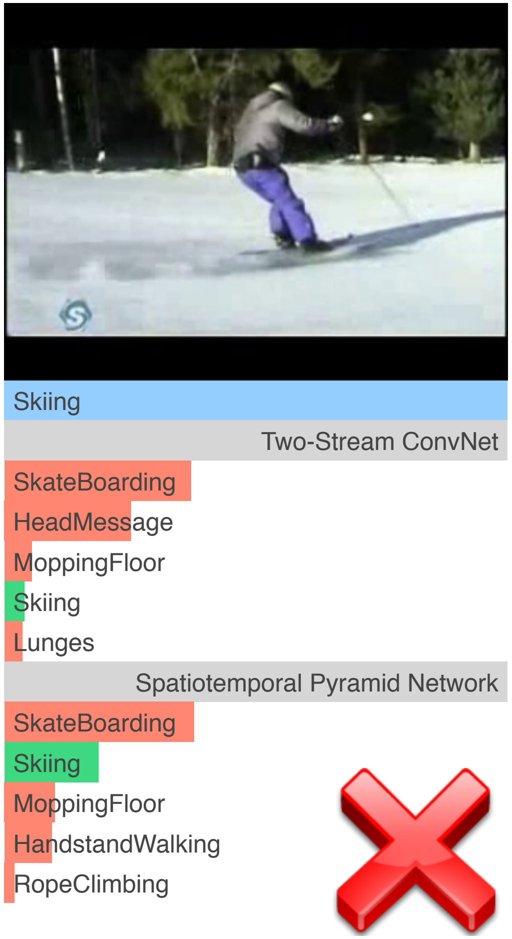
Method	UCF101	HMDB51
Slow Fusion CNN [12]	65.4%	-
LRCN [5]	82.9%	-
C3D [28]	85.2%	-
Two-Stream (AlexNet) [22]	88.0%	59.4%
Two-Stream + LSTM [37]	88.6%	-
Two-Stream + Pooling [37]	88.2%	-
Transformation [33]	92.4%	62.0%
Two-Stream (VGG-16) [6]	90.6%	58.2%
Two-Stream + Fusion [6]	92.5%	65.4%
TSN (BN-Inception) [32]	94.0%	68.5%
Ours (VGG-16)	93.2%	66.1%
Ours (ResNet-50)	93.8%	66.5%
Ours (BN-Inception)	**94.6%**	**68.9%**



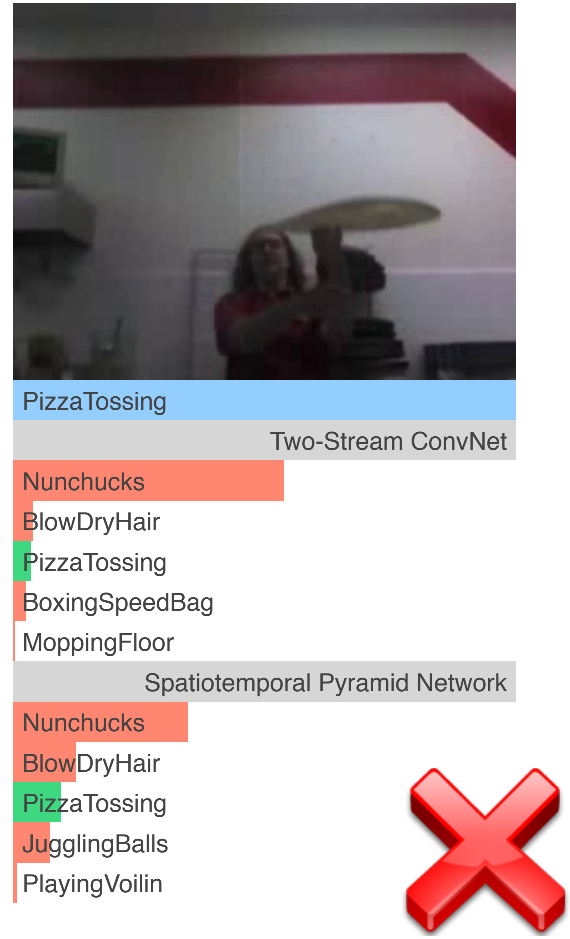


- Spatially ambiguous classes can be separated by the **attention** mechanism. *E.g.* ***Front Crawl vs. Breast Stroke***
- Multi-snippets **temporal fusion** produces more global features and can easily differentiate actions that look similar in short-term. *E.g.* ***Pull-ups vs. Rope-climbing***

Future Work



Similar action
different backgrounds



Similar action
different objects in hands

Thank you!

https://github.com/thuml/stpyramid