Action Recognition

ICIP2019 Tutorial

Outline

- Problem space
- Datasets
  - RGB
  - RGB-D
- Skeleton-based approaches
- Video based approaches

Problem space

- Gesture, action, activity
- Classification, detection, online recognition
- RGB, depth, skeleton

Gesture, Action, Activity

- Hand gesture
  - Short, single person, focused on hands
    - American Sign Language
- Action
  - Short, single person, involving the body
    - Throw, catch, clap
- Activity
  - Longer, one or multiple people
    - Reading a book, making a phone call, eating
    - Talking to each other, hugging, playing basketball

Classification, Detection, Online Recognition

- Classification
  - Given a pre-segmented clip, predict its action class label

Classification, Detection, Online Recognition

- Detection
  - Multiple actions may occur simultaneously in different locations and/or at different times

Where
When
What

Classification, Detection, Online Recognition

- Online recognition
  - No future frames available
  - Recognizing when an action starts/ends
- Action prediction
  - prediction with partial observation

Outline

- Problem space
- Datasets
  - RGB
  - RGB-D
- Skeleton-based approaches
- Video based approaches

Datasets - RGB

Dataset	Classes	Examples	Duration	State-of-art(Acc)
UCF101	101	13320	2~16 s	98%
HMDB51	51	6849	1~10s	82.1%
Kinetics	400/600	500K	~10s	~79%
sports1M	487	1133158	>5min	~73.3%
charades	157	~8k train;~1.8k validation ; ~2ktest		~39.5%
Moments in Time	339	~1million	~3s	
YouTube-8M	4800	8million	120-500s	

Datasets - RGBD

Dataset	year	Acquisition device	Seg/Con	Modality	#Class	#Subjects	#Samples	#Views	Metric
CMU Mocap	2001	Mocap	Seg	RGB,S	45	144	2235	1	Accuracy
HDM05	2007	Mocap	Seg	RGB,S	130	5	2337	1	Accuracy
MSR-Action3D	2010	Kinect v1	Seg	S,D	20	10	567	1	Accuracy
MSRC-12	2012	Kinect v1	Seg	S	12	30	594	1	Accuracy
MSR DailyActivity3D	2012	Kinect v1	Seg	RGB,D,S	16	10	320	1	Accuracy
UTKinect	2012	Kinect v1	Seg	RGB,D,S	10	10	200	1	Accuracy
G3D	2012	Kinect v1	Seg	RGB,D,S	5	5	200	1	Accuracy
SBU Kinect Interaction	2012	Kinect v1	Seg	RGB,D,S	7	8	300	1	Accuracy
Berkeley MHAD	2013	Mocap Kinect v1	Seg	RGB,D,S,Au,Ac	12	12	660	4	Accuracy
Multiview Action3D	2014	Kinect v1	Seg	RGB,D,S	10	10	1475	3	Accuracy
ChaLearn LAP IsoGD	2016	Kinect v1	Seg	RGB,D	249	21	47,933	1	Accuracy
NTU RGB+D	2016	Kinect v2	Seg	RGB,D,S,IR	60	40	56,880	80	Accuracy
ChaLearn2014	2014	Kinect v1	Con	RGB,D,S,Au	20	27	13,858	1	Accuracy JI etc.
ChaLearn LAP ConGD	2016	Kinect v1	Con	RGB,D	249	21	22,535	1	JI
PKU-MMD	2017	Kinect v2	Con	RGB,D,S,IR	51	66	1076	3	JI etc.

Outline

- Problem space
- Datasets
  - RGB
  - RGB-D
- Skeleton-based approaches
- Video based approaches
  - CNN features

Action Recognition

- Feature representation
- Classifier
- Spatial-temporal modeling

Feature Representation

- Hand-crafted Feature: HOG, HOF, dense Trajectory
- Skeleton
  - Skeleton Joints: ST-NBNN, ST-GCN, …
  - Skeleton Heatmaps
- Two Stream: RGB + Optical flow
- 3D (spatial-temporal space) convolution

ST-NBNN

- Motivation
  - Non-parametric model like NBNN has not been well explored in this field
    - NBNN has been successful applied in image recognition
  - Recognition of a certain action only related to movement of a subset of joints (spatial)and to a few certain frames (temporal)

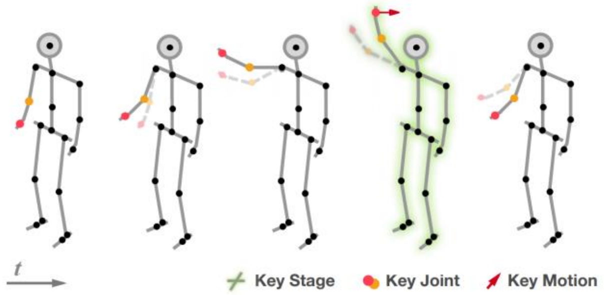
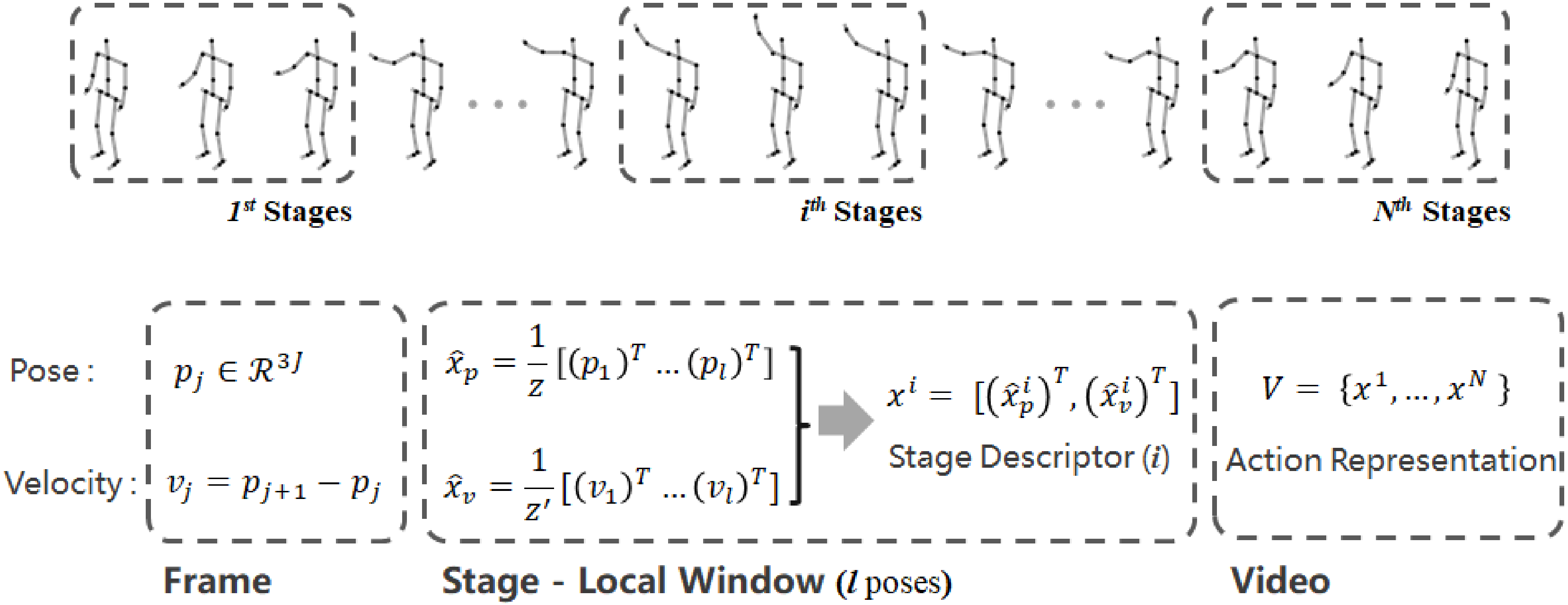


Figure 1. An Illustration of Key Stage, Joints, and Motion for the action of waving right hand action.

Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition,Junwu Weng Chaoqun Weng Junsong Yuan, CVPR2017

ST-NBNN

- Representation



Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition,Junwu Weng Chaoqun Weng Junsong Yuan, CVPR2017

ST-NBNN

- Method

NBNN:

$$\hat{c} = arg\min_{c} \sum_{i=1}^{N} \left\| x^i - NN_c(x^i) \right\|^2 = arg\min_{c} sum(X_c)$$

sum() : Summation of elements in X_c

NBNN+SVM:

1) Too many parameters
2) Easy to over-fitting

$$\hat{c} = arg\min_{c} w^T x_c$$

w^T Weights learnt by linear SVM

x_c Vectorized X_c

ST-NBNN:

$$\hat{c} = arg\min_{c} (u_c^s)^T X_c u_c^t = arg\min_{c} f_c(X_c)$$

u_c^s Spatial Weights

u_c^t Temporal Weights

Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition,Junwu Weng Chaoqun Weng Junsong Yuan, CVPR2017

ST-NBNN

- Experiments

Method	MSR	UTK	UCB
NBNN-N	91.7	95.5	88.0
NBNN+SVM	92.4	94.0	**100.0**
Best Method	**94.8**[6][33]	**98.2**[32]	**100.0**[6]
Ours	**94.8**	**98.0**	**100.0**

Table.1 Results on MSR-Action3D, UT-Kinect, Berkeley MHAD

Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition,Junwu Weng Chaoqun Weng Junsong Yuan, CVPR2017

Summary for ST-NBNN

- Feature Representation
  - Joint position & Velocity
- Classifier
  - NBNN
- Spatial-temporal modeling
  - Spatial / temporal weights

Deformable Pose Traversal Convolution

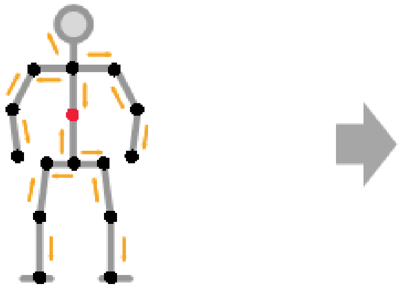
- Motivation
  - More discriminative feature representation
    - Pose information exchange
- Temporal modeling

Deformable Pose Traversal Convolution

- Pose Traversal to transfer graph into vector

Undirected acyclic graph

Vector



$\boldsymbol{x} \in \mathbb{R}^{J}$ One-channel version

$\boldsymbol{X} \in \mathbb{R}^{J \times C}$ C-channel version

- Most of the joints are visited more than once
- the spatial neighborhood relationship among joints is preserved
- Each sequence is represented as $V = \{\boldsymbol{x}^t\}_{t=1}^{T}$

Deformable Pose Traversal Convolution for 3D Action and Gesture Recognition, Junwu Weng, Mengyuan Liu, Xudong Jiang, Junsong Yuan, ECCV2018

Deformable Pose Traversal Convolution

- Regular sampling

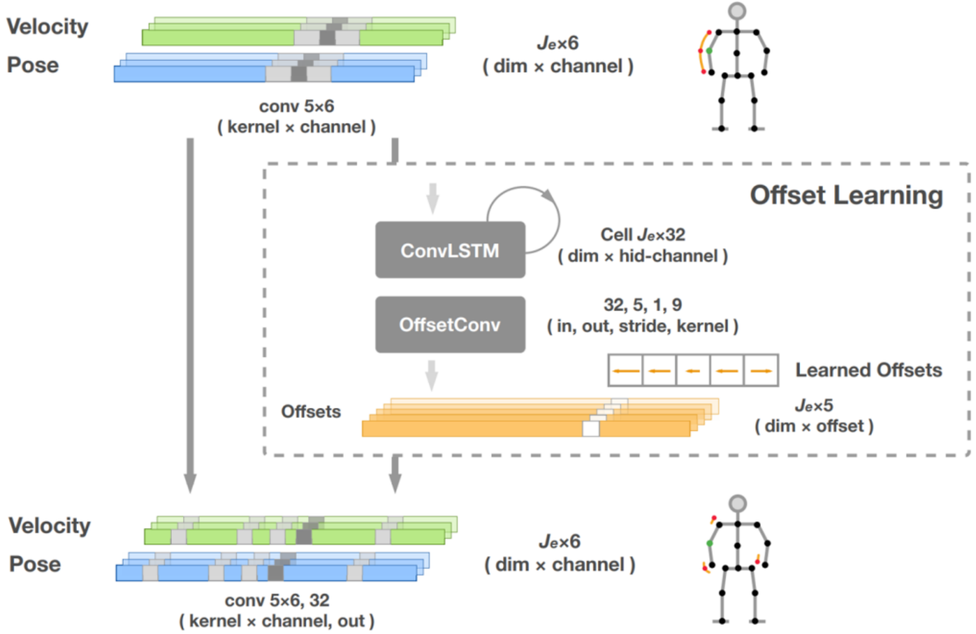
$$\boldsymbol{y}(i_0) = \sum_{i_n \in \mathbf{G}} \boldsymbol{w}(i_n) \cdot \boldsymbol{x}(i_0 + i_n) \qquad \mathbf{G} = \{-M, ..., -1, 0, 1, ..., M\}$$

- Deformable sampling

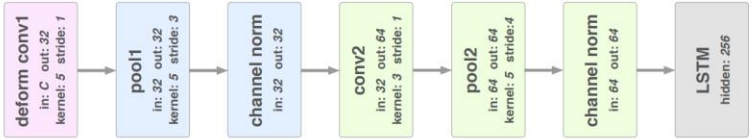
$$\boldsymbol{y}(i_0) = \sum_{(i_n, \delta_n) \in \tilde{\mathbf{G}}} \boldsymbol{w}(i_n) \cdot \boldsymbol{x}(i_0 + i_n + \delta_n) \qquad \tilde{\mathbf{G}} = \{(i_n, \delta_n)\}_{n=1}^{N}$$

Deformable Pose Traversal Convolution

- Method



(a) Structure of Pose Traversal Convolution Network



(b) Structure of Deformable Pose Traversal Convolution Network

Deformable Pose Traversal Convolution for 3D Action and Gesture Recognition, Junwu Weng, Mengyuan Liu, Xudong Jiang, Junsong Yuan, ECCV2018

Deformable Pose Traversal Convolution

- Experiment

Method	DHG-F	DHG-C	DHG14	DHG28	MHAD	NTU.CS	NTU.CV
Pose Chain	76.2	90.4	80.4	75.7	96.4	75.2	83.4
Pose Traversal	77.1	91.8	81.1	76.6	98.6	76.1	84.3
D-Pose Traversal	81.9	95.2	85.8	80.2	100.0	76.8	84.9
Best Method	73.6	88.3	83.1	80.0	100.0	83.2	89.3

Deformable Pose Traversal Convolution for 3D Action and Gesture Recognition, Junwu Weng, Mengyuan Liu, Xudong Jiang, Junsong Yuan, ECCV2018

Summary

- Feature Representation
    - Joint position & Velocity + deformable pose traversal convolution
- Classifier
    - LSTM
- Spatial-temporal modeling
    - Spatial: deformable pose traversal convolution
    - Temporal: LSTM

ST-GCN

- Motivation
  - Encode the spatial and temporal structure of joints

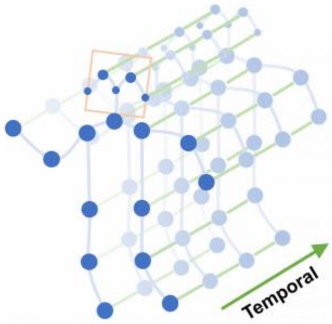
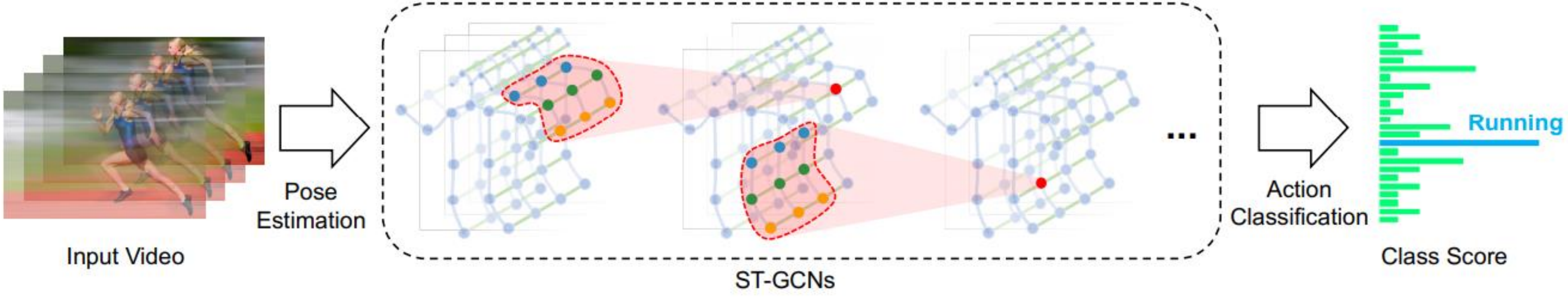


Figure 1: The spatial temporal graph of a skeleton sequence used in this work where the proposed ST-GCN operate on. Blue dots denote the body joints. The intra-body edges between body joints are defined based on the natural connections in human bodies. The inter-frame edges connect the same joints between consecutive frames. Joint coordinates are used as inputs to the ST-GCN.



Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, Sijie Yan and Yuanjun Xiong and Dahua Lin, AAAI 2018

ST-GCN

- Spatial Graph Convolutional Neural Network

$$\mathbf{f}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{f}_{in}\mathbf{W},$$

$$\Lambda^{ii} = \sum_j (A^{ij} + I^{ij}).$$

Network architecture and training. Since the ST-GCN share weights on different nodes, it is important to keep the scale of input data consistent on different joints. In our experiments, we first feed input skeletons to a batch normalization layer to normalize data. The ST-GCN model is composed of 9 layers of spatial temporal graph convolution operators (ST-GCN units). The first three layers have 64 channels for output. The follow three layers have 128 channels for output. And the last three layers have 256 channels for output. These layers have 9 temporal kernel size. The Resnet mechanism is applied on each ST-GCN unit. And we randomly dropout the features at 0.5 probability after each ST-GCN unit to avoid overfitting. The strides of the 4-th and the 7-th temporal convolution layers are set to 2 as pooling layer. After that, a global pooling was performed on the resulting tensor to get a 256 dimension feature vector for each sequence. Finally, we feed them to a SoftMax classifier. The

Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, Sijie Yan and Yuanjun Xiong and Dahua Lin, AAAI 2018

ST-GCN

- Experiments

	Top-1	Top-5
Baseline TCN	20.3%	40.0%
Local Convolution	22.0%	43.2%
Uni-labeling	19.3%	37.4%
Distance partitioning*	23.9%	44.9%
Distance Partitioning	29.1%	51.3%
Spatial Configuration	29.9%	52.2%
ST-GCN + Imp.	**30.7%**	**52.8%**

Table 1: Ablation study on the Kinetics dataset. The "ST-GCN+Imp." is used in comparison with other state-of-the-art methods. For meaning of each setting please refer to Sec.4.2.

	Top-1	Top-5
RGB(Kay et al. 2017)	57.0%	77.3%
Optical Flow (Kay et al. 2017)	49.5%	71.9%
Feature Enc. (Fernando et al. 2015)	14.9%	25.8%
Deep LSTM (Shahroudy et al. 2016)	16.4%	35.3%
Temporal Conv. (Kim and Reiter 2017)	20.3%	40.0%
ST-GCN	**30.7%**	**52.8%**

Table 2: Action recognition performance for skeleton based models on the Kinetics dataset. On top of the table we list the performance of frame based methods.

	X-Sub	X-View
Lie Group (Veeriah, Zhuang, and Qi 2015)	50.1%	52.8%
H-RNN (Du, Wang, and Wang 2015)	59.1%	64.0%
Deep LSTM (Shahroudy et al. 2016)	60.7%	67.3%
PA-LSTM (Shahroudy et al. 2016)	62.9%	70.3%
ST-LSTM+TS (Liu et al. 2016)	69.2%	77.7%
Temporal Conv (Kim and Reiter 2017).	74.3%	83.1%
C-CNN + MTLN (Ke et al. 2017)	79.6%	84.8%
ST-GCN	**81.5%**	**88.3%**

Table 3: Skeleton based action recognition performance on NTU-RGB+D datasets. We report the accuracies on both the cross-subject (X-Sub) and cross-view (X-View) benchmarks.

Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, Sijie Yan and Yuanjun Xiong and Dahua Lin, AAAI 2018

ST-GCN

- Extensions
  - 2s-AGCN
    - Predefined Graph structure
    - Graph structure fixed for all layers and shared for all the classes
  - AGC-LSTM
    - capture discriminative features in spatial configuration and temporal dynamics, but also explore the co-occurrence relationship between spatial and temporal domains

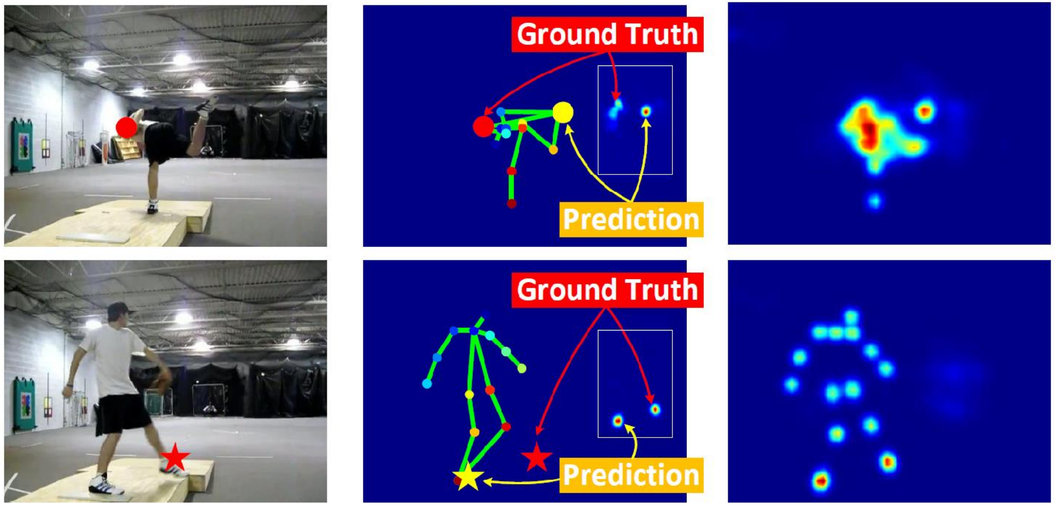
Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, Sijie Yan and Yuanjun Xiong and Dahua Lin, AAAI 2018
Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition, Lei Shi, Yifan Zhang, Jian Cheng, Hanqing Lu, CVPR2019
An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition, Chenyang Si, Wentao Chen, Wei Wang,Liang Wang, Tieniu Tan, CVPR2019

Summary for ST-GCN

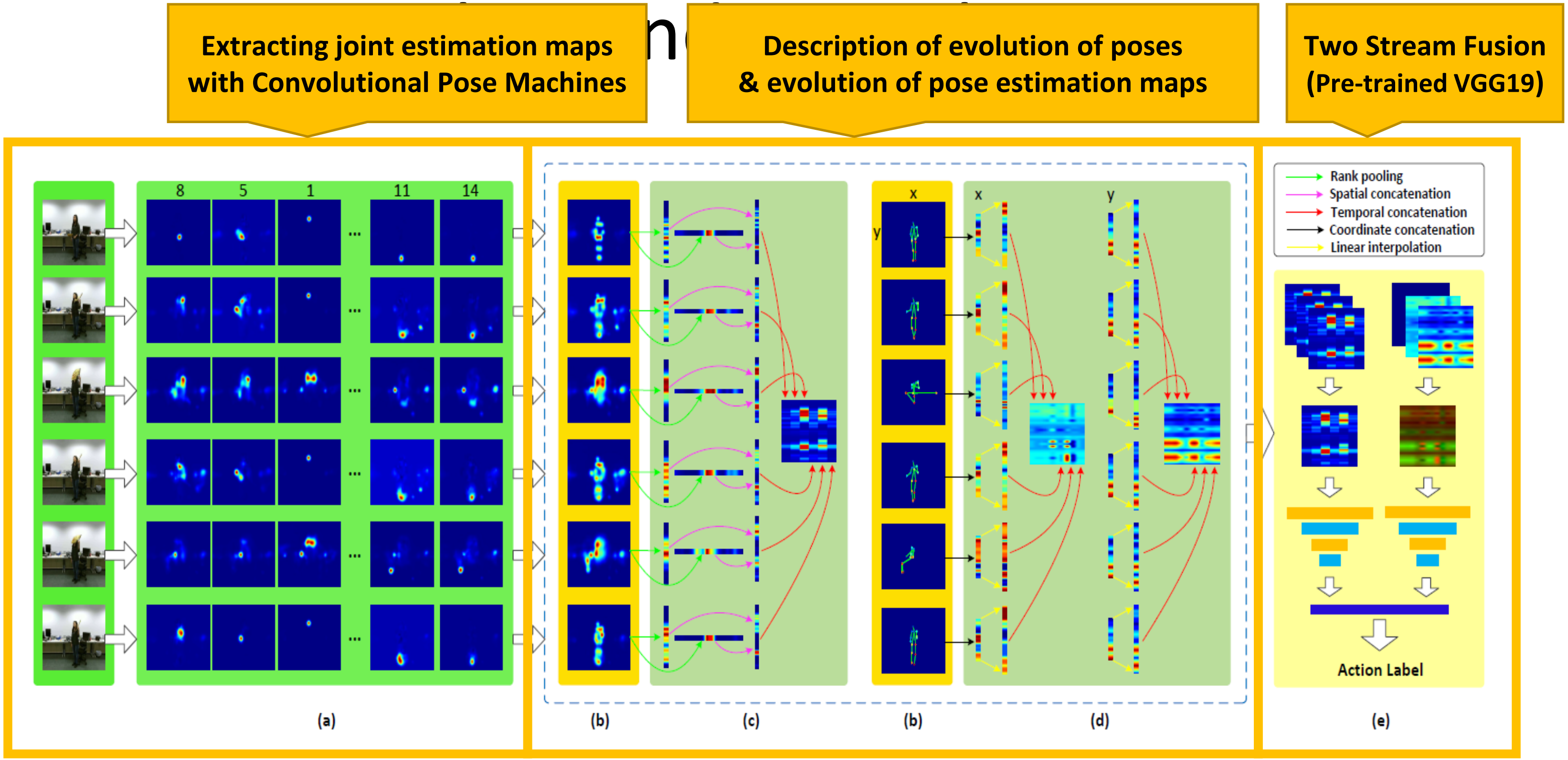
- Feature Representation
  - 2D/3D Joint position
- Classifier
  - GCN
- Spatial-temporal modeling
  - Spatial-temporal Adjacency matrix

Pose Estimation Maps

- Motivation
  - Estimate **2d poses** from RGB frames are usually **noisy** due to partial occlusions and self-similarities.
  - **Pose estimation map** provides global body shape, which can be used to **correct noisy** pose joints.



Recognizing Human Actions as the Evolution of Pose Estimation Maps, Mengyuan Liu, Junsong Yuan, CVPR2018



1. We design compact signatures for evolution of poses and evolution of pose estimation maps
2. We test the performance of action recognition using sole estimated 2d poses
3. We fuse both cues and achieve compatable performances with 3d poses (from Kinect)

Evaluation on NTU RGB+D dataset

Largest dataset for 3D pose-based recognition task

Data	Method	Type	Year	Cross Subject	Cross View
estimated 3d pose using Kinect sensor (from depth)	Super Norm[illegible]	[illegible]-crafted	2014	31.82%	13.61%
	Deep R[illegible]	[illegible]NN	2016	[illegible]	[illegible]09%
	GCA-LSTM [26]	[illegible]	[illegible]017	[illegible]	[illegible]80%
	Clips + CNN + MTLN [20]	[illegible]	[illegible]	[illegible]	[illegible]83%
estimated 2d pose (from rgb)	S-P	[illegible]	[illegible]	[illegible]	[illegible]21%
pose estimation map (from rgb)	S-PEM	[illegible]	[illegible]018	72.75%	78.35%
2d pose + pose estimation map	Two Stream	CN[illegible]	2018	78.80%	84.21%

State-of-the-art method

State-of-the-art method based on CNN

Compatable

Pose estimation

They benefit each other!

56880 Videos; 60 actions; performed by 40 subjects; recorded from various views

Cross Subject: 40320 videos for training; 16560 videos for testing

Cross View: 37920 videos for training; 18960 videos for testing

[50] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. CVPR, 2014.

[35] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. CVPR, 2016.

[26] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention LSTM networks for 3D action recognition. CVPR, 2017.

[20] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3D action recognition. CVPR, 2017.

Summary

- Feature Representation
  - Joint Position + Heatmaps
- Classifier
  - Two-steam CNN
- Spatial-temporal modeling
  - Temporal evolution

Outline

- Problem space
- Datasets
  - RGB
  - RGB-D
- Skeleton-based approaches
- Video based approaches

TSN

- Motivation
  - discover the principles to design effective ConvNet architectures for action recognition

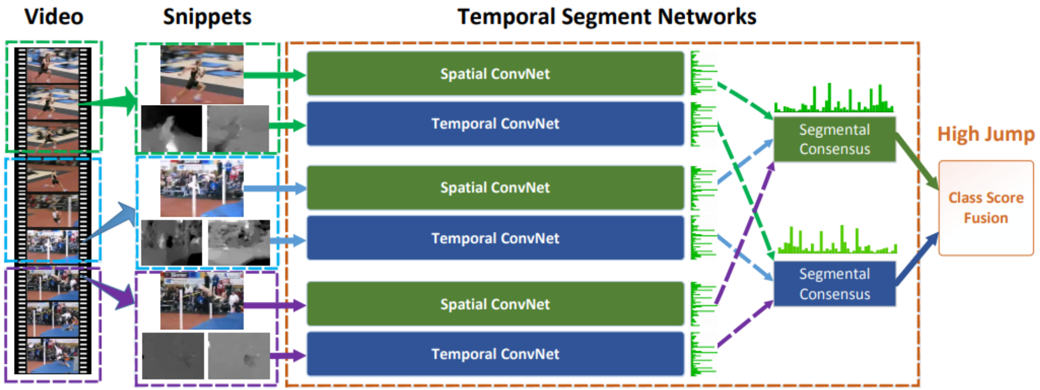


Fig. 1. Temporal segment network: One input video is divided into K segments and a short snippet is randomly selected from each segment. The class scores of different snippets are fused by an the segmental consensus function to yield segmental consensus, which is a video-level prediction. Predictions from all modalities are then fused to produce the final prediction. ConvNets on all snippets share parameters.

Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, ECCV2016

TSN

- Multiple-modalities
  - RGB images
  - Stacked optical flow
  - Warped optical flow

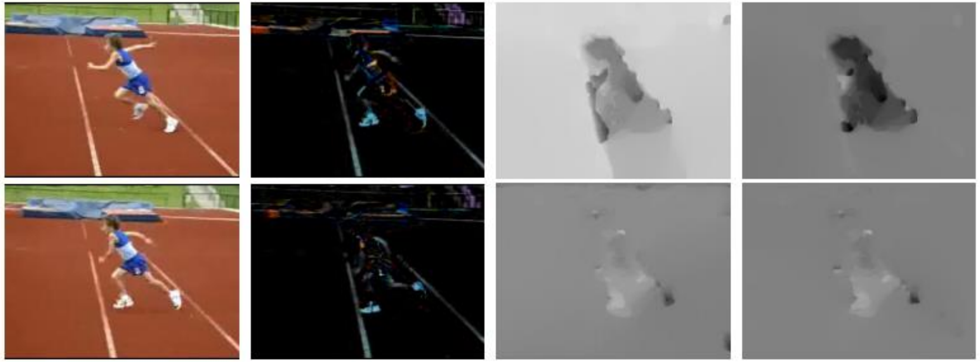


Fig. 2. Examples of four types of input modality: RGB images, RGB difference, optical flow fields (x,y directions), and warped optical flow fields (x,y directions)

Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, ECCV2016

TSN

- Experiments

Table 5. Component analysis of the proposed method on the UCF101 dataset (split 1). From left to right we add the components one by one. BN-Inception [23] is used as the ConvNet architecture.

Component	Basic Two-Stream [1]	Cross-Modality Pre-training	Partial BN with dropout	Temporal Segment Networks
Accuracy	90.0%	91.5	92.0%	93.5%

Table 6. Comparison of our method based on temporal segment network(TSN) with other state-of-the-art methods. We separately present the results of using two input modalities (RGB+Flow) and three input modalities (RGB+Flow+Warped Flow).

HMDB51		UCF101	
DT+MVSV [37]	55.9%	DT+MVSV [37]	83.5%
iDT+FV [2]	57.2%	iDT+FV [38]	85.9%
iDT+HSV [25]	61.1%	iDT+HSV [25]	87.9%
MoFAP [39]	61.7%	MoFAP [39]	88.3%
Two Stream [1]	59.4%	Two Stream [1]	88.0%
VideoDarwin [18]	63.7%	C3D (3 nets) [13]	85.2%
MPR [40]	65.5%	Two stream +LSTM [4]	88.6%
F_{ST}CN (SCI fusion) [28]	59.1%	F_{ST}CN (SCI fusion) [28]	88.1%
TDD+FV [5]	63.2%	TDD+FV [5]	90.3%
LTC [19]	64.8%	LTC [19]	91.7%
KVMF [41]	63.3%	KVMF [41]	93.1%
TSN (2 modalities)	68.5%	TSN (2 modalities)	94.0%
TSN (3 modalities)	**69.4**%	TSN (3 modalities)	**94.2**%

Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, ECCV2016

Summary for TSN

- Feature Representation
  - RGB, optical flow, …
- Classifier
  - CNN
- Spatial-temporal modeling
  - Weak

C3D

- Motivation
  - Is 3D convolution more suitable for action recognition?

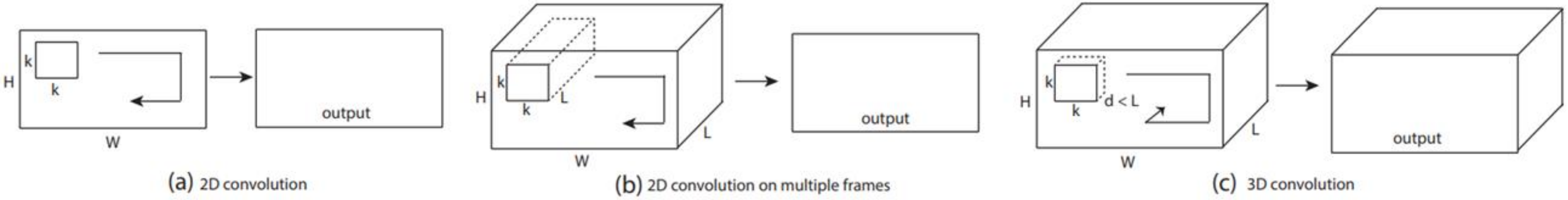


Figure 1. **2D and 3D convolution operations**. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

Learning Spatiotemporal Features with 3D Convolutional Networks, Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, ICCV2015

C3D

- Method

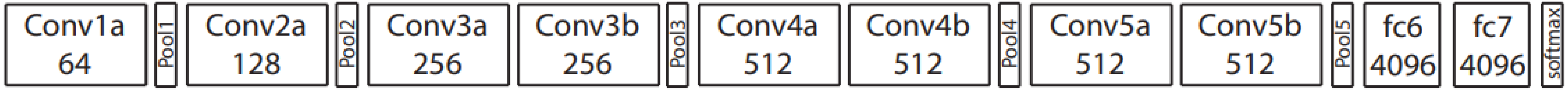


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling kernels are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

Learning Spatiotemporal Features with 3D Convolutional Networks, Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, ICCV2015

C3D

- Experiments

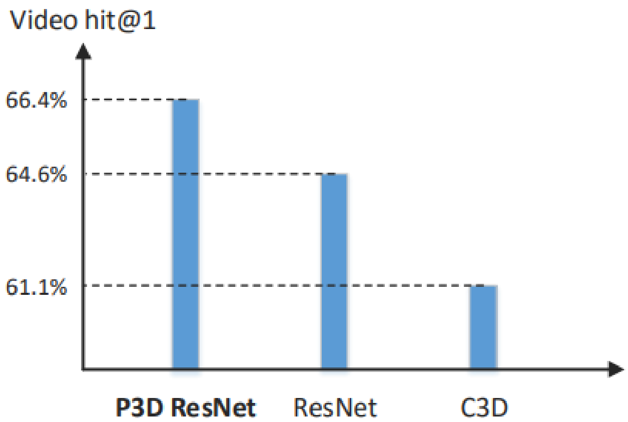
Dataset Task	Sport1M action recognition	UCF101 action recognition	ASLAN action similarity labeling	YUPENN scene classification	UMD scene classification	Object object recognition
Method	[29]	[39]([25])	[31]	[9]	[9]	[32]
Result	**90.8**	75.8 (89.1)	68.7	96.2	77.7	12.0
C3D	85.2	**85.2 (90.4)**	**78.3**	**98.1**	**87.7**	**22.3**

Table 1. **C3D compared to best published results**. C3D outperforms all previous best reported methods on a range of benchmarks except for Sports-1M and UCF101. On UCF101, we report accuracy for two groups of methods. The first set of methods use only RGB frame inputs while the second set of methods (in parentheses) use all possible features (e.g. optical flow, improved Dense Trajectory).

Learning Spatiotemporal Features with 3D Convolutional Networks, Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, ICCV2015

P3D

- Motivation
  - Expensive computational cost and memory demand for C3D



Method	Depth	Model size
C3D	11	321MB
ResNet	152	235MB
P3D ResNet	199	261MB

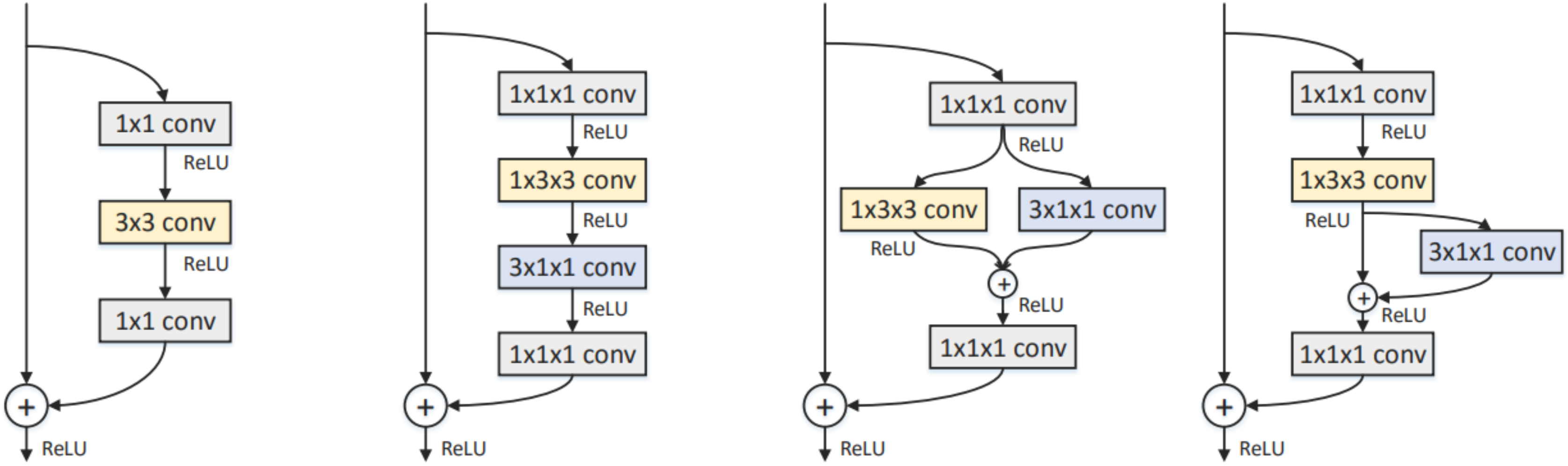
Figure 1. Comparisons of different models on Sports-1M dataset in terms of accuracy, model size and the number of layers.

Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks, Zhaofan Qiu,, Ting Yao,, and Tao Mei, ICCV2017

P3D

- Method

Figure 4. P3D ResNet by interleaving P3D-A, P3D-B and P3D-C.



(a) Residual Unit [7] (b) P3D-A (c) P3D-B (d) P3D-C

Figure 3. Bottleneck building blocks of Residual Unit and our Pseudo-3D.

Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks, Zhaofan Qiu,, Ting Yao,, and Tao Mei, ICCV2017

P3D

- Experiments

Table 2. Comparisons in terms of pre-train data, clip length, Top-1 clip-level accuracy and Top-1&5 video-level accuracy on Sports-1M.

Method	Pre-train Data	Clip Length	Clip hit@1	Video hit@1	Video hit@5
Deep Video (Single Frame) [10]	ImageNet1K	1	41.1%	59.3%	77.7%
Deep Video (Slow Fusion) [10]	ImageNet1K	10	41.9%	60.9%	80.2%
Convolutional Pooling [37]	ImageNet1K	120	70.8%	72.3%	90.8%
C3D [31]	–	16	44.9%	60.0%	84.4%
C3D [31]	I380K	16	46.1%	61.1%	85.2%
ResNet-152 [7]	ImageNet1K	1	46.5%	64.6%	86.4%
P3D ResNet (ours)	ImageNet1K	16	47.9%	66.4%	87.4%

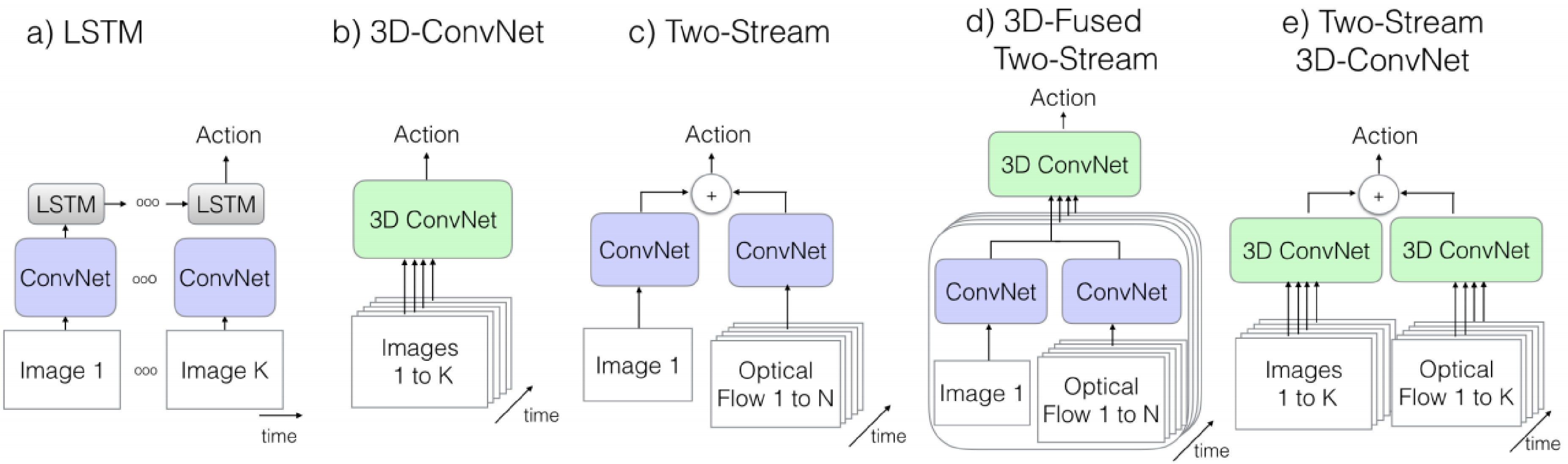
Table 3. Performance comparisons with the state-of-the-art methods on UCF101 (3 splits). TSN: Temporal Segment Networks [36]; TDD: Trajectory-pooled Deep-convolutional Descriptor [35]; IDT: Improved Dense Trajectory [34]. We group the approaches into three categories, i.e., end-to-end CNN architectures which are fine-tuned on UCF101 at the top, CNN-based video representation extractors with linear SVM classifier in the middle and approaches fused with IDT at the bottom. For the methods in the first direction, we report the performance of only taking frames and frames plus optical flow (in brackets) as inputs, respectively.

Method	Accuracy
End-to-end CNN architecture with fine-tuning	
Two-stream ConvNet [25]	73.0% (88.0%)
Factorized ST-ConvNet [29]	71.3% (88.1%)
Two-stream + LSTM [37]	82.6% (88.6%)
Two-stream fusion [6]	82.6% (92.5%)
Long-term temporal ConvNet [33]	82.4% (91.7%)
Key-volume mining CNN [39]	84.5% (93.1%)
ST-ResNet [4]	82.2% (93.4%)
TSN [36]	85.7% (94.0%)
CNN-based representation extractor + linear SVM	
C3D [31]	82.3%
ResNet-152	83.5%
P3D ResNet	**88.6%**
Method fusion with IDT	
IDT [34]	85.9%
C3D + IDT [31]	90.4%
TDD + IDT [35]	91.5%
ResNet-152 + IDT	92.0%
P3D ResNet + IDT	**93.7%**

Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks, Zhaofan Qiu,, Ting Yao,, and Tao Mei, ICCV2017

I3D

- Motivation
  - Efficient spatial-temporal representation



Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Joao Carreira, Andrew Zisserman, CVPR2017

I3D

- Method

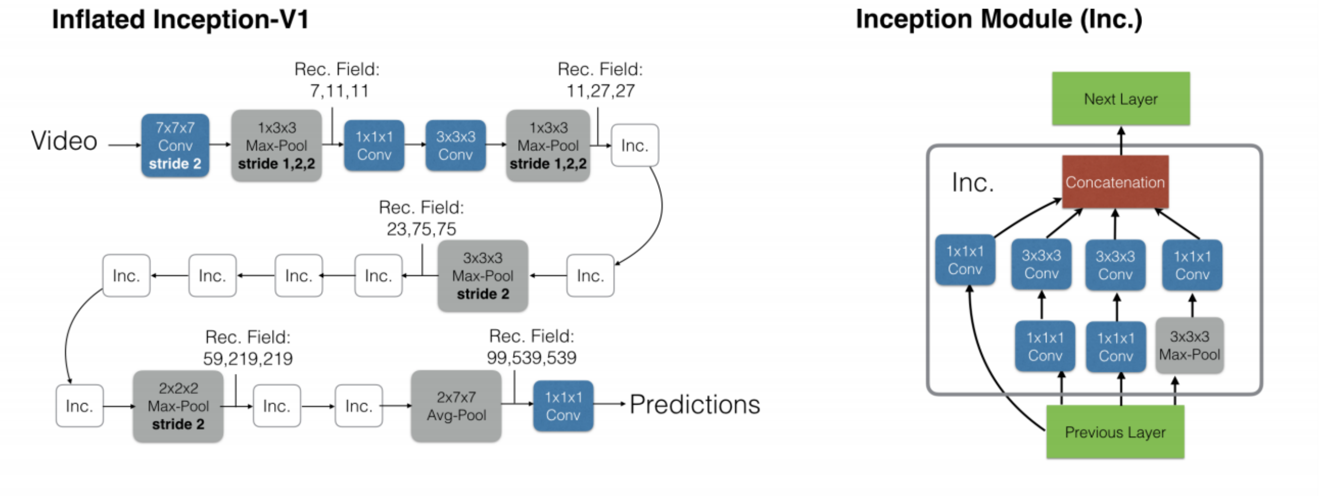


Figure 3. The Inflated Inception-V1 architecture (left) and its detailed inception submodule (right). The strides of convolution and pooling operators are 1 where not specified, and batch normalization layers, ReLu's and the softmax at the end are not shown. The theoretical sizes of receptive field sizes for a few layers in the network are provided in the format "time,x,y" – the units are frames and pixels. The predictions are obtained convolutionally in time and averaged.

Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Joao Carreira, Andrew Zisserman, CVPR2017

I3D

- Experiments

Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

Table 2. Architecture comparison: (left) training and testing on split 1 of UCF-101; (middle) training and testing on split 1 of HMDB-51; (right) training and testing on Kinetics. All models are based on ImageNet pre-trained Inception-v1, except 3D-ConvNet, a C3D-like [31] model which has a custom architecture and was trained here from scratch. Note that the Two-Stream architecture numbers on individual RGB and Flow streams can be interpreted as a simple baseline which applies a ConvNet independently on 25 uniformly sampled frames then averages the predictions.

Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

Table 3. Performance training and testing on Kinetics with and without ImageNet pretraining. Numbers in brackets () are the Top-5 accuracy, all others are Top-1.

Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, Joao Carreira, Andrew Zisserman, CVPR2017

Summary

- Feature Representation
  - RGB video frames
- Classifier
  - 3D convolution
- Spatial-temporal modeling
  - 3D convolution

SlowFast

- Motivation
  - Combine spatial semantics and motion at fine temporal resolution

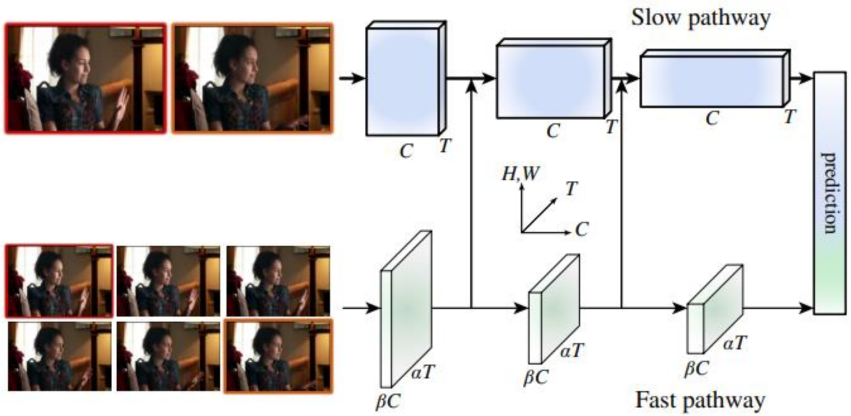


Figure 1. **A SlowFast network** has a low frame rate, low temporal resolution *Slow* pathway and a high frame rate, $\alpha\times$ higher temporal resolution *Fast* pathway. The Fast pathway is lightweight by using a fraction (β, *e.g.*, 1/8) of channels. Lateral connections fuse them. This sample is from the AVA dataset [17] (annotation: hand wave).

SlowFast Networks for Video Recognition, Christoph Feichtenhofer Haoqi Fan Jitendra Malik Kaiming He, ICCV 2019

SlowFast

- Method

(i) *Time-to-channel*: We reshape and transpose $\{\alpha T, S^2, \beta C\}$ into $\{T, S^2, \alpha\beta C\}$, meaning that we pack all α frames into the channels of one frame.

(ii) *Time-strided sampling*: We simply sample one out of every α frames, so $\{\alpha T, S^2, \beta C\}$ becomes $\{T, S^2, \beta C\}$.

(iii) *Time-strided convolution*: We perform a 3D convolution of a 5×1^2 kernel with $2\beta C$ output channels and stride $= \alpha$.

The output of the lateral connections is fused into the Slow pathway by summation or concatenation.

stage	*Slow* pathway	*Fast* pathway	output sizes $T\times S^2$
raw clip	-	-	64×224^2
data layer	stride 16, 1^2	stride **2**, 1^2	*Slow* : 4×224^2 *Fast* : 32×224^2
conv_1	1×7^2, 64 stride 1, 2^2	$\underline{5\times 7^2}$, 8 stride 1, 2^2	*Slow* : 4×112^2 *Fast* : 32×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 4×56^2 *Fast* : 32×56^2
res_2	$\begin{bmatrix} 1\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	$\begin{bmatrix} \underline{3\times 1^2}, 8 \\ 1\times 3^2, 8 \\ 1\times 1^2, 32 \end{bmatrix}\times 3$	*Slow* : 4×56^2 *Fast* : 32×56^2
res_3	$\begin{bmatrix} 1\times 1^2, 128 \\ 1\times 3^2, 128 \\ 1\times 1^2, 512 \end{bmatrix}\times 4$	$\begin{bmatrix} \underline{3\times 1^2}, 16 \\ 1\times 3^2, 16 \\ 1\times 1^2, 64 \end{bmatrix}\times 4$	*Slow* : 4×28^2 *Fast* : 32×28^2
res_4	$\begin{bmatrix} \underline{3\times 1^2}, 256 \\ 1\times 3^2, 256 \\ 1\times 1^2, 1024 \end{bmatrix}\times 6$	$\begin{bmatrix} \underline{3\times 1^2}, 32 \\ 1\times 3^2, 32 \\ 1\times 1^2, 128 \end{bmatrix}\times 6$	*Slow* : 4×14^2 *Fast* : 32×14^2
res_5	$\begin{bmatrix} \underline{3\times 1^2}, 512 \\ 1\times 3^2, 512 \\ 1\times 1^2, 2048 \end{bmatrix}\times 3$	$\begin{bmatrix} \underline{3\times 1^2}, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	*Slow* : 4×7^2 *Fast* : 32×7^2
global average pool, concate, fc			# classes

Table 1. **An example instantiation of the SlowFast network**. The dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes. Strides are denoted as {temporal stride, spatial stride^2}. Here the speed ratio is $\alpha = 8$ and the channel ratio is $\beta = 1/8$. τ is 16. The green colors mark *higher* temporal resolution, and orange colors mark *fewer* channels, for the Fast pathway. Non-degenerate temporal filters are underlined. Residual blocks are shown by brackets. The backbone is ResNet-50.

SlowFast

- Experiments

model	pretrain	top-1	top-5	GFLOPs×views
I3D [2]	-	71.9	90.1	108 × N/A
StNet-IRv2 RGB [18]	ImgNet+Kin400	79.0	N/A	N/A
SlowFast 4×16, R50	-	78.8	94.0	36.1 × 30
SlowFast 8×8, R50	-	79.9	94.5	65.7 ×30
SlowFast 8×8, R101	-	80.4	94.8	106 × 30
SlowFast 16×8, R101	-	81.1	95.1	213 × 30
SlowFast 16×8, R101+NL	-	**81.8**	**95.1**	234 × 30

Table 3. **Comparison with the state-of-the-art on Kinetics-600.** SlowFast models the same as in Table 2.

model	pretrain	mAP	GFLOPs×views
CoViAR, R-50 [55]	ImageNet	21.9	N/A
Asyn-TF, VGG16 [39]	ImageNet	22.4	N/A
MultiScale TRN [58]	ImageNet	25.2	N/A
Nonlocal, R101 [52]	ImageNet+Kinetics400	37.5	544 × 30
STRG, R101+NL [53]	ImageNet+Kinetics400	39.7	630 × 30
our baseline (Slow-only)	Kinetics-400	39.0	187 × 30
SlowFast	Kinetics-400	42.1	213 × 30
SlowFast, +NL	Kinetics-400	42.5	234 × 30
SlowFast, +NL	Kinetics-600	**45.2**	234 × 30

Table 4. **Comparison with the state-of-the-art on Charades.** All our variants are based on $T\times\tau = 16\times8$, R-101.

model	flow	pretrain	top-1	top-5	GFLOPs×views
I3D [3]		ImageNet	72.1	90.3	108 × N/A
Two-Stream I3D [3]	✓	ImageNet	75.7	92.0	216 × N/A
S3D-G [57]	✓	ImageNet	77.2	93.0	143 × N/A
Nonlocal R50 [52]		ImageNet	76.5	92.6	282 × 30
Nonlocal R101 [52]		ImageNet	77.7	93.3	359 × 30
R(2+1)D Flow [47]	✓	-	67.5	87.2	152 × 115
STC [7]		-	68.7	88.5	N/A × N/A
ARTNet [50]		-	69.2	88.3	23.5 × 250
S3D [57]		-	69.4	89.1	66.4 × N/A
ECO [59]		-	70.0	89.4	N/A × N/A
I3D [3]	✓	-	71.6	90.0	216 × N/A
R(2+1)D [47]		-	72.0	90.0	152 × 115
R(2+1)D [47]	✓	-	73.9	90.9	304 × 115
SlowFast 4×16, R50		-	75.6	92.1	36.1 × 30
SlowFast 8×8, R50		-	77.0	92.6	65.7 × 30
SlowFast 8×8, R101		-	77.9	93.2	106 × 30
SlowFast 16×8, R101		-	78.9	93.5	213 × 30
SlowFast 16×8, R101+NL		-	**79.8**	**93.9**	234 × 30

Table 2. **Comparison with the state-of-the-art on Kinetics-400.** In the last column, we report the inference cost with a single "view" (temporal clip with spatial crop) × the numbers of such views used. The SlowFast models are with different input sampling ($T\times\tau$) and backbones (R-50, R-101, NL). "N/A" indicates the numbers are not available for us.

SlowFast Networks for Video Recognition, Christoph Feichtenhofer Haoqi Fan Jitendra Malik Kaiming He, ICCV 2019

SlowFast

- Experiments

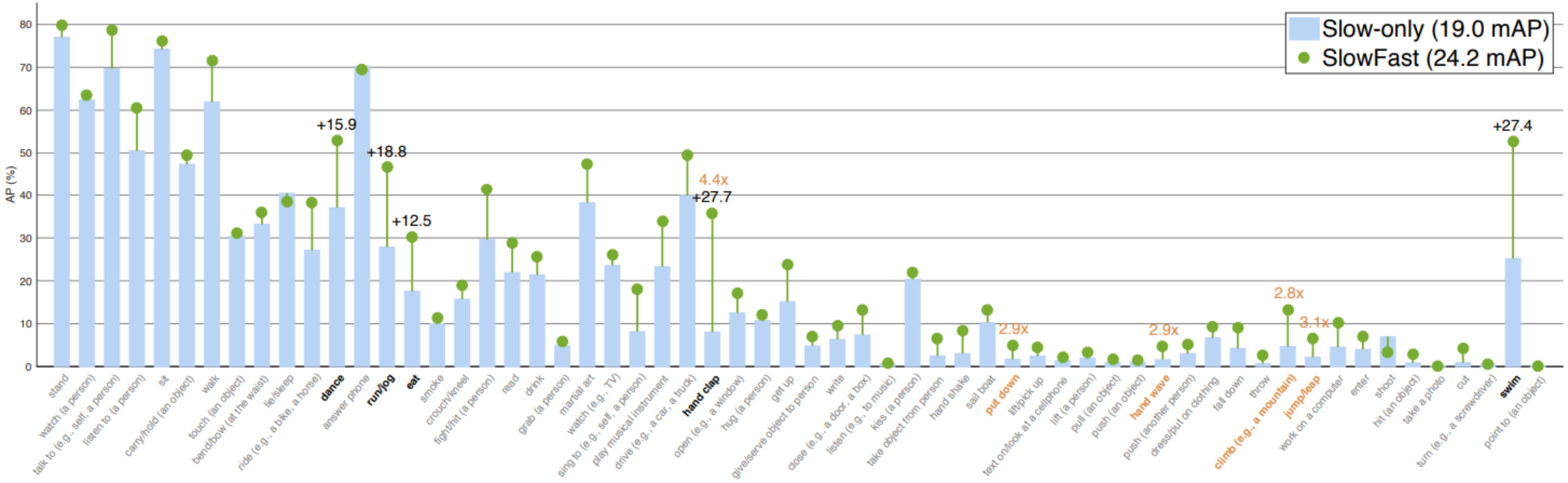


Figure 4. **Per-category AP on AVA**: a Slow-only baseline (19.0 mAP) *vs*. its SlowFast counterpart (24.2 mAP). The highlighted categories are the 5 highest absolute increase (**black**) or 5 highest relative increase with Slow-only AP > 1.0 (**orange**). Categories are sorted by number of examples. Note that the SlowFast instantiation in this ablation is not our best-performing model.

SlowFast Networks for Video Recognition, Christoph Feichtenhofer Haoqi Fan Jitendra Malik Kaiming He, ICCV 2019

Summary for SlowFast

- Feature Representation
  - RGB Frames with two path (slow & fast)
- Classifier
  - 3D convolution
- Spatial-temporal modeling
  - 3D convolution

Human Centric Spatio-Temporal Action Localization

- Motivation
  - Combine spatial & temporal information

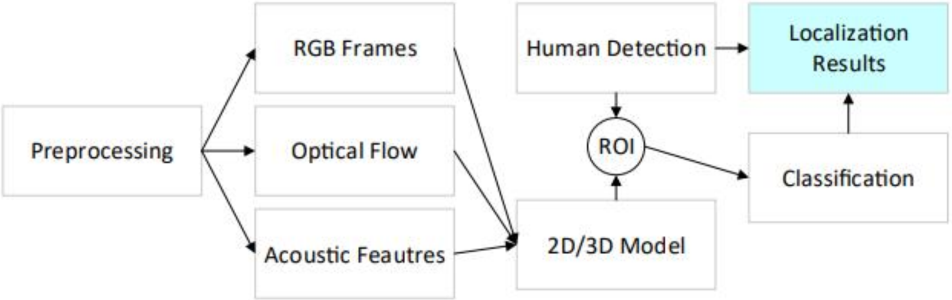


Fig. 1. The designed framework in our method. We split the spatio-temporal action localization into two subtasks, including human detection and action classification. Given the detections, we mainly focus on extracting multi vision cues, such as appearance information, motion information, and acoustic features. By applying ROI pooling, we can integrate the results from different models.

Human Centric Spatio-Temporal Action Localization, Jiang etc, http://www.skicyyu.org/AVA/AVA_report.pdf

Human Centric Spatio-Temporal Action Localization

- Motivation
  - Combine spatial & temporal information

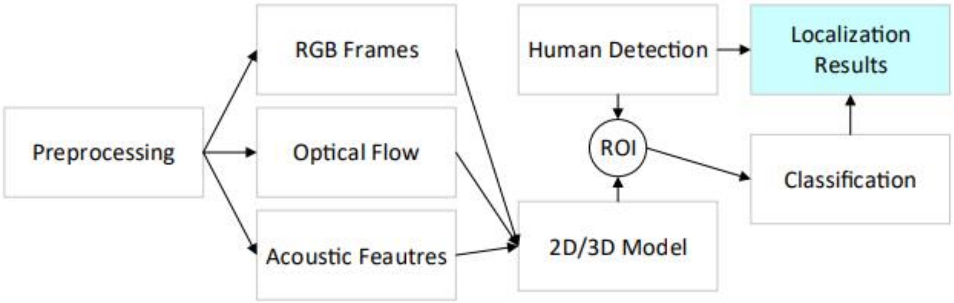


Fig. 1. The designed framework in our method. We split the spatio-temporal action localization into two subtasks, including human detection and action classification. Given the detections, we mainly focus on extracting multi vision cues, such as appearance information, motion information, and acoustic features. By applying ROI pooling, we can integrate the results from different models.

Human Centric Spatio-Temporal Action Localization, Jiang etc, http://www.skicyyu.org/AVA/AVA_report.pdf

Human Centric Spatio-Temporal Action Localization

- Method

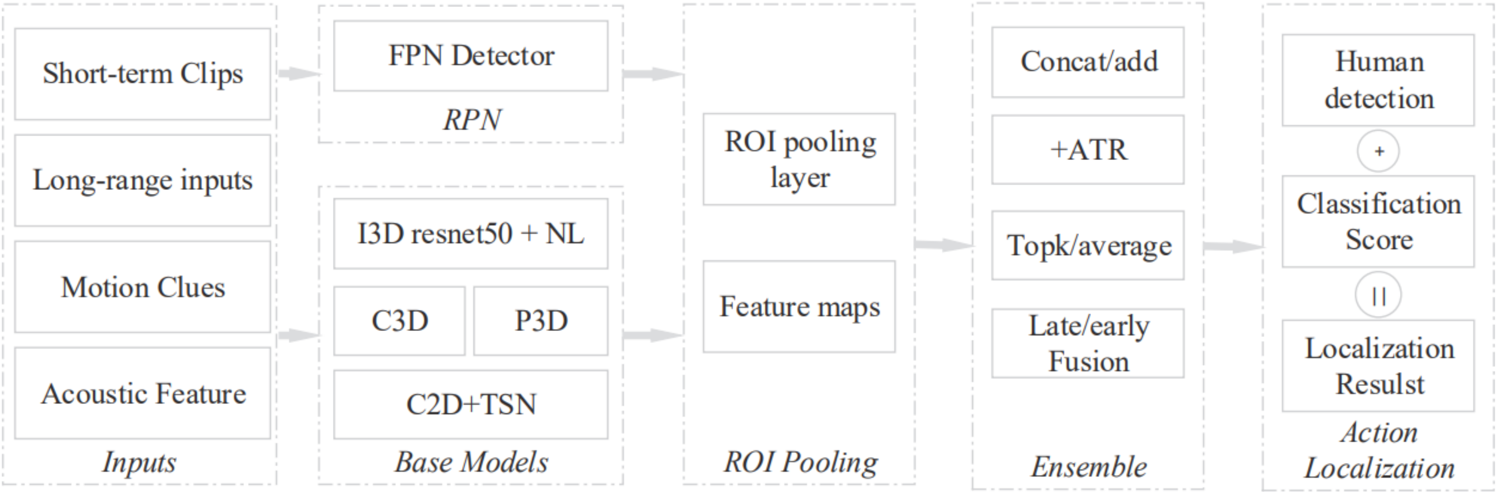


Fig. 2. The overview of our method. First, we explore different vision cues, which are respectively fed into RPN and feature extractors. Then we apply ROI pooling operation based on the proposal regions and the corresponding feature maps. After that, we explore different integration strategies on the applied models. Finally, we calculate the location results by considering the classification results and proposal regions.

Human Centric Spatio-Temporal Action Localization, Jiang etc, http://www.skicyyu.org/AVA/AVA_report.pdf

Human Centric Spatio-Temporal Action Localization

- Experiments

TABLE I
RESULTS ON VALIDATION SET.

Model	Input	Modality	Operation	mAP (%)
Faster-RCNN [4]	(3, 40(RGB)+40(Flow), 360, 400)	RGB + Flow	-	16.2
i3d resnet50 + NL	(3, 20, 224, 224)	RGB	-	19.33
	(3, 20, 224, 224)	RGB	ATR	20.01
	(3, 40, 224, 224)	RGB	40 clips	19.37
	(3, 20, 360, 400)	RGB	(360,400) size	19.86
	(3, 20(RGB)+20(Flow), 224, 224)	RGB + Flow	add	21.66
P3D199	(3, 20(RGB)+20(Flow), 224, 224)	RGB + Flow	-	17.87
resnet152	(3, 20, 224, 224)	RGB	TSN	14.68
artnet18	(3, 20, 224, 224)	RGB	-	16.67
Vgg16	-	Audio	-	6.5
Ensemble(Vison Only)				**25.63**
Ensemble (Full)				**25.75**

Human Centric Spatio-Temporal Action Localization, Jiang etc, http://www.skicyyu.org/AVA/AVA_report.pdf

Conclusion

- Feature Representation is important for Action Recognition
  - Skeleton
    - Pros: Simple and efficient to compute, good results
    - Cons: skeleton itself may not be accurate
  - Two-Steam
    - Pros: easy to deploy
    - Cons: spatial and temporal are decoupled
  - 3D Convolution
    - Pros: promising results to model both spatial and temporal info
    - Cons: data hungray