Learning Transferable Self-attentive Representations for Action Recognition in Untrimmed Videos with Weak Supervision

Xiao-Yu Zhang[1] Haichao Shi[1,2,4] Changsheng Li[3,4]

Kai Zheng[3,4] Xiaobin Zhu[5] Lixin Duan[3,4]

[1]Institute of Information Engineering, Chinese Academy of Sciences

[2]School of Cyber Security, University of Chinese Academy of Sciences

[3]University of Electronic Science and Technology of China

[4]Youedata Co., Ltd., Beijing

[5]Beijing Technology and Business University

Presenter: Haichao Shi

Outline

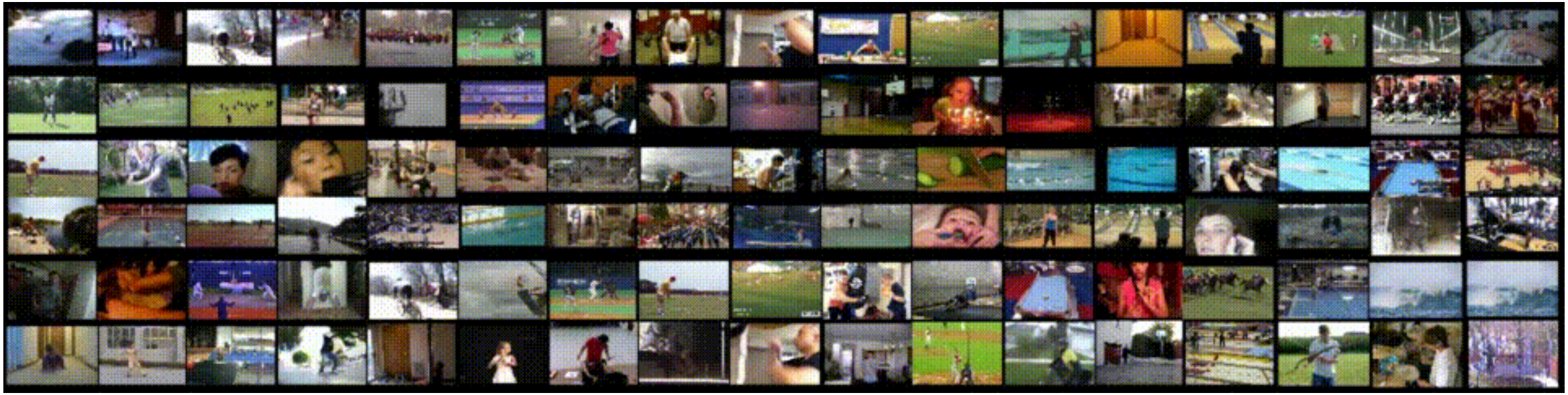
- 1. Introduction
- 2. Our Method (TSRNet)
- 3. Evaluation
- 4. Conclusion

Outline

- 1. Introduction
- 2. Our Method (TSRNet)
- 3. Evaluation
- 4. Conclusion

Motivation

Action Recognition in videos



- 1. Action recognition "in the lab" : KTH, Weizmann etc.
- 2. Action recognition "in TV, Movies" :UCF Sports, Holloywood etc.
- 3. Action recognition "in Web Videos" :HMDB, UCF101, THUMOS, ActivityNet etc.

Motivation

Opportunities and Challenges

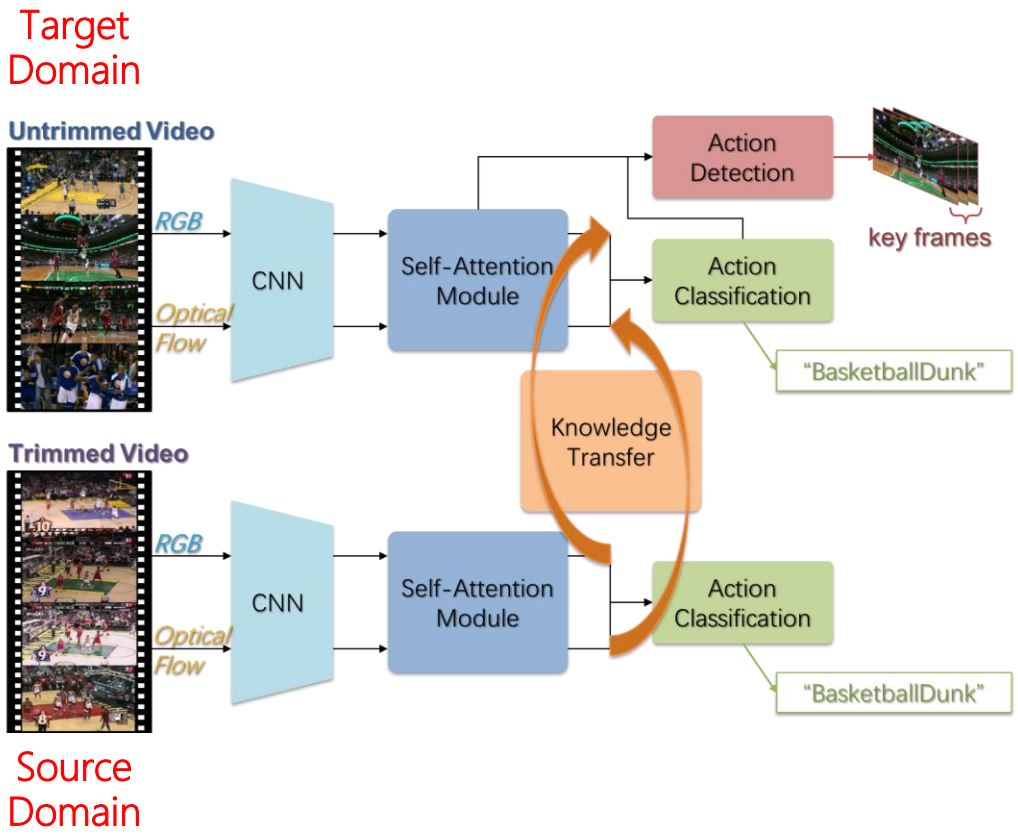
- Opportunities
  - Videos provide huge and rich data for visual learning
  - Action is important in motion perception and has many applications

- Challenges
  - Temporal models and representations
  - High computational and memory cost
  - Noisy and weakly labels

Overview

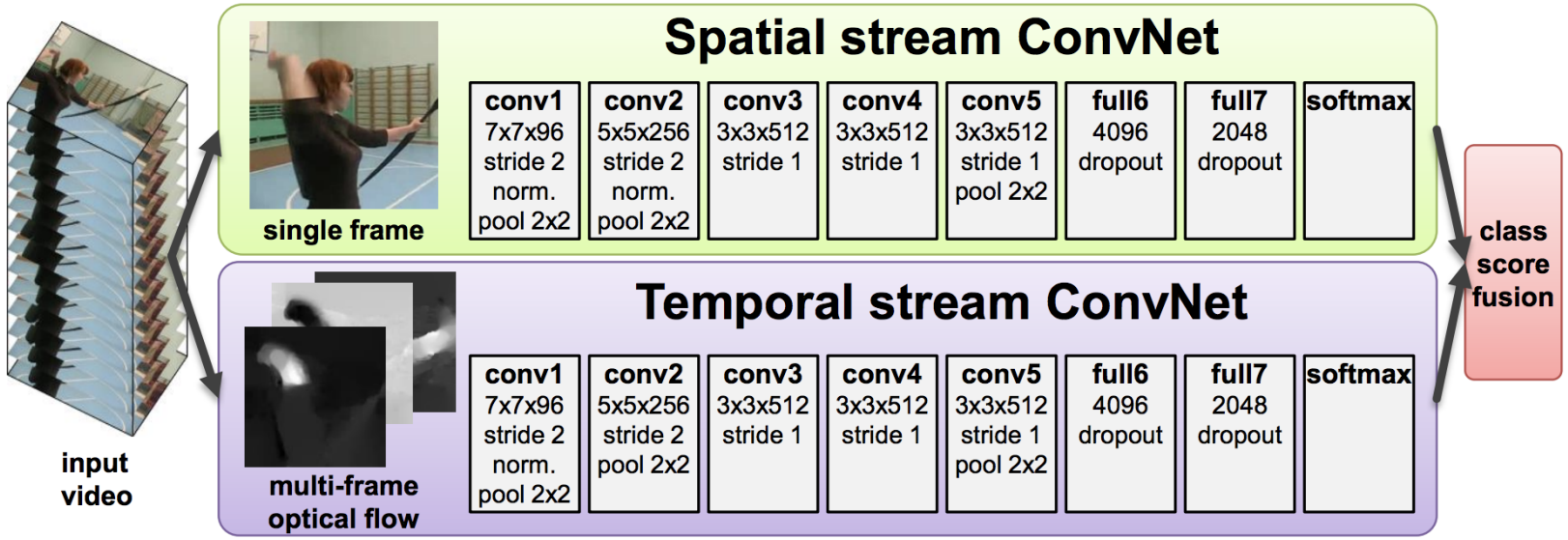
Domain adaptation and Method

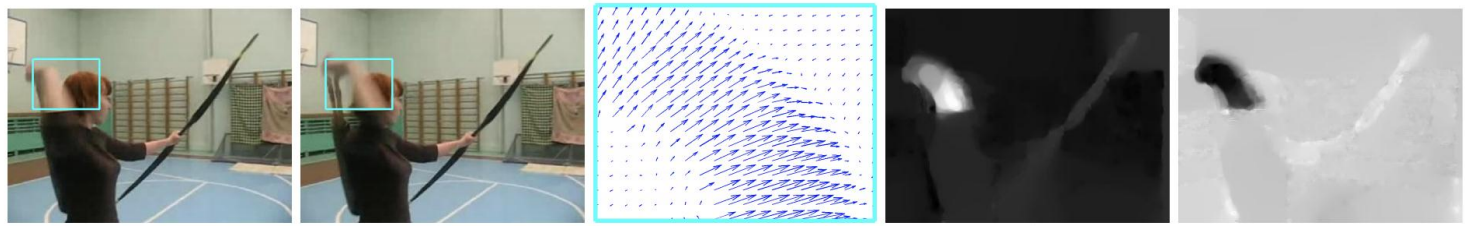
- Domain adaptation
  - Different data distributions in the source and target domain
  - Data from source and target domains have low level distribution difference and similar high level distributions
- Method
  - A self-attention module for each domain: capture specific domain properties.
  - A transfer module: capture representations shared by domains.



Related Work

Action Recognition - two stream CNN

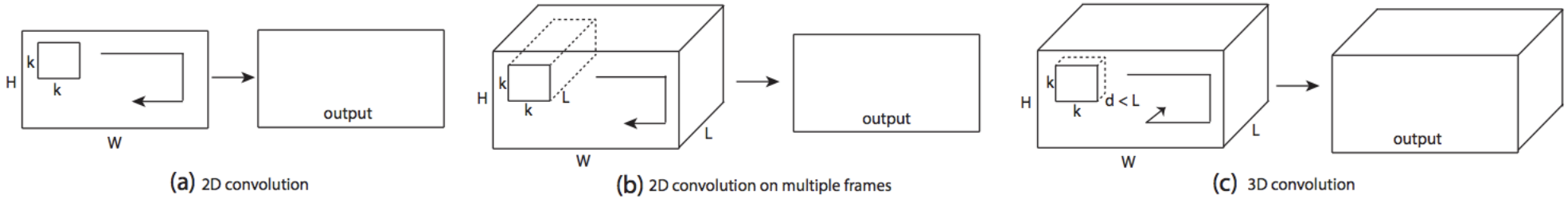




Karen Simonyan and Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, in NIPS, 2014.

Related Work

Action Recognition – C3D



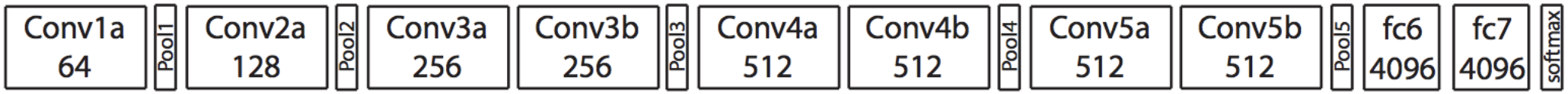
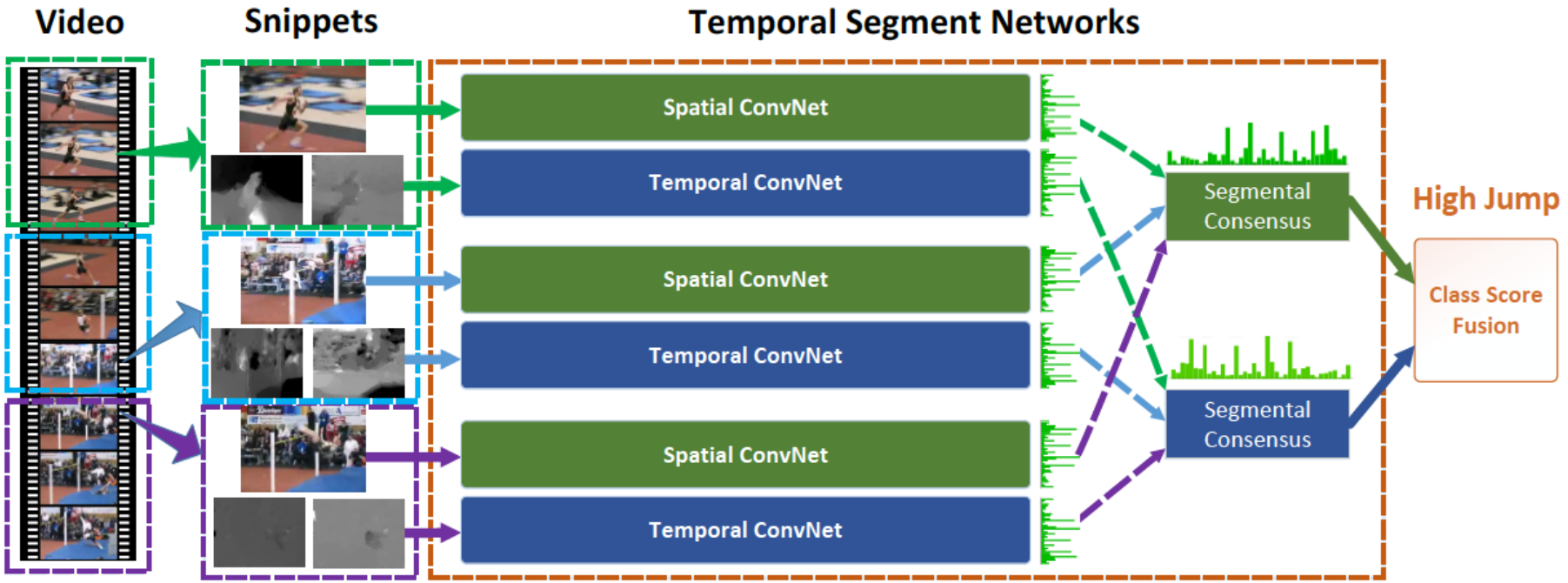


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling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

Du Tran et al. Learning Spatiotemporal Features with 3D Convolutional Networks, in ICCV, 2015.

Related Work

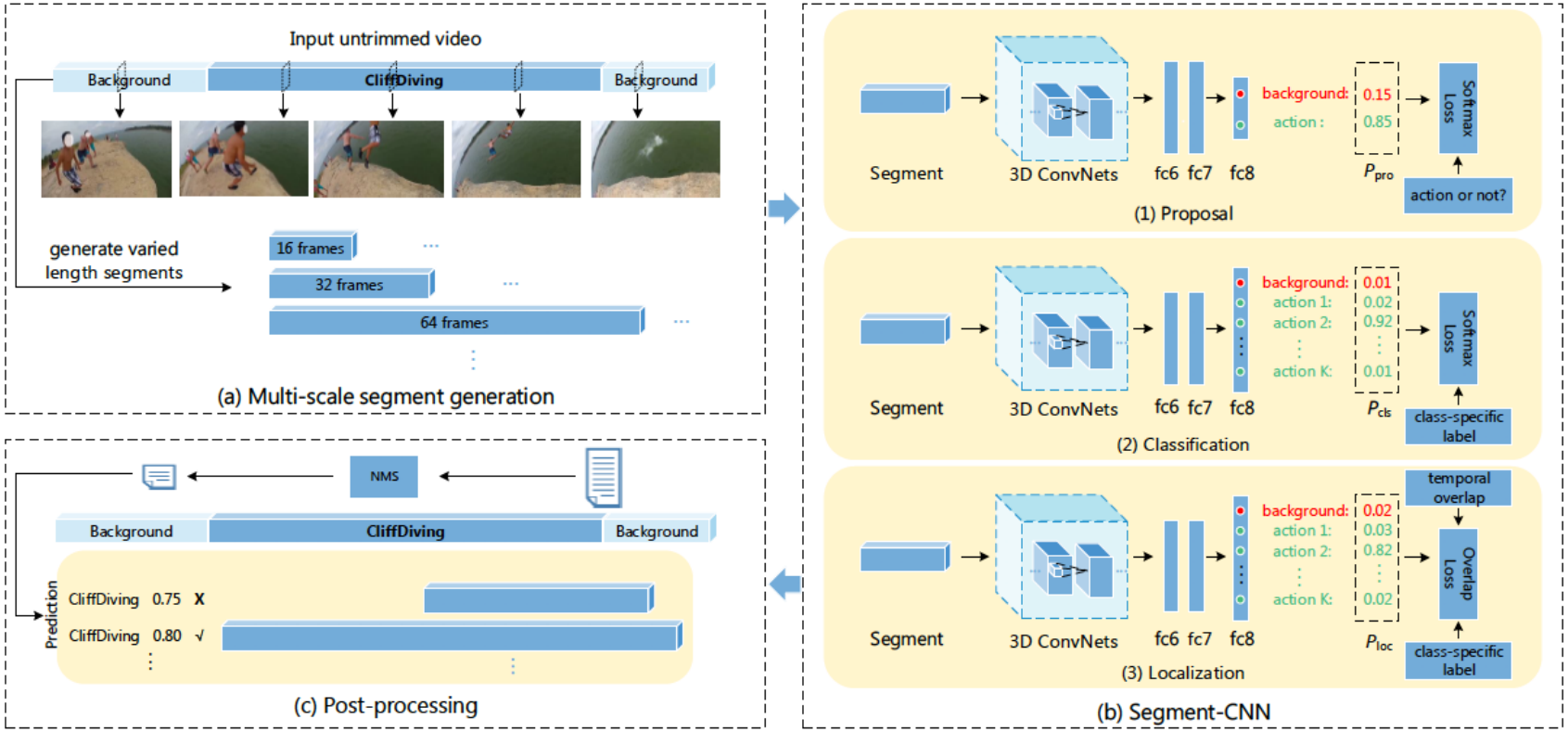
Action Recognition - Temporal Segment Network(TSN)



Limin Wang, Yuanjun Xiong et al. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, in ECCV 2016

Related Work

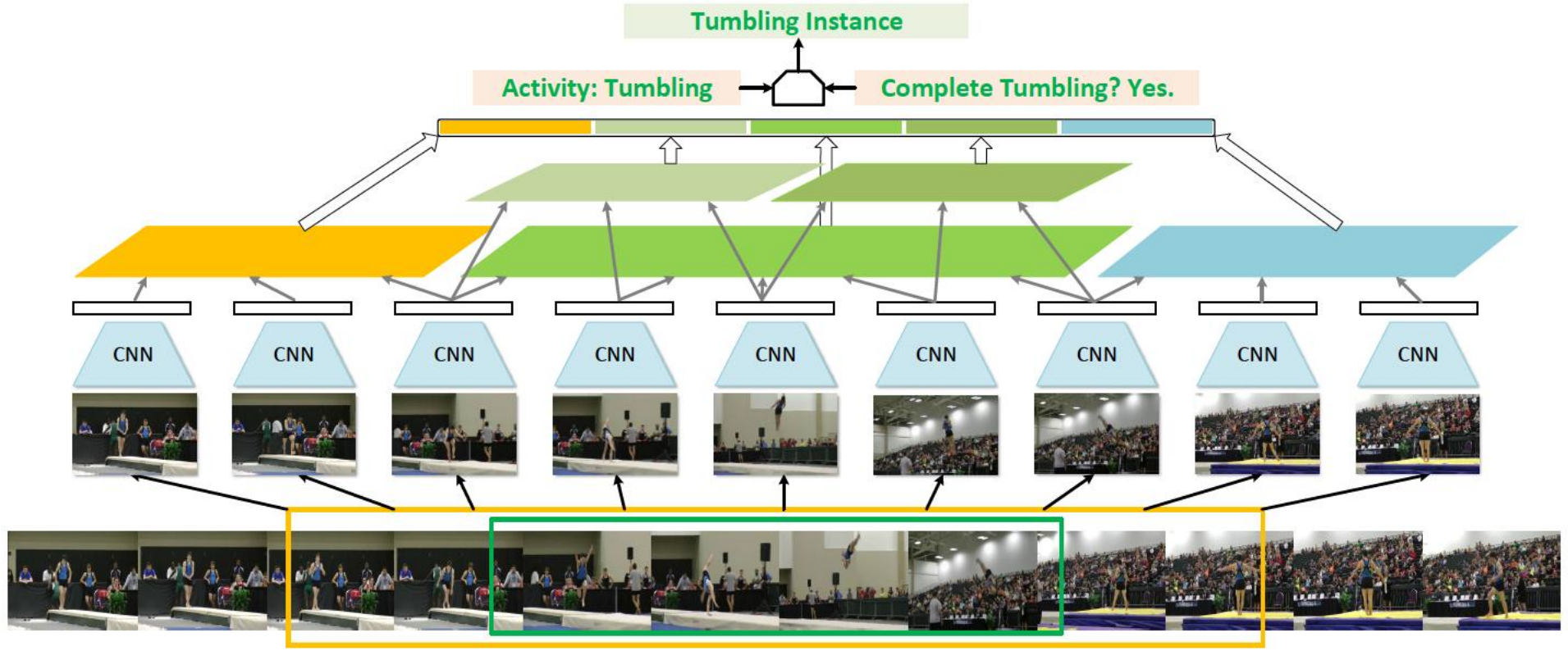
Temporal Action Detection – Multi stage CNNs(Fully-supervised)



Zheng Shou et al. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs, in CVPR 2016

Related Work

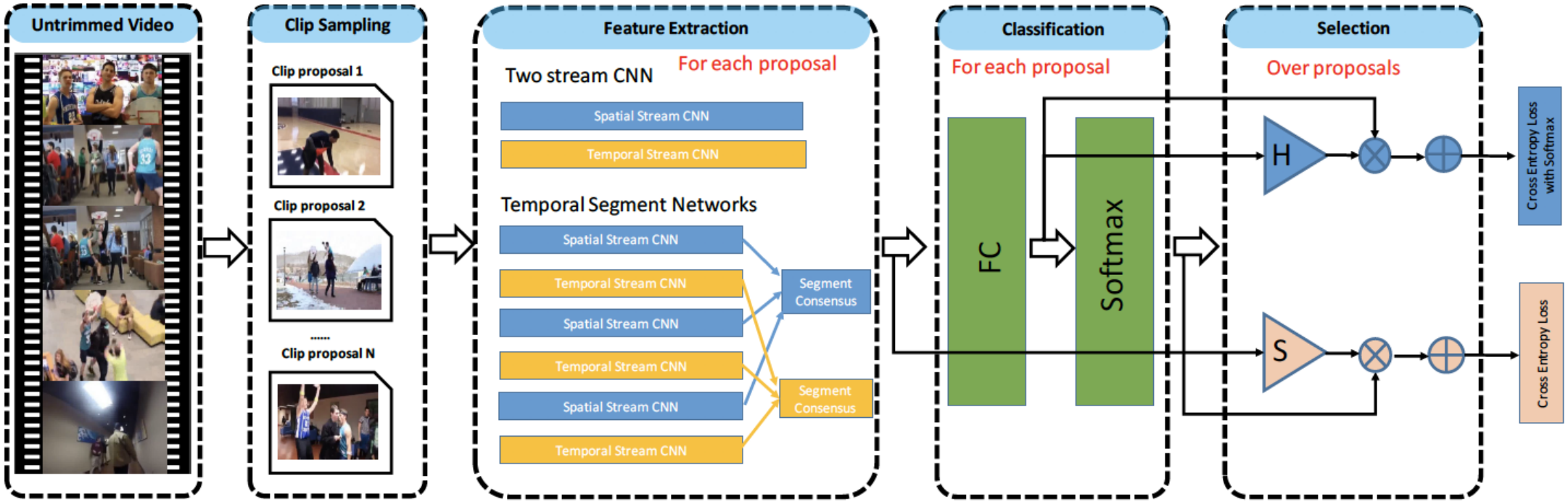
- **Temporal Action Detection – SSN(Fully-supervised)**



Yue Zhao et al. Temporal Action Detection with Structured Segment Networks, in ICCV 2016

Related Work

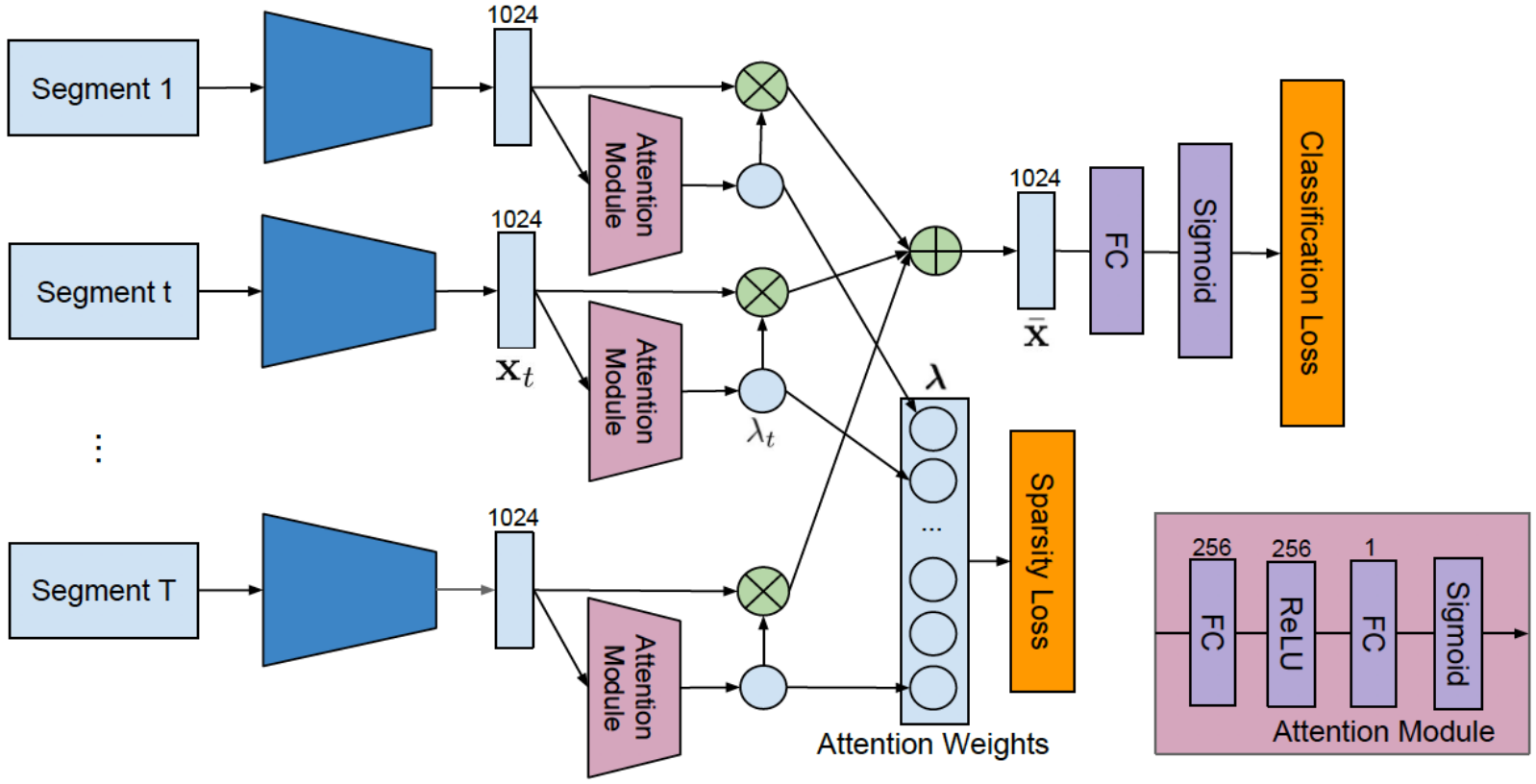
Temporal Action Detection – UntrimmedNets(Weakly-supervised)



Limin Wang et al. UntrimmedNets forWeakly Supervised Action Recognition and Detection, in CVPR 2017

Related Work

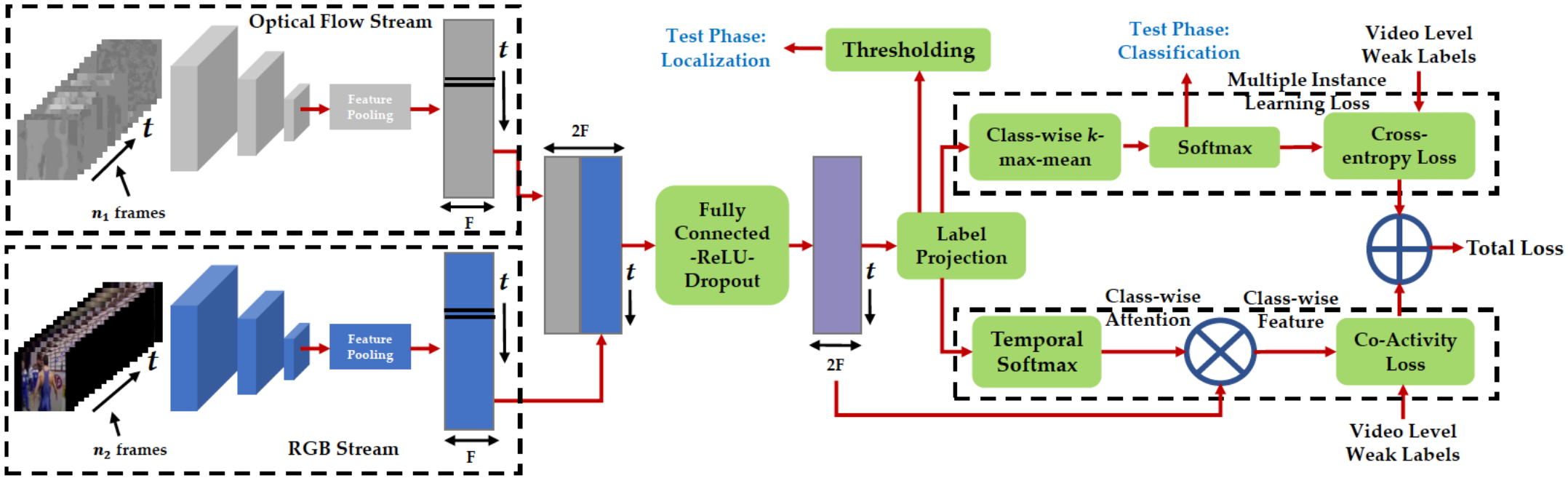
Temporal Action Detection – STPN(Weakly-supervised)



Phuc Nguyen et al. Weakly Supervised Action Localization by Sparse Temporal Pooling Network, in CVPR 2018

Related Work

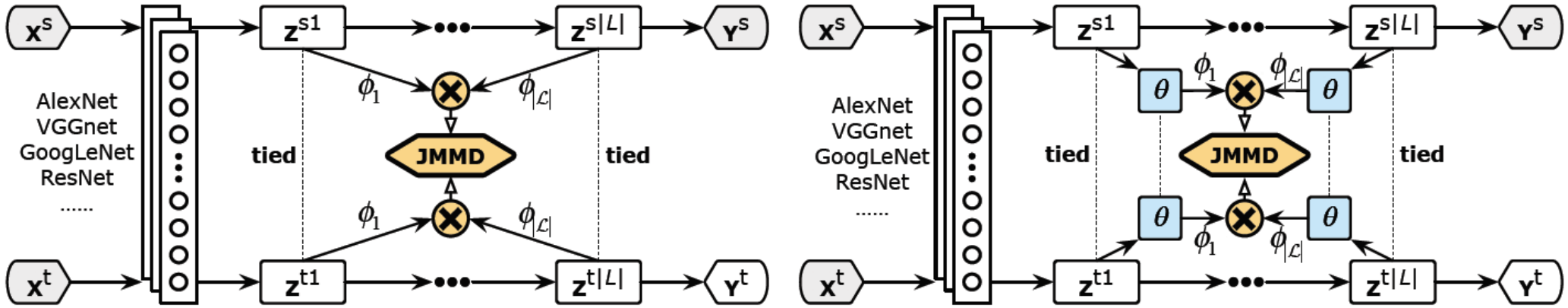
Temporal Action Detection – W-TALC(Weakly-supervised)



Sujoy Paul et al. W-TALC: Weakly-supervised Temporal Activity Localization and Classification, in ECCV 2018

Related Work

Transfer Learning – Joint Adaptation Networks(JAN)



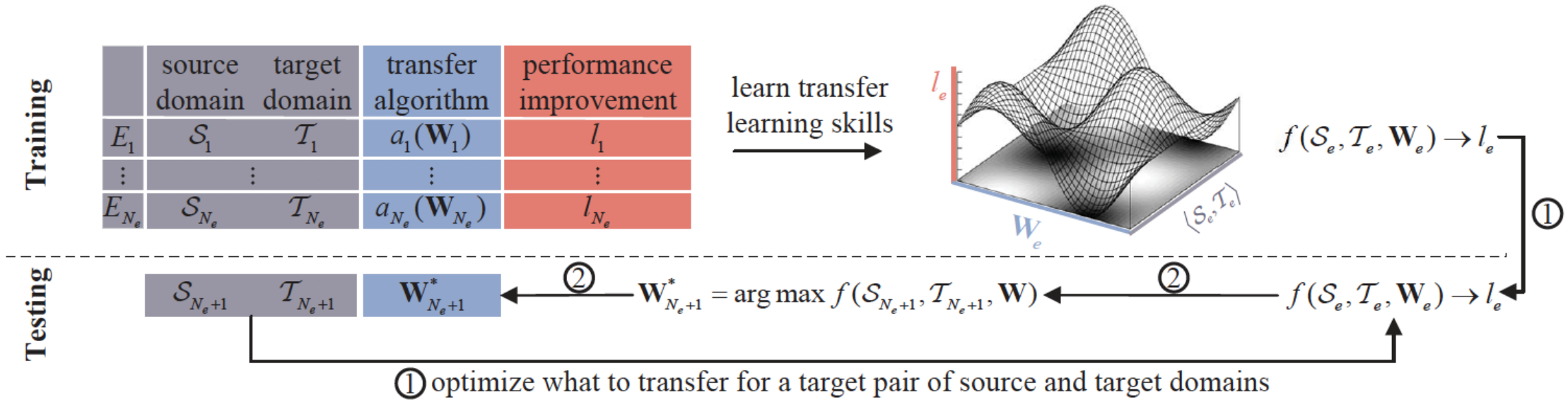
(a) Joint Adaptation Network (JAN)

(b) Adversarial Joint Adaptation Network (JAN-A)

Minsheng Long et al. Deep Transfer Learning with Joint Adaptation Networks, in ICML 2017

Related Work

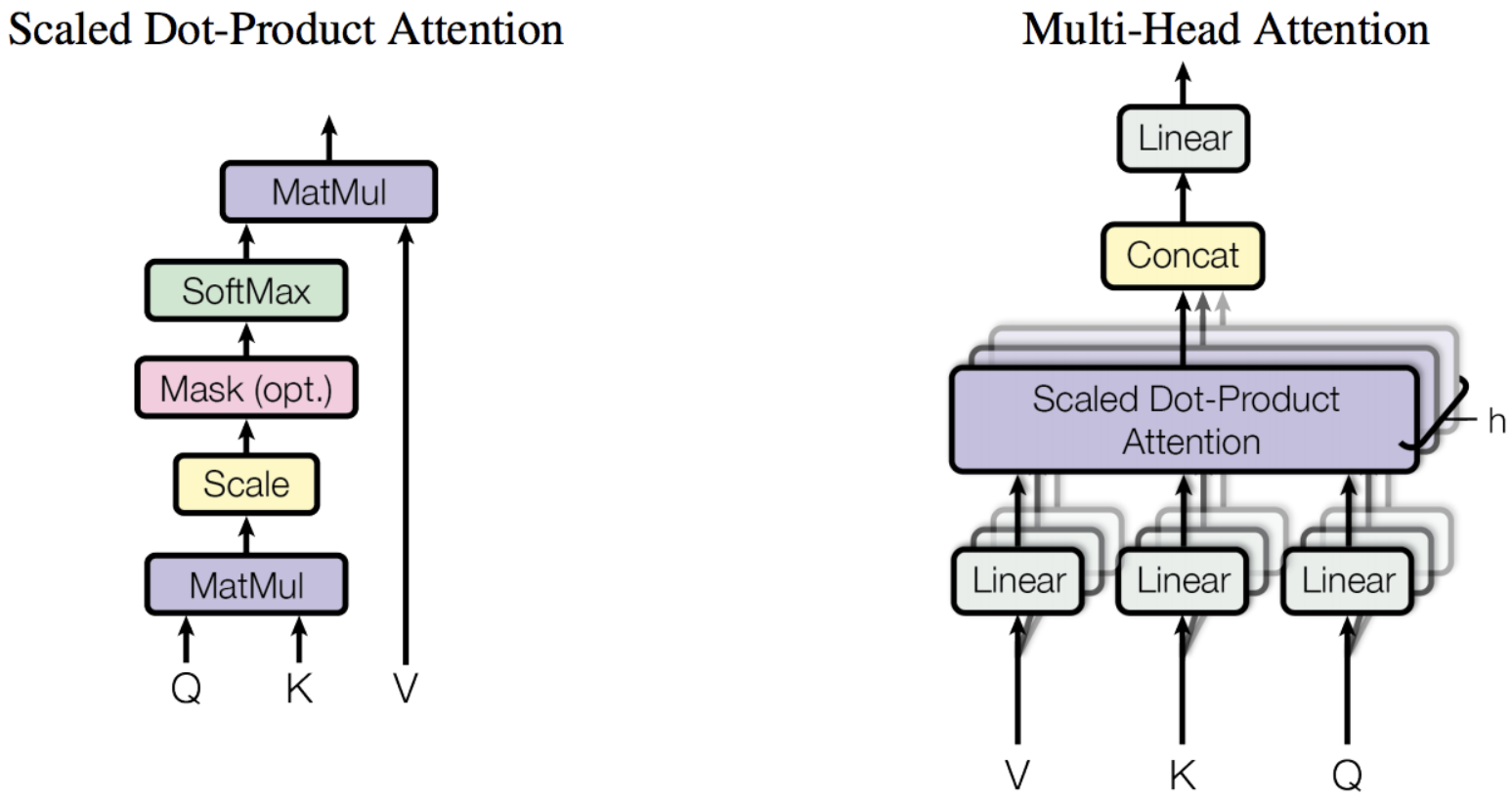
Transfer Learning – Learning to Transfer



Ying Wei et al. Transfer Learning via Learning to Transfer, in ICML 2018

Related Work

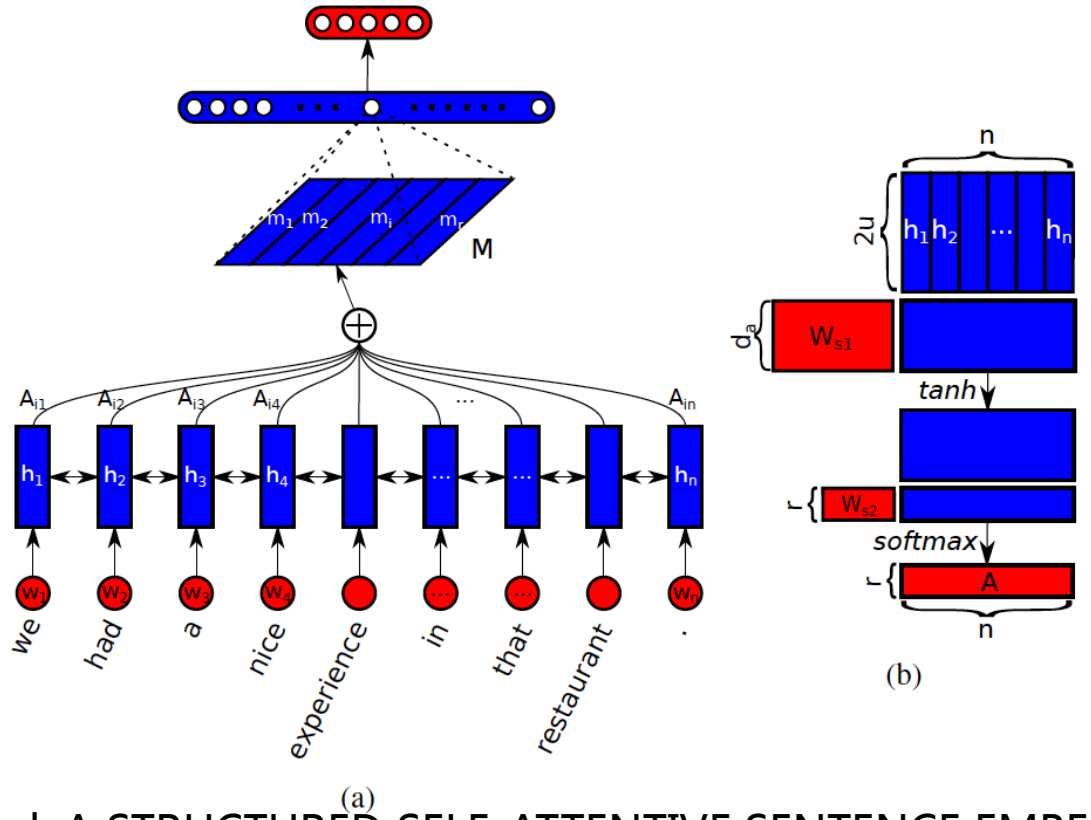
Attention Mechanism – Attention



Ashish Vaswani et al. Attention Is All You Need, in NIPS 2017

Related Work

Attention Mechanism – Self-attention

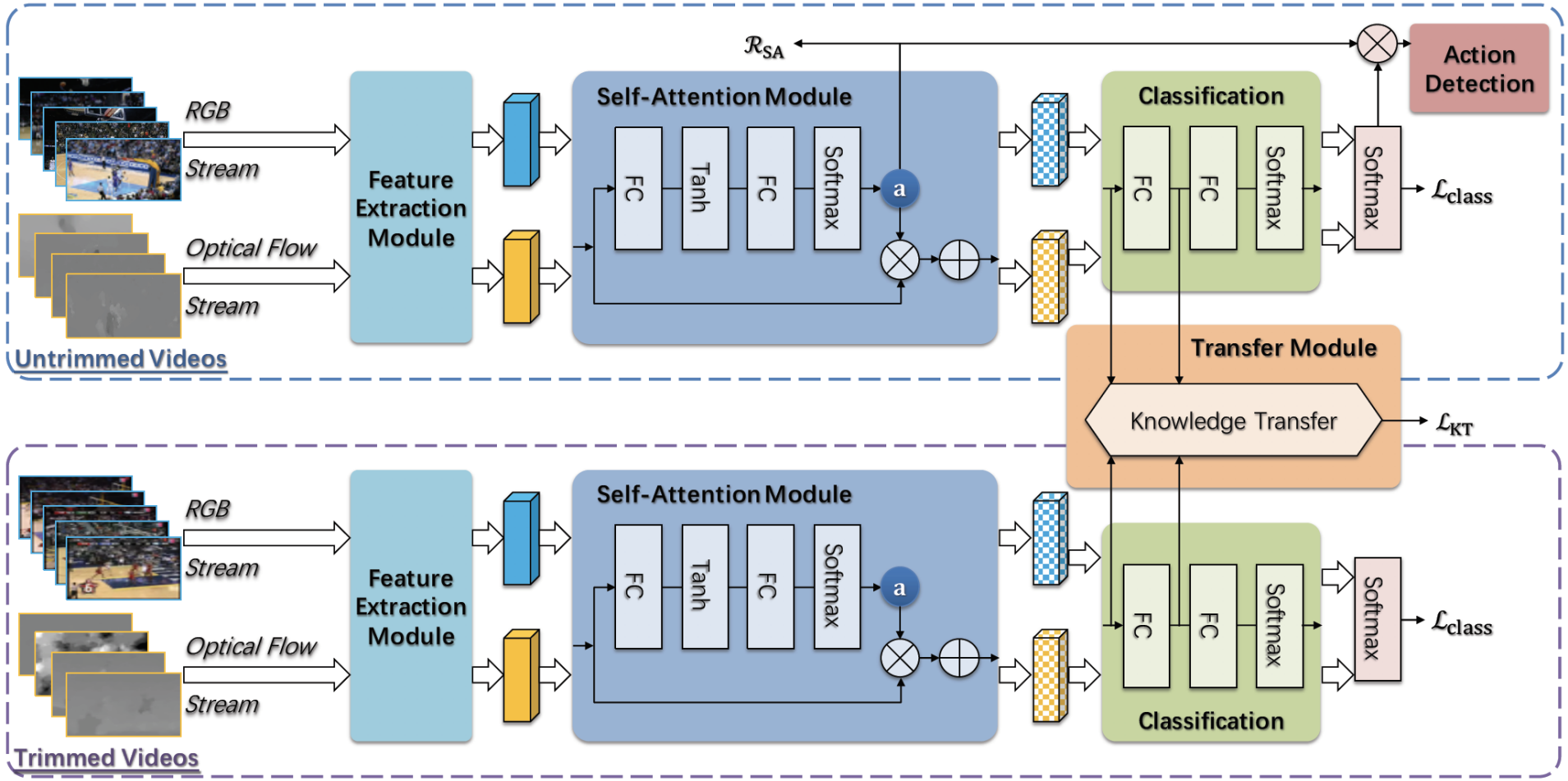


Zhouhan Lin et al. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING, in ICLR 2017

Outline

- 1. Introduction
- 2. Our Method (TSRNet)
- 3. Evaluation
- 4. Conclusion

Overview



Two-stream CNNS: RGB and Optical Flow

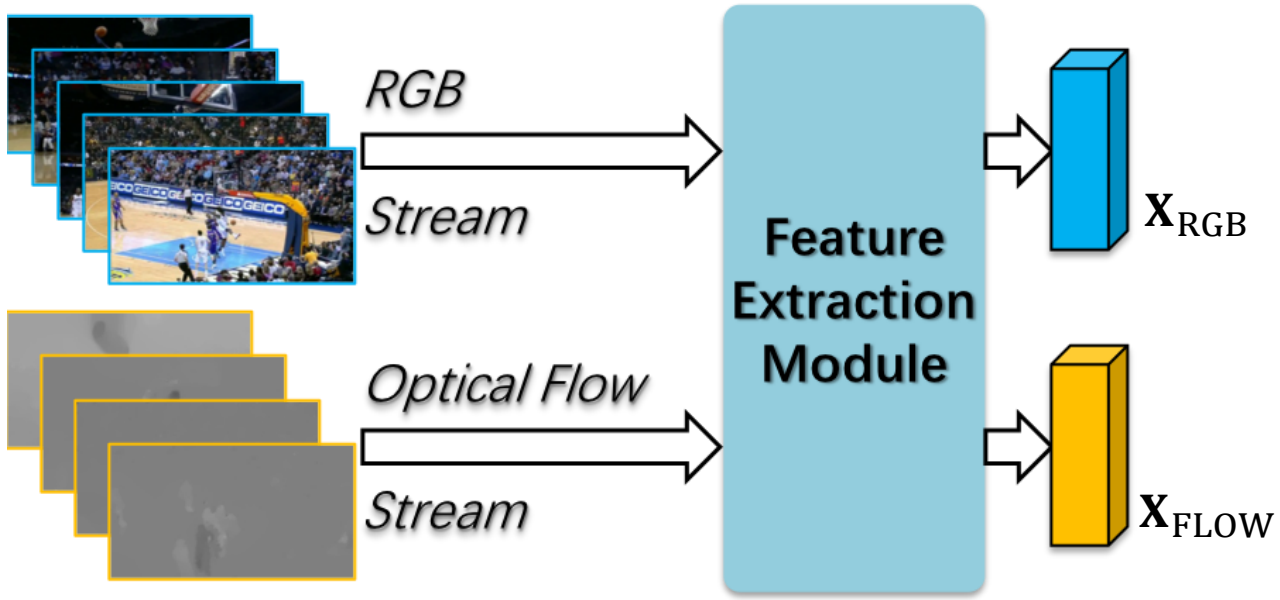
a: the attention vector

Contributions

- TSRNet is the first to introduce transfer learning for action recognition in untrimmed videos with weak supervision.
- TSRNet is able to obtain the self-attention weights at frame levels which can be used to localize frames for action detection.
- TSRNet achieves superior performance on two challenging untrimmed video datasets.

Feature Extraction

Two-Stream Feature Extraction



Base network: ResNet101

$\mathbf{X}_{\mathbf{RGB}}$: RGB feature matrices

$\mathbf{X}_{\mathbf{FLOW}}$: Optical flow feature matrices

Learning

Self-attentive Action Classification

$$\mathbf{m} = \mathbf{Xa} = \mathbf{X}(\mathrm{softmax}(\mathbf{w_2} \cdot \tanh(\mathbf{W_1 X})))^{\mathrm{T}}$$

$$\mathcal{L}_{SA} = \mathcal{L}_{class} + \mathcal{R}_{SA}$$

$$\mathcal{R}_{SA} = \alpha \mathcal{R}_{smooth} + \beta \mathcal{R}_{sparsity}$$

$$\mathcal{R}_{smooth} = \sum_{i=1}^{n-1} (a_i - a_{i+1})^2 , \mathcal{R}_{sparsity} = ||a||_1$$

$\mathcal{L}_{class}$: the standard multi-label cross-entropy loss

a: attention weights vector

$\mathbf{X}$: feature matrix, $\mathbf{m}$: a weighted sum of feature vectors

Learning

Knowledge Transfer between Trimmed and Untrimmed Videos

$$\mathcal{L}_{KT} = \mathcal{L}_{FC1} + \mathcal{L}_{FC2}$$

$$\mathcal{L}_{FC1} = MMD^2(\mathcal{T}, \mathcal{U})$$

$$= \frac{1}{n_T^2}\sum_{i=1}^{n_T}\sum_{j=1}^{n_T} k(t_i, t_j) + \frac{1}{n_U^2}\sum_{i=1}^{n_U}\sum_{j=1}^{n_U} k(u_i, u_j) - \frac{2}{n_T \cdot n_U}\sum_{i=1}^{n_T}\sum_{j=1}^{n_U} k(t_i, u_j)$$

$$\mathcal{L}_{FC2} = MMD^2(FC1(\mathcal{T}), FC1(\mathcal{U}))$$

$\mathcal{T} = \{\boldsymbol{t}_i|_{i=1}^{n_T}\}$, $\mathcal{U} = \{\boldsymbol{u}_i|_{i=1}^{n_U}\}$, represent the sets of trimmed and untrimmed videos features.

$k(\cdot,\cdot)$: the predefined Gaussian kernel function.

Total Loss: $\mathcal{L} = \mathcal{L}_{SA} + \mathcal{L}_{KT}$

Localization

$$w_i^c = a_i s_c$$
$$\overline{w}_i^c = \theta \cdot w_{i,RGB}^c + (1-\theta) \cdot w_{i,Flow}^c$$
$$t_{start} = \frac{ind_{start}}{F}, t_{end} = \frac{ind_{end}}{F}$$

w_i^c: the weighted score of each frame i for class c.

s_c: $s_c = [s_1, s_2, \dots, s_m]^T \in \mathbb{R}^{m \times 1}$ is the output of softmax layer.

$[ind_{start}, ind_{end}]$: the frames indices of starting and ending positions.

F: the fps(frames per second) of videos.

Outline

- 1. Introduction
- 2. Our Method (TSRNet)
- 3. Evaluation
- 4. Conclusion

Settings

Evaluation is based on training on the paired datasets.

- Data for training:

  Source domain training: Trimmed videos from the source domain.

  Domain adaptation training: Untrimmed videos from the target domain.

- Test:

  Test set from the target domain

- Transfer scenarios:

  (a). UCF101 to THUMOS14

  (b). UCF101 to ActivityNet1.3

Accuracy

Action Classification Results on THUMOS14

Table 1: Classification accuracy (%) of all the methods on the THUMOS14 dataset for action recognition. Note that SRNet is a simpler version of TSRNet, which excludes the knowledge transfer module.

	RGB	Optical Flow	Fusion
(Wang and Schmid 2013)	-	-	63.1
(Wang et al. 2016)(3 seg)	-	-	78.5
(Wang et al. 2017)	-	-	82.2
Two-Stream	68.2	71.6	73
SRNet	72.3	76.2	79.4
TSRNet	**74.4**	**79.6**	**87.1**

Accuracy

Action Classification Results on ActivityNet1.3

Table 2: Classification accuracy (%) of all the methods on the ActivityNet1.3 dataset for action recognition. Note that SRNet is a simpler version of TSRNet, which excludes the knowledge transfer module.

	RGB	Optical Flow	Fusion
Two-Stream	71.4	73.5	79.2
SRNet	74.3	80.1	86.9
TSRNet	**79.7**	**84.3**	**91.2**

Accuracy

Action Detection Results on THUMOS14

Table 3: Comparisons on the THUMOS14 dataset for action detection.

	Method	mAP@IoU (%)								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Full supervision	(Richard and Gall 2016)	39.7	35.7	30.0	23.2	15.2	-	-	-	-
	(Shou, Wang, and Chang 2016)	47.7	43.5	36.3	28.7	19.0	10.3	5.3	-	-
	(Yeung et al. 2016)	48.9	44.0	36.0	26.4	17.1	-	-	-	-
	(Alwassel, Heilbron, and Ghanem 2017)	49.6	44.3	38.1	28.4	19.8	-	-	-	-
	(Lin, Zhao, and Shou 2017)	50.1	47.8	43.0	35.0	24.6	-	-	-	-
	(Yuan et al. 2016)	51.4	42.6	33.6	26.1	18.8	-	-	-	-
	(Shou et al. 2017)	-	-	40.1	29.4	23.3	13.1	7.9	-	-
	(Xu, Das, and Saenko 2017)	54.5	51.5	44.8	35.6	28.9	-	-	-	-
	(Zhao et al. 2017)	**66.0**	**59.4**	**51.9**	**41.0**	**29.8**	-	-	-	-
Weak supervision	(Wang et al. 2017)	44.4	37.7	28.2	21.1	13.7	-	-	-	-
	(Singh and Lee 2017)	36.4	27.8	19.5	12.7	6.8	-	-	-	-
	(Nguyen et al. 2017)	52.0	44.7	35.5	25.8	16.9	9.9	4.3	1.2	0.1
	(Nguyen et al. 2017)	45.3	38.8	31.1	23.5	16.2	9.8	5.1	2.0	0.3
	TSRNet (w/o $\mathcal{L}_{FC2}$)	53.5	45.3	35.9	26.5	17.2	10.4	5.31	1.93	0.21
	TSRNet	**55.9**	**46.9**	**38.3**	**28.1**	**18.6**	**11.0**	**5.59**	**2.19**	**0.29**

Accuracy

Action Detection Results on ActivityNet1.3

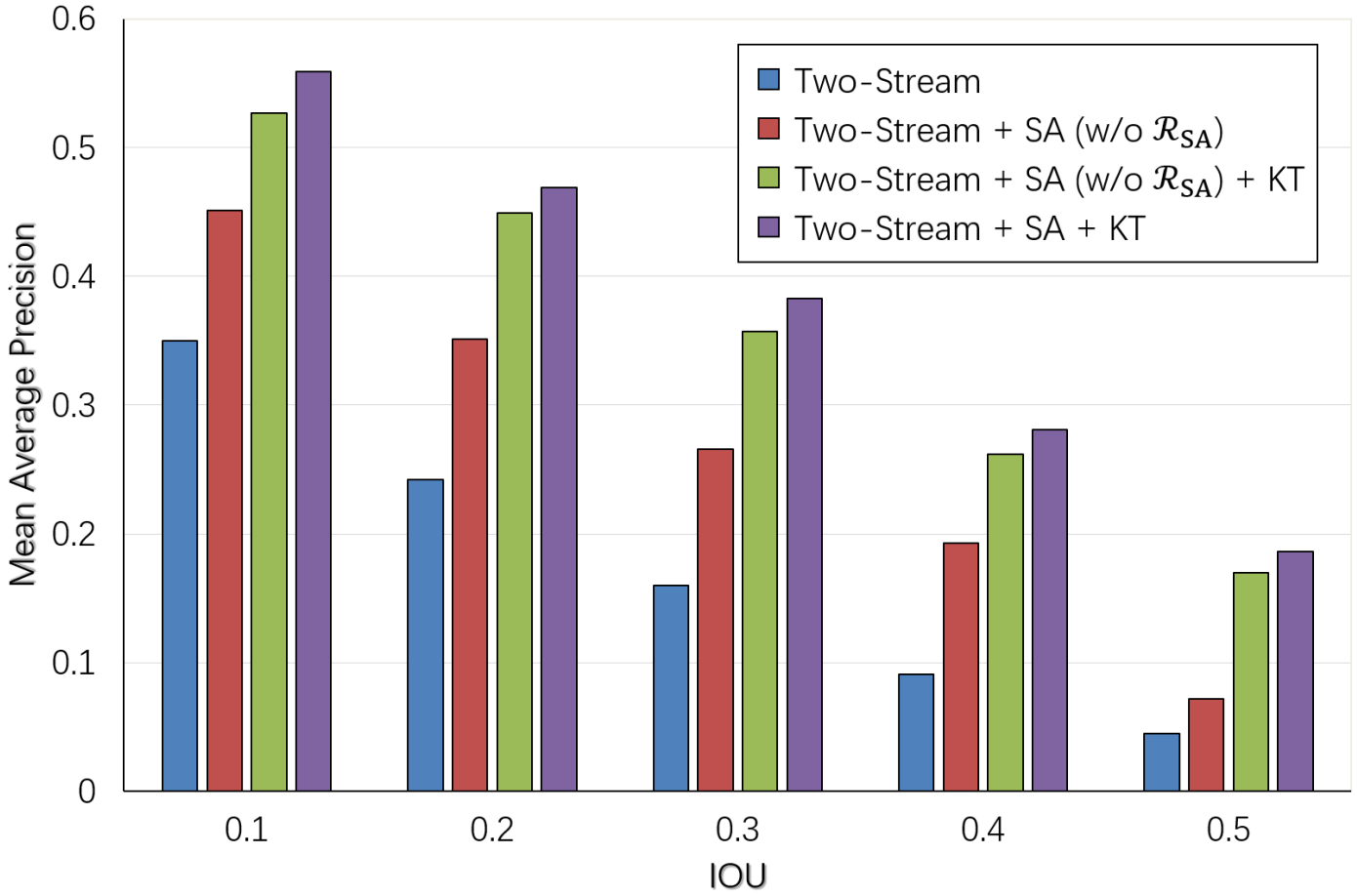
Table 4: Comparisons on the ActivityNet1.3 dataset for action detection.

	Methods	mAP@IoU (%)			
		0.5	0.75	0.95	Average
Full supervision	(Singh and Cuzzolin 2016)	34.5	-	-	11.3
	(Xu, Das, and Saenko 2017)	26.8	-	-	-
	(Xiong et al. 2017)	29.1	23.5	5.5	-
	(Heilbron et al. 2017)	40.0	17.9	4.7	21.7
	(Shou et al. 2017)	45.3	26.0	0.2	23.8
	(Zhao et al. 2017)	39.12	23.48	5.49	23.98
	(Lin et al. 2018)	**52.50**	**33.53**	**8.85**	**33.72**
Weak supervision	(Nguyen et al. 2017)	29.3	16.9	2.6	-
	TSRNet (pretrained:[ResNet101@ImageNet])	29.9	17.2	2.71	19.56
	TSRNet (pretrained:[TSRNet@overlap30])	**33.1**	**18.7**	**3.32**	**21.78**

pretrained:[TSRNet@overlap30]: we use the classes with overlapping labels found between the UCF101 and ActivityNet1.3 datasets to initialize the TSRNet and train it using the whole classes.
pretrained:[ResNet101@ImageNet]: we use the ResNet101 pretrained on ImageNet dataset to initialize the TSRNet and train it using the whole classes.

Analysis

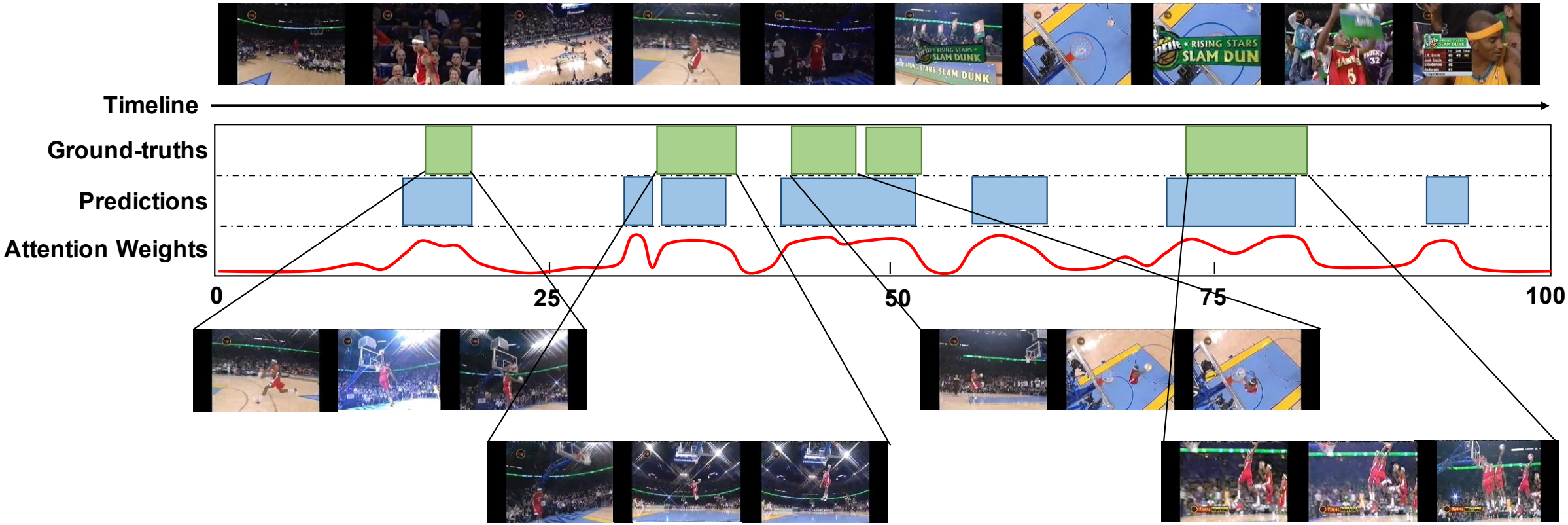
Ablation study of TSRNet



SA: Self-attention module
KT: Knowledge transfer module

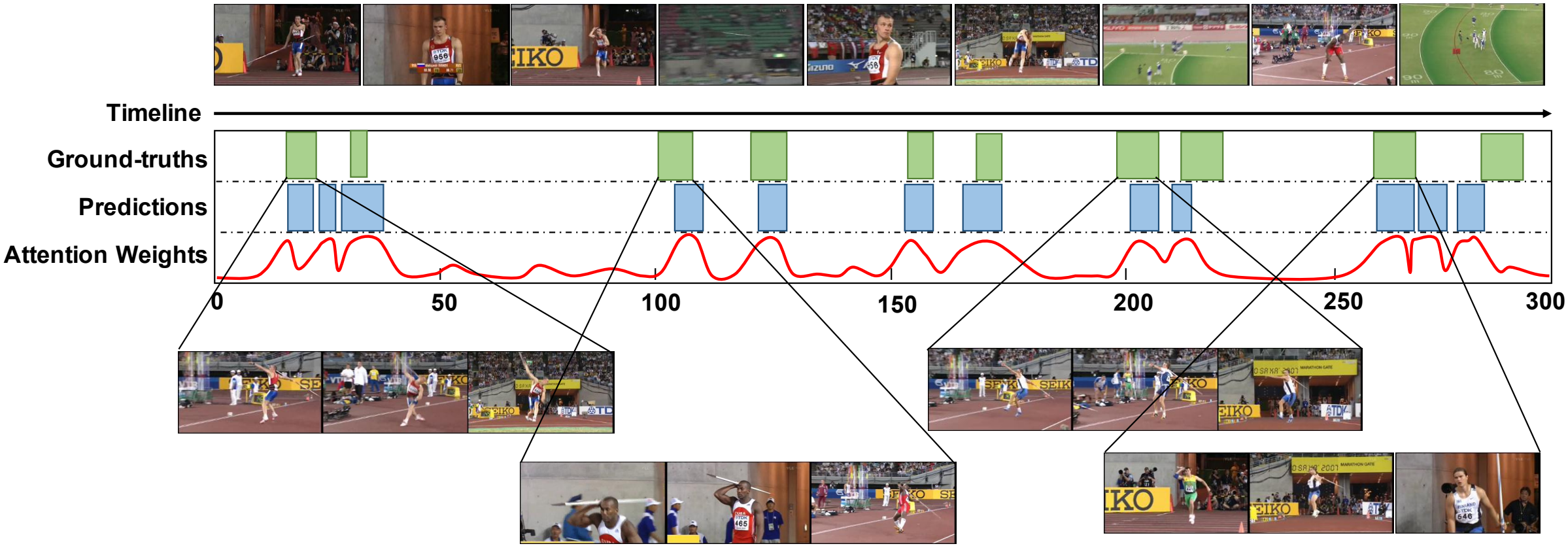
Analysis

Ablation study of TSRNet



Analysis

Ablation study of TSRNet



Outline

- 1. Introduction
- 2. Our Method (TSRNet)
- 3. Evaluation
- 4. Conclusion

Conclusion

- TSRNet transfers knowledge extract from publicly available trimmed videos for action recognition and detection in untrimmed videos.
- TSRNet is able to learn transferable self-attentive representations which preserves strong discriminability in action recognition by introducing the self-attention mechanism.
- TSRNet can automatically weight each videos frame which can be used to localize frames for action detection.
- TSRNet achieves superior performance on two challenging untrimmed video datasets.

Thank you!

Questions & Answers!