Lecture 7

Human Action Detection

Rui Dai

rui.dai@inria.fr

About Me

Rui Dai

- Home page: https://dairui01.github.io/
- Ph.D. candidate at INRIA, STARS team.
- Research topic: “Action detection using Deep Learning methods”.

Outline

- Introduction
  - Definition? Application?
- Datasets
  - THUMOS, ActivityNet, EPIC-Kitchen, Charades
- Evaluation Metrics
  - Event-level
  - Frame-level
- Methods
  - Sliding window
  - Anchor-based
  - Actioness-based
  - Seq-to-Seq

Section 1

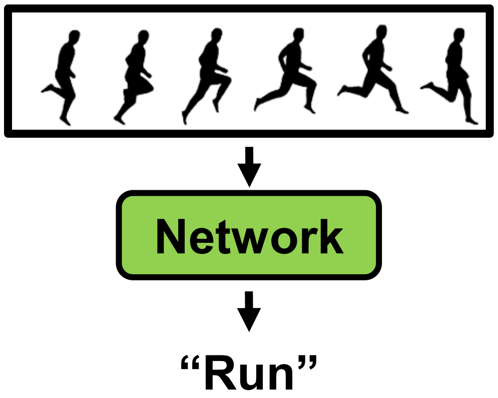
Introduction

Action Recognition

Recap…

Input: A clipped video (a sequence of frames)

Output: An action label



Videos are untrimmed in real-world

Example 1

Challenges :

1. Composite Activities
   e.g. Cook
3. Low Camera Framing
   e.g. Dump in Trash

Person 02

Camera 03

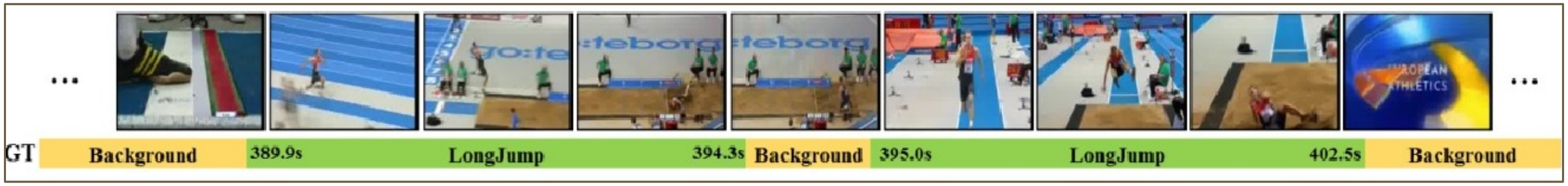
Frame 2379

Annotated Activities By Category

Single	**Composite & Elementary**	**Object-based**
Take_sth._off_table Walk	Cook	

Action Detection/Localization

- Given a long untrimmed video that contains many activities, we detect the start and the end of each activity and the activities labels.

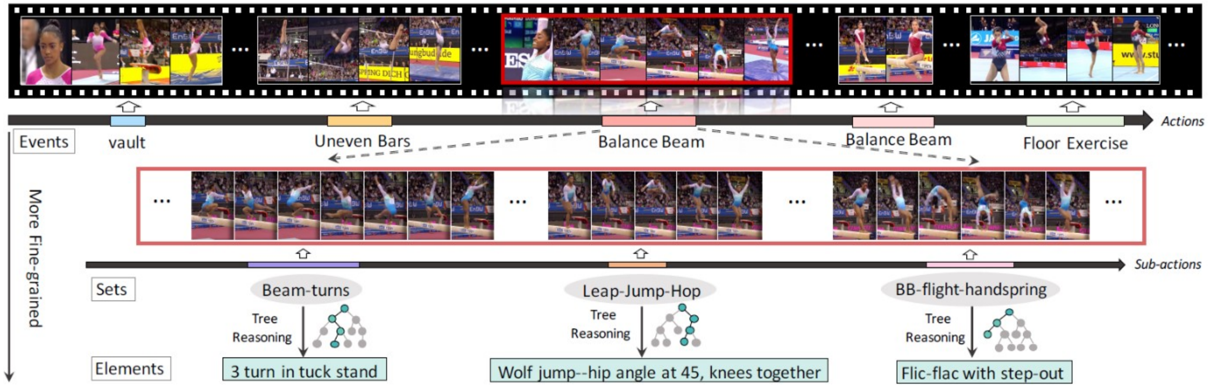


- [Same task with Different names] Action detection, or Action localization, or Temporal action localization… (CVPR2017 ActivityNet challenges)
- [Close task] Action Segmentation, different in Evaluation Metrics

Applications

- Public video surveillance (Smart Mart)

- Skill assessment (Tennis/Basketball)



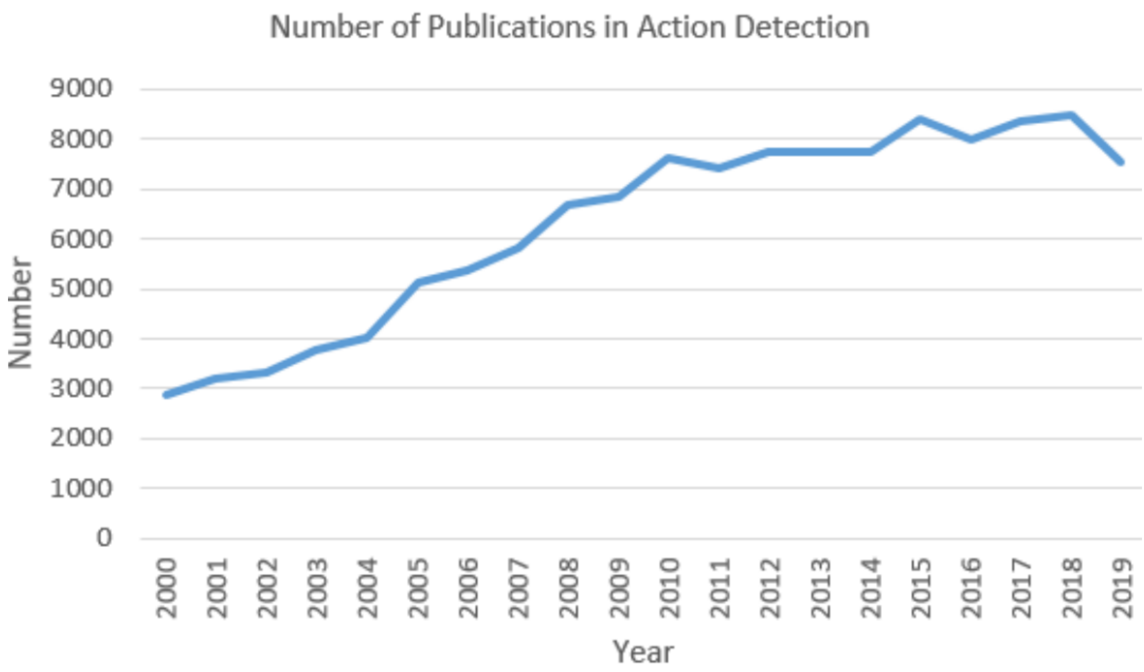
- Daily life security
- Video summarization (YouTube)

Challenges

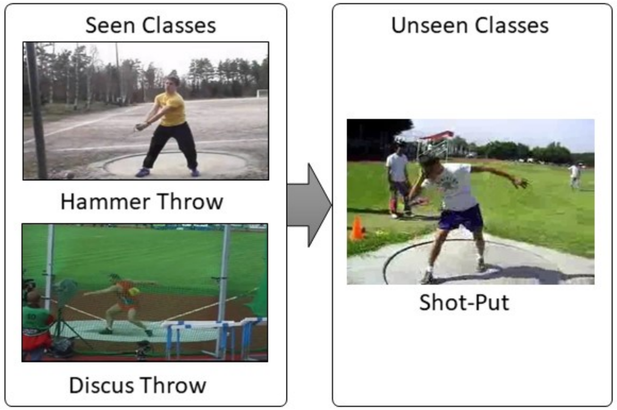
- Unclear boundary
- Large temporal spans
- Open environment
  - Multi-scale
  - Multi-target
  - Camera movement

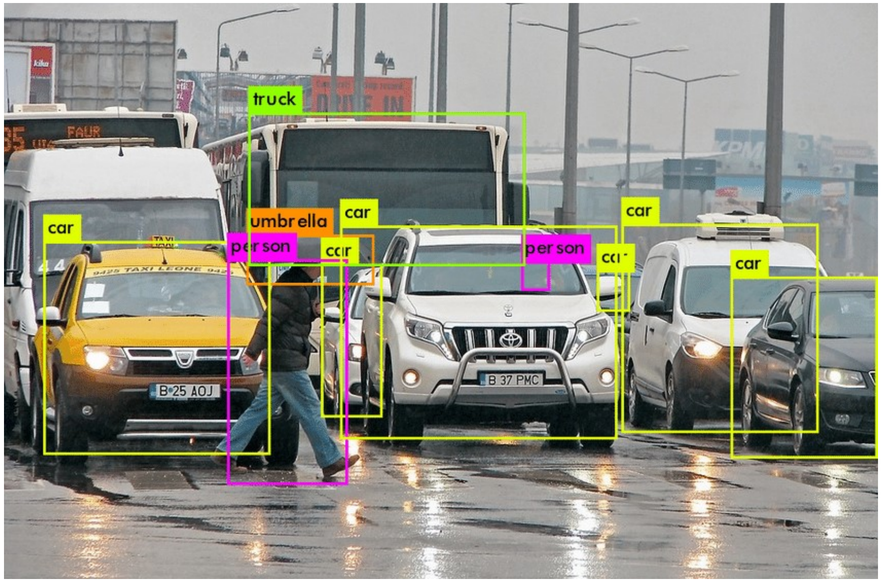
There is still no robust solution for this task currently

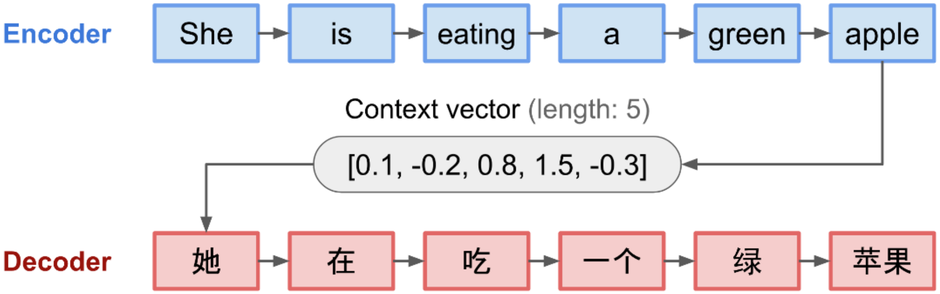
Popular research domain



Similar tasks





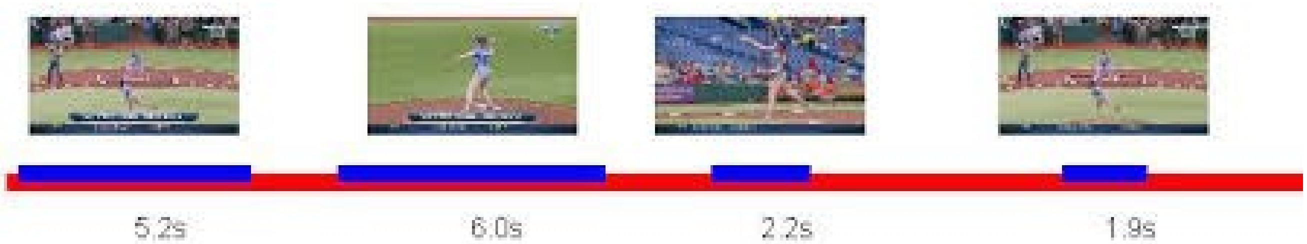


Section 2

Datasets

THUMOS14

- Source: Web/YouTube
- Type: Sport
- Average duration: 2-3 mins
- Action classes: 20



ActivityNet

- Source: Web/YouTube
- Type: Mixed (Daily Living, Sport…)
- Average Duration: 2-3 mins
- Action classes: 200
- In total 648 hrs. of video

EPIC-Kitchen

- Source: Self-recorded
- Type: Cooking
- Env: 45 kitchens
- 100 hrs. of recording
- 97 verb+300 noun
- 90K action segments
- Object-relevant actions

Charades

- Source: Self-recorded
- Type: Actions of Daily Living (ADL)
- Env: Home
- 157 action classes
- Avg duration: 30s
- 9800+ videos
- **Densely annotated**

Section 3

Evaluation Metrics

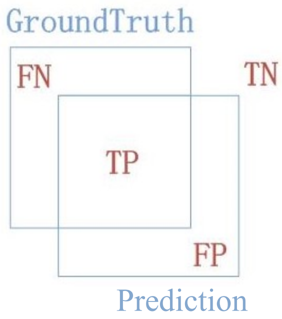
Section 3.1

Event-level

Basic Concepts

TP, TN, FP, FN

Recap…



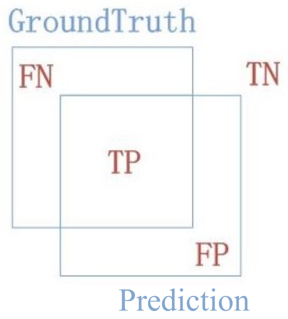
Ground Truth	**Detector Output**	
	Present	**Absent**
Present	True Positive (TP)	False Negative (FN)
Absent	False Positive (FP)	True Negative (TN)

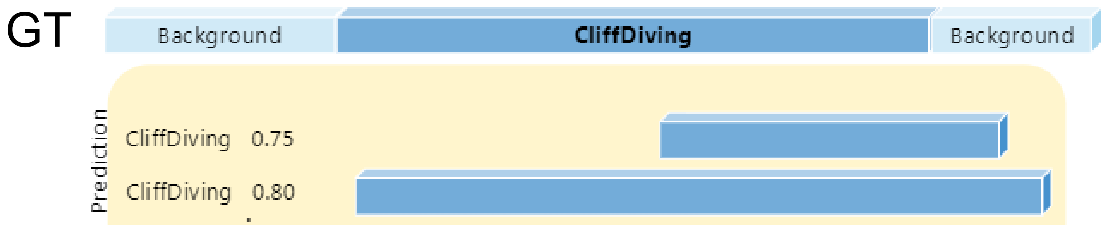
Detection result interpretation based on an object's presence in **Ground Truth** (GT) and **Detector output**.

Basic Concepts

TP, TN, FP, FN

Recap…

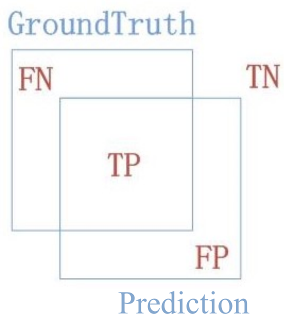




Basic Concepts

- **Recall** is the coverage of predicting correctly. Specifically, recall is that how many real positive samples in the testing set were identified. The formula is as follows.

$$recall = \frac{TP}{TP + FN}$$

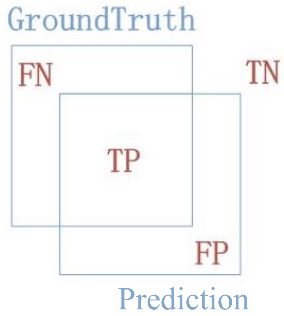


Basic Concepts

- Specifically, **precision** is the percentage of the predicted real positive samples in predicted results. The formula is as follows

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{n}$$

- In which, n is the sum of True Positive and False Positive, and n is also the total number of samples identified by the system.



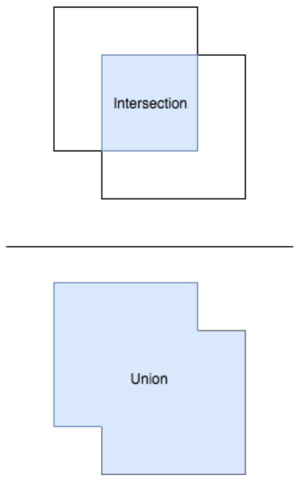
Basic Concepts

Intersection-over-Union (IoU)

- IoU can be understand as the overlap between the predicted detection box by the model and the ground truth for the object detection in images. In fact, it is the accuracy of detection. The calculation formula is the intersection of Detection Result and Ground Truth compared to their union

$$IoU = \frac{predicted\ detection\ box \cap ground\ truth}{predicted\ detection\ box \cup ground\ truth}$$

=



Basic Concepts

- IoU is used to check whether the IoU between the predicted result and the ground truth is greater than a predicted threshold.
- We often set 0.5 as the threshold. If the IoU is greater than 0.5, the object will be identified as ‘‘detected successfully’’, otherwise it will be identified as ‘‘missed’’.
- In temporal action detection, IoU is changed into t-IoU for time which has only one dimension.

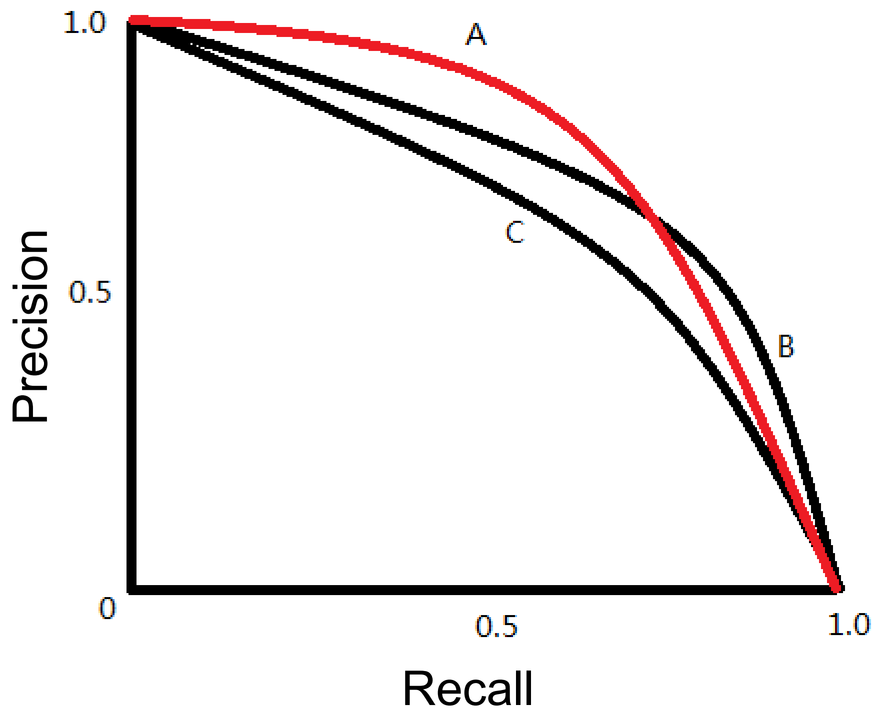
Evaluation Metrics

Precision-Recall Curve [Lecture 3]

Every class has a curve

AP: Surface under the curve

mAP: mean of all the APs



Summary

Under a certain t-IoU,

- **AP** is the average accuracy of the predicted proposals of class C in a video.
- **MAP** is the mean of the average accuracy of the predicted proposals of all classes in all testing videos.

Following the standard evaluation protocol, almost all papers report mAP at different thresholds of t-IoU.

Section 3.1

Frame-level

Frame-wise Accuracy

- Represents the ratio of correctly classified frames to all frames in the dataset.
- **N_c** is the number of frames labelled **c** in the ground truth.

$$\mathcal{FA}_1 = \frac{\sum_{c \in \mathcal{C}} TP^c}{\sum_{c \in \mathcal{C}} N_c}$$

F-Score

- This metric combines Precision **P** and Recall **R** is defined as the harmonic mean of these two values.

- **P^c:** Precision for class **c**
- **R^c:** Recall for class **c**
- **C:** is the number classes in the dataset

$$\mathcal{P}^c = \frac{TP^c}{TP^c + FP^c} \qquad \mathcal{R}^c = \frac{TP^c}{TP^c + FN^c}$$

$$F\text{-}Score = \frac{2}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \times \frac{\mathcal{P}^c \times \mathcal{R}^c}{\mathcal{P}^c + \mathcal{R}^c}$$

Section 4

Methods for Human Action Detection

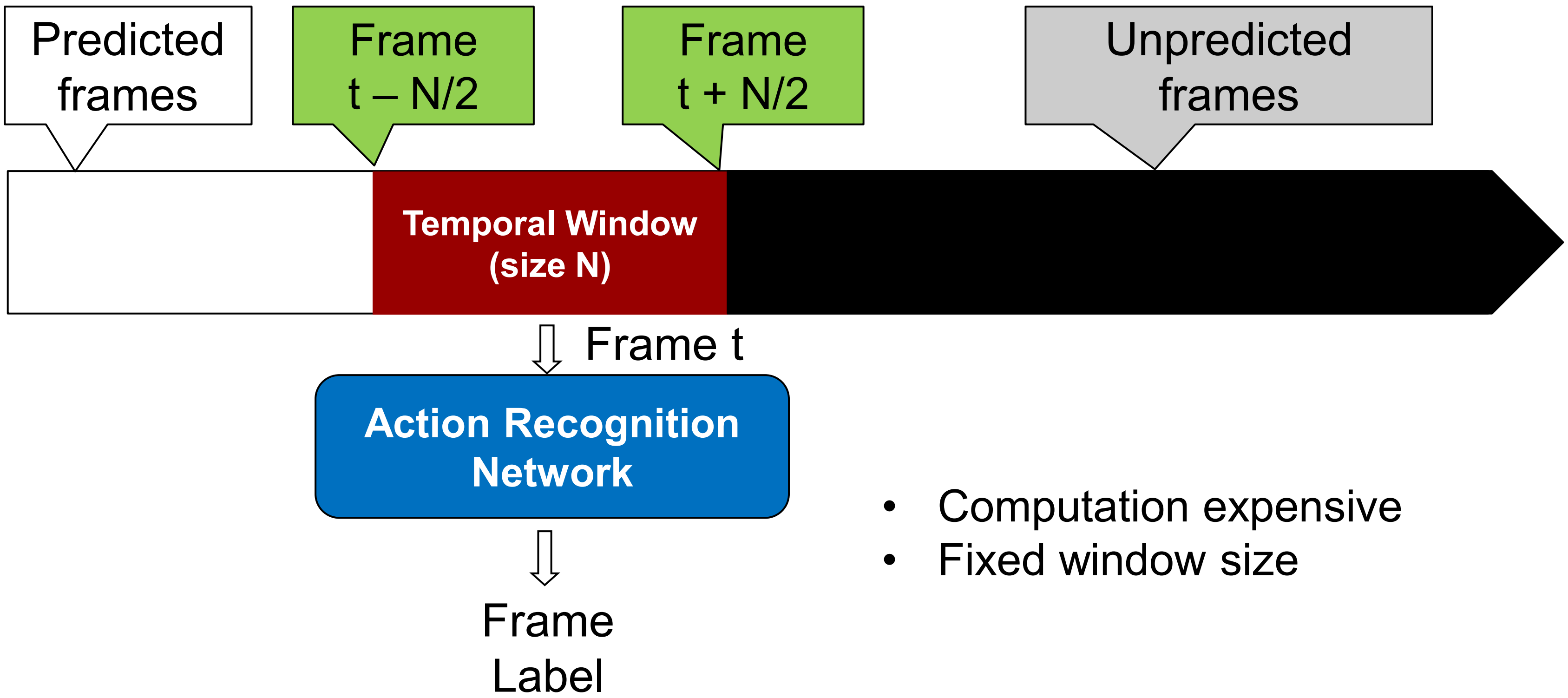
Section 4.1

Sliding Window

Siding window approach

Frame-level Action detection

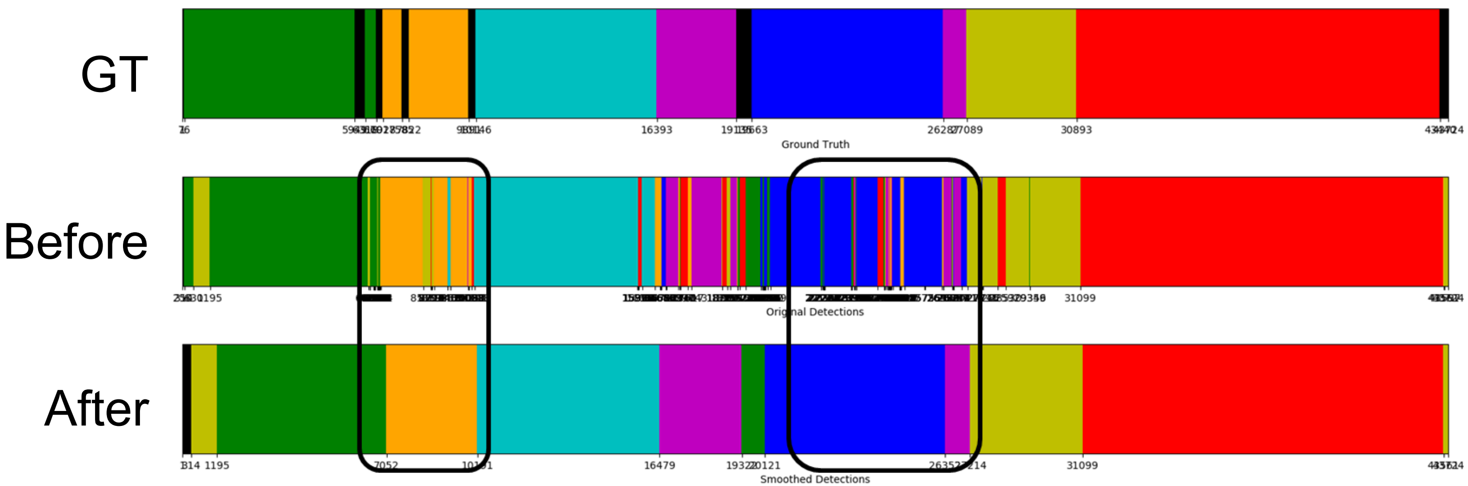
Extend directly from Action Recognition.



- Computation expensive
- Fixed window size

Post-processing

- Refine the prediction
  (Remove noisy false detection)



Post-processing

- Filter out false detections based on the average duration of the activities calculated from training split

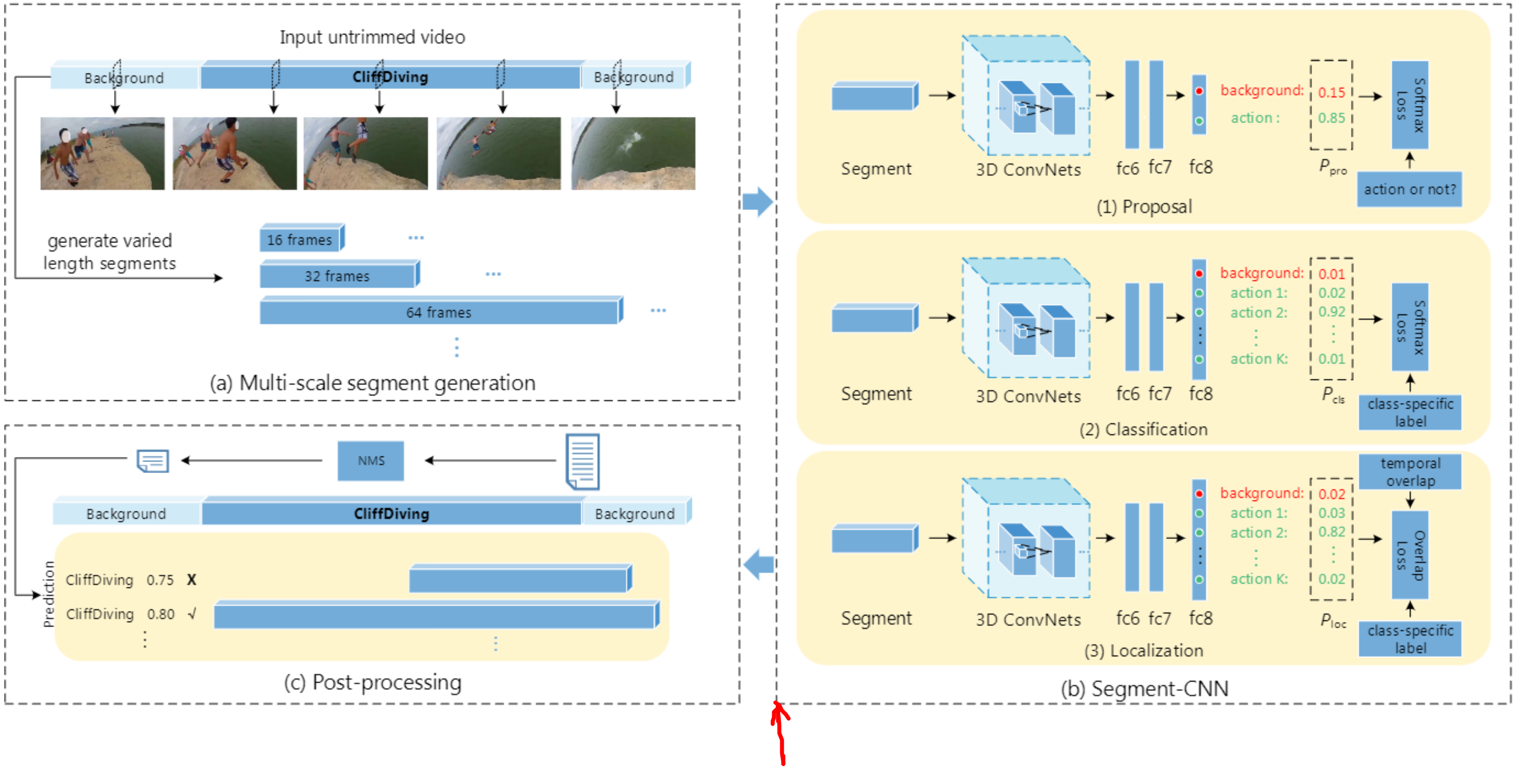
Algorithm 1 The proposed post-processing algorithm depending on activities duration

```
Result: Post processed activity intervals
input_dataframe = start - end frames and name of activities
avg_length = Lookup containing average lengths of activities
n_start = start frame of the fine-tuned activities (init = 0)
intervals_to_delete = index of intervals identified as noise
threshold = Hyperparameter (0.1 by validation)
counter = -1
for action, start_frame, end_frame in input_dataframe do
    counter ← counter+1
    activity_length ← end_frame - start_frame + 1
    avg_length_action ← avg_length[activity]
    greedy_criterion ← (activity_length/avg_length_activity)
    if greedy_criterion < threshold then
        if n_start = 0 then n_start ← start_frame ;
        else continue;
        intervals_to_delete.add(counter)
    else
        if n_start ≠ 0 then
          input_dataframe['start_frame'][counter] ← n_start;
        else continue;
    end
end
final_dataframe = input_dataframe.remove(intervals_to_delete)
```

Dependent to datasets!

SCNN [CVPR'16]

a) Multi-scale segment generation
b) Segment-CNN
c) Post-processing



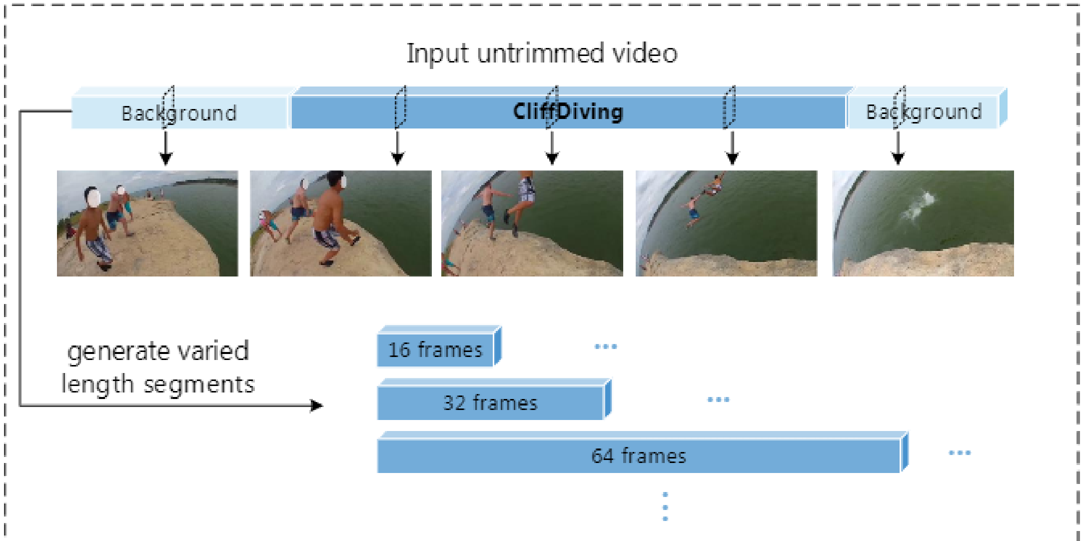
SCNN [CVPR'16]

a) Multi-scale segment generation

Win Size= [16,32,64,128,256,512]

Overlapping = 75%

Segment length = 16 frames (Sampling from Win)

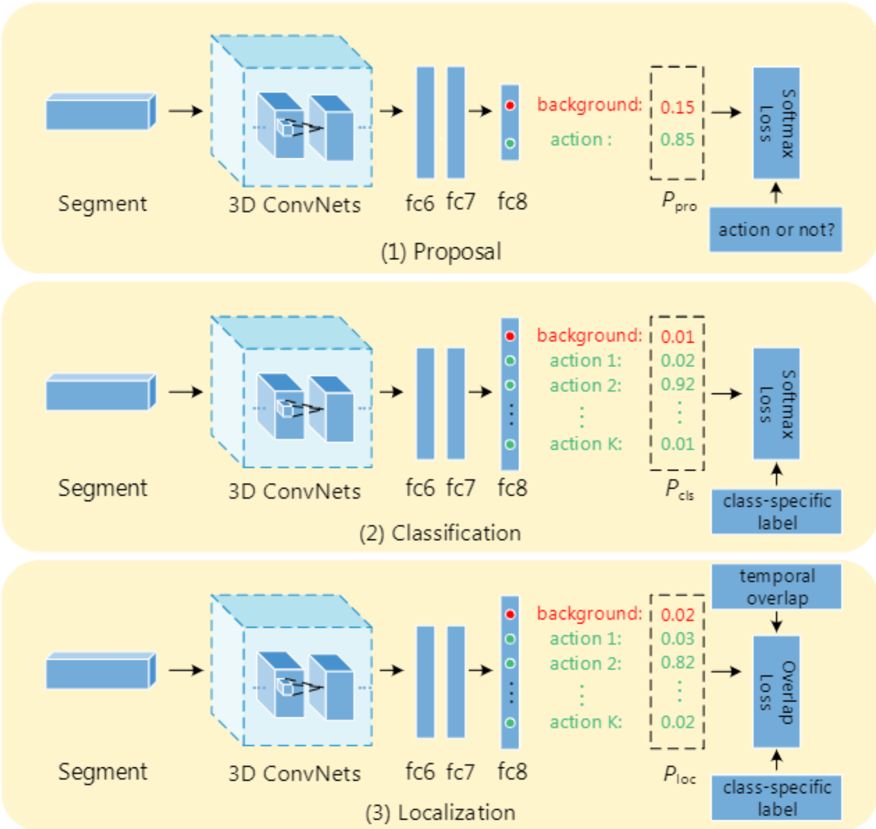


(a) Multi-scale segment generation

SCNN [CVPR'16]

b) Segment-CNN

C3D as 3D ConvNets

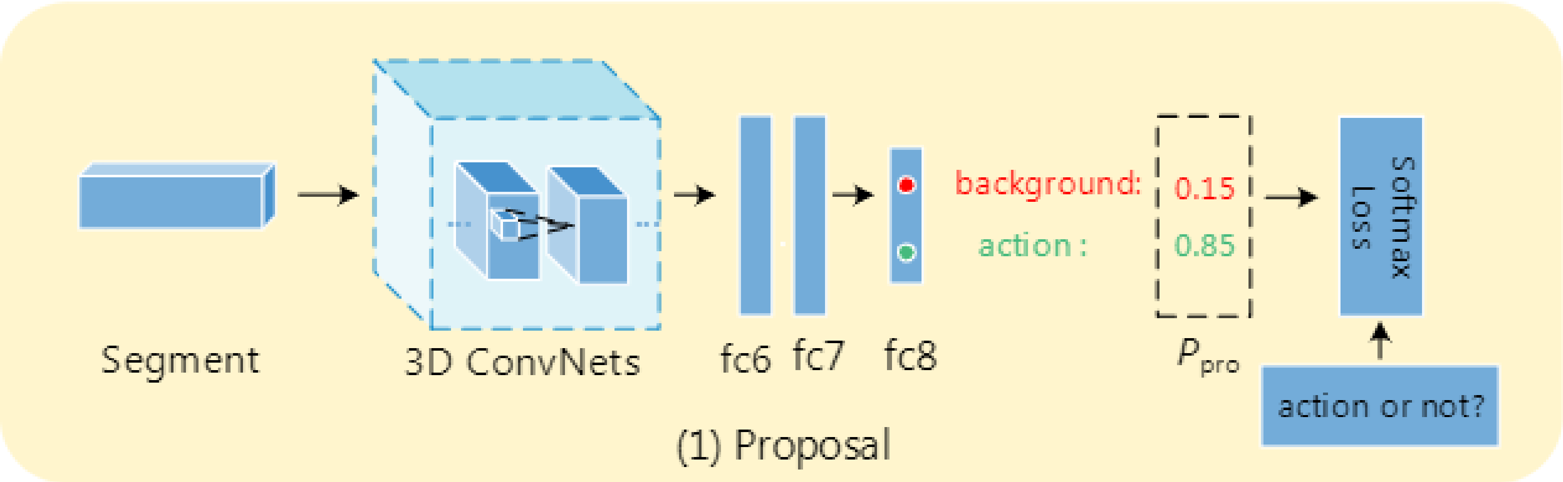


SCNN [CVPR'16]

b) Segment-CNN-proposal

Based on the segments, filter them

- IoU>0.7 Action
- IoU<0.3 Background
- Other Remove
- Sampling: Foreground=Background

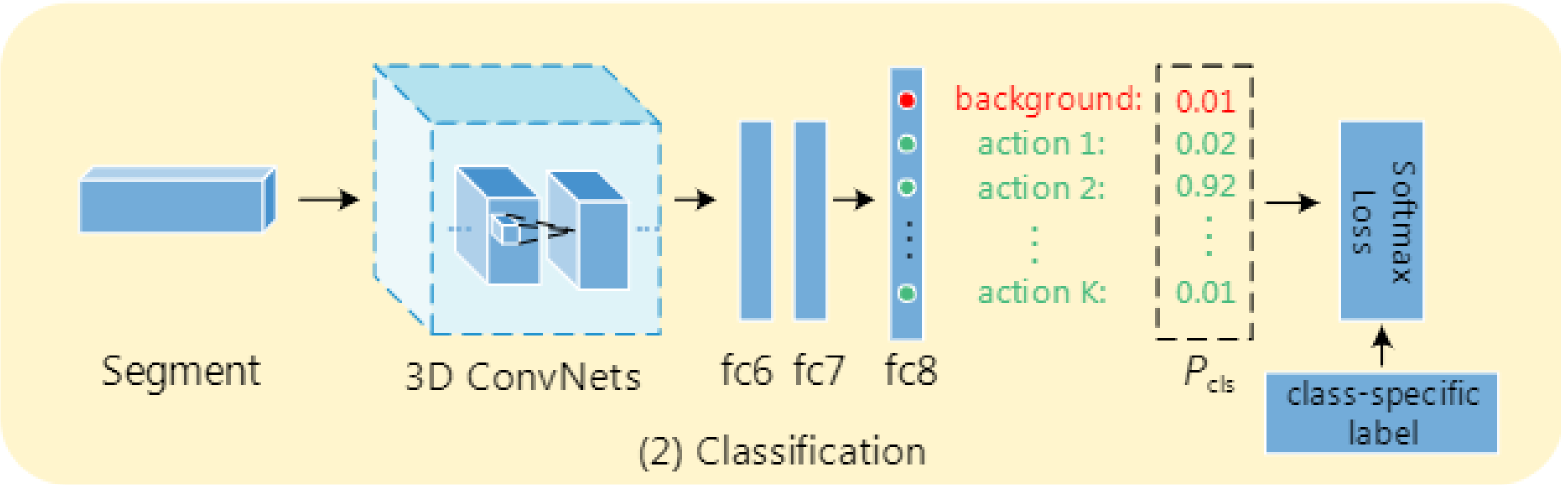


SCNN [CVPR'16]

b) Segment-CNN-classification

After the Proposal, classify the segments

Background + action classes



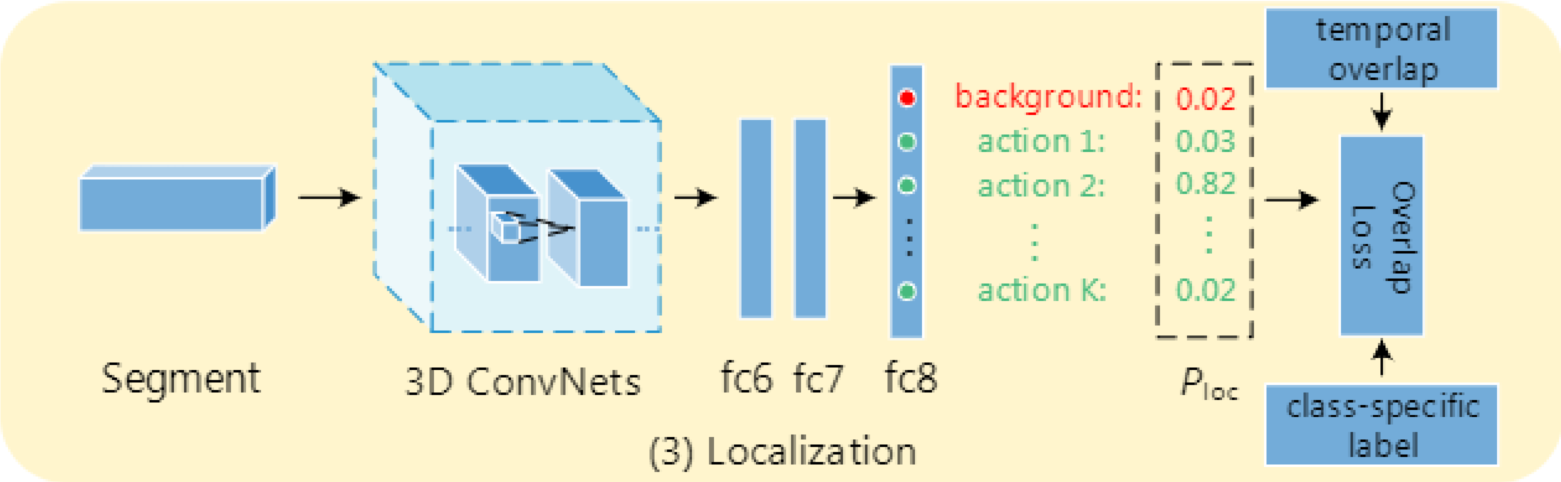
SCNN [CVPR'16]

b) Segment-CNN-localization

Initialization by classification Net (same weights)

$$\mathcal{L}_{\text{overlap}} = \frac{1}{N} \sum_n \left(\frac{1}{2} \cdot \left(\frac{\left(P_n^{(k_n)}\right)^2}{(v_n)^\alpha} - 1 \right) \cdot [k_n > 0] \right)$$

$$\mathcal{L} = \mathcal{L}_{\text{softmax}} + \lambda \cdot \mathcal{L}_{\text{overlap}}$$



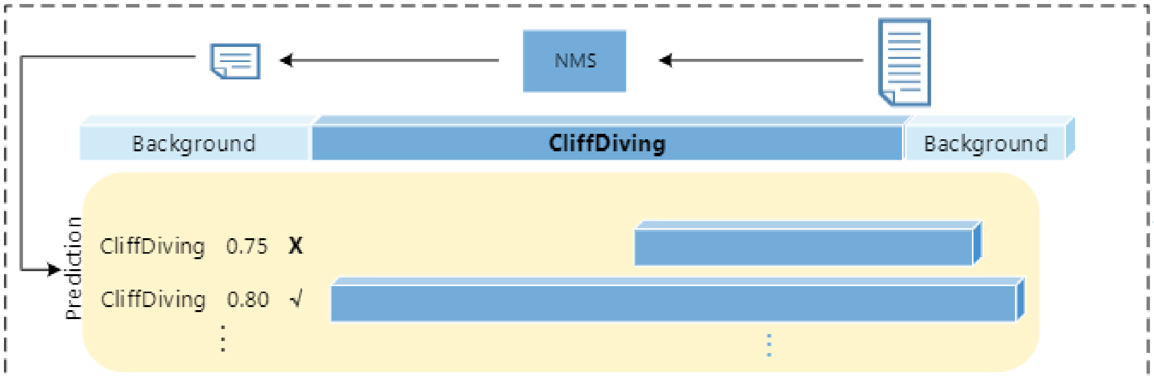
Inference time

Use only Proposal + Localization Network

Proposal: Action score >0.7 ?

- No: Background
- Yes: Localization Net: Action label

c) Postprocessing: NMS ($\theta = 0.1$)



(c) Post-processing

Non-Maximum-Suppression (NMS)

Input : $\mathcal{B} = \{b_1, .., b_N\}$, $\mathcal{S} = \{s_1, .., s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow \text{argmax } \mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$
      end
      NMS

      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$
      Soft-NMS
    end
  end
  return $\mathcal{D}, \mathcal{S}$
end

Non-Maximum-Suppression (NMS)

- **Input:** A list of Proposal boxes B, corresponding confidence scores S and overlap threshold N.
- **Output:** A list of filtered proposals D.

Algorithm

1. Select the proposal with highest confidence score, remove it from B and add it to the final proposal list D. (Initially D is empty).
2. Now compare this proposal with all the proposals — calculate the IOU of this proposal with every other proposal. If the IOU is greater than the threshold N, remove that proposal from B.
3. Again take the proposal with the highest confidence from the remaining proposals in B and remove it from B and add it to D.
4. Once again calculate the IOU of this proposal with all the proposals in B and eliminate the boxes which have high IOU than threshold.
5. This process is repeated until there are no more proposals left in B.

SCNN [CVPR'16]

Drawbacks:

- Computation expensive $\Leftrightarrow$ Precision
  (over-lapping, redundancy…)

- Large complexity
  - Generating different segments
  - Multiple 3D ConvNets

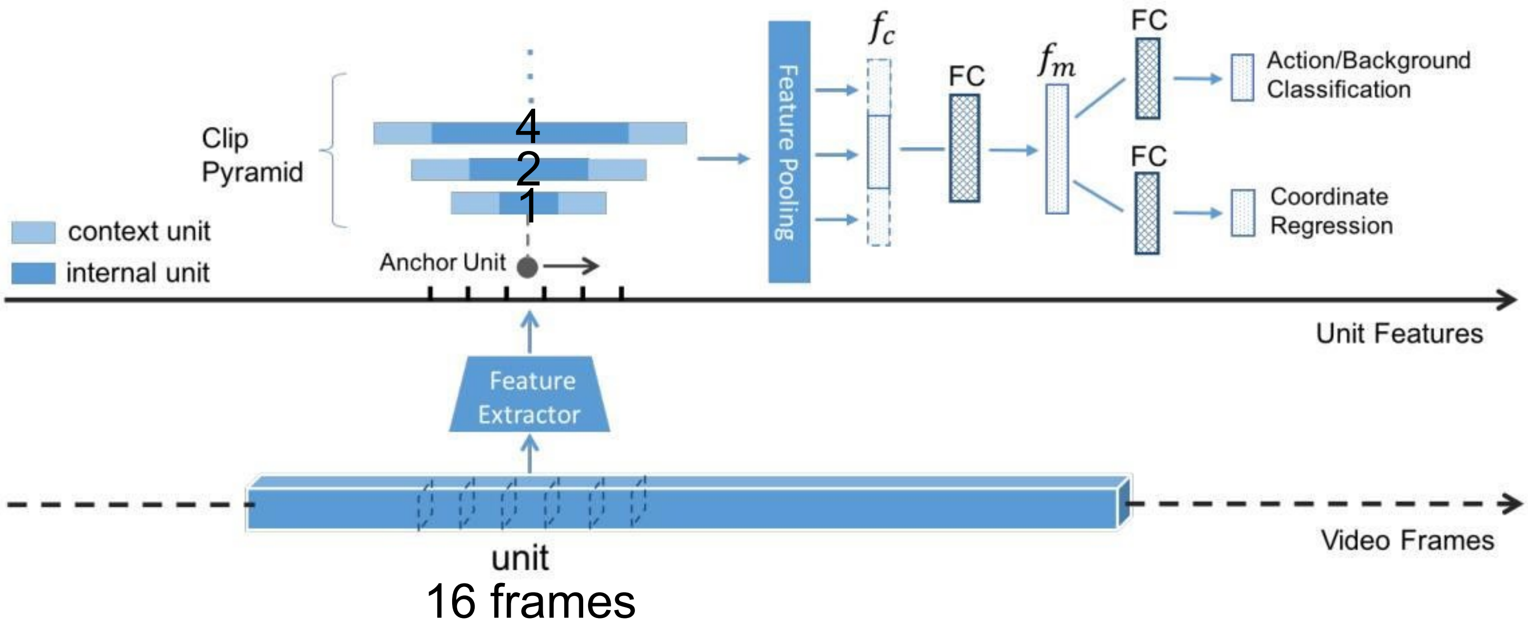
Section 4.2

Anchor based

Temporal Unit Regression Network (TURN)

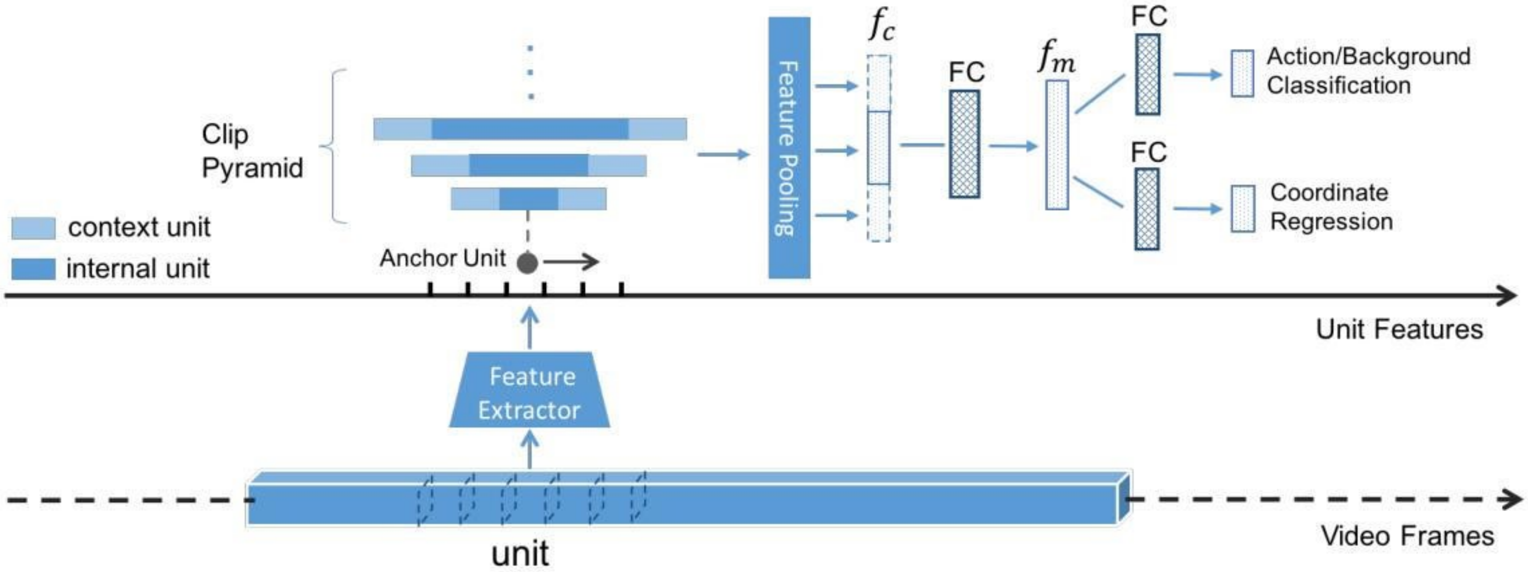
Adapt from Faster-RCNN

Avoid processing high overlapping windows



$$f_c = P(\{u_j\}_{s_u - n_{ctx}}^{s_u}) \parallel P(\{u_j\}_{s_u}^{e_u}) \parallel P(\{u_j\}_{e_u}^{e_u + n_{ctx}})$$

Temporal Unit Regression Network (TURN)



Multi-tasks

Positive: t-IoU>0.5

$$o_s = s_{clip} - s_{gt}, \;\; o_e = e_{clip} - e_{gt}$$

$$L_{reg} = \frac{1}{N_{pos}} \sum_{i=1}^{N} l_i^* |(o_{s,i} - o_{s,i}^*) + (o_{e,i} - o_{e,i}^*)|$$

$$L = L_{cls} + \lambda L_{reg}$$

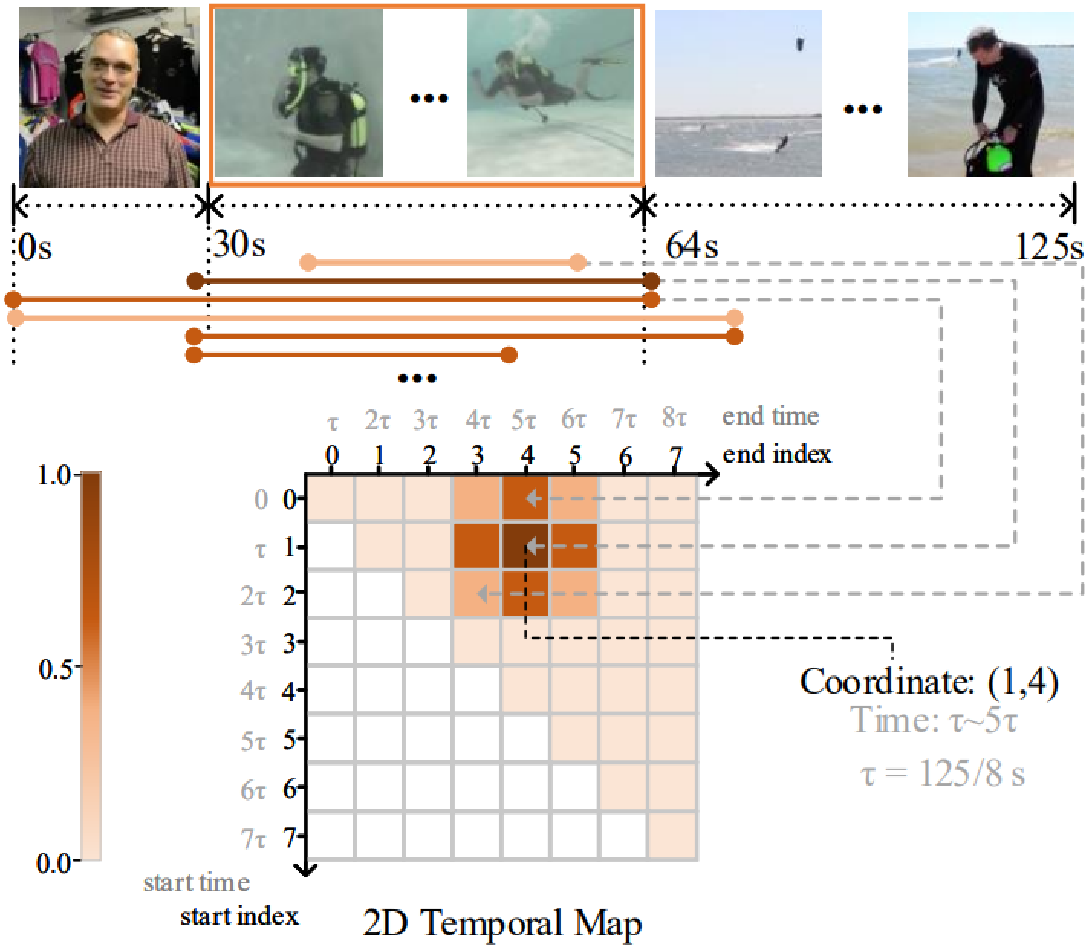
l_i^*: 0 background, 1 positive samples

Temporal Unit Regression Network (TURN)

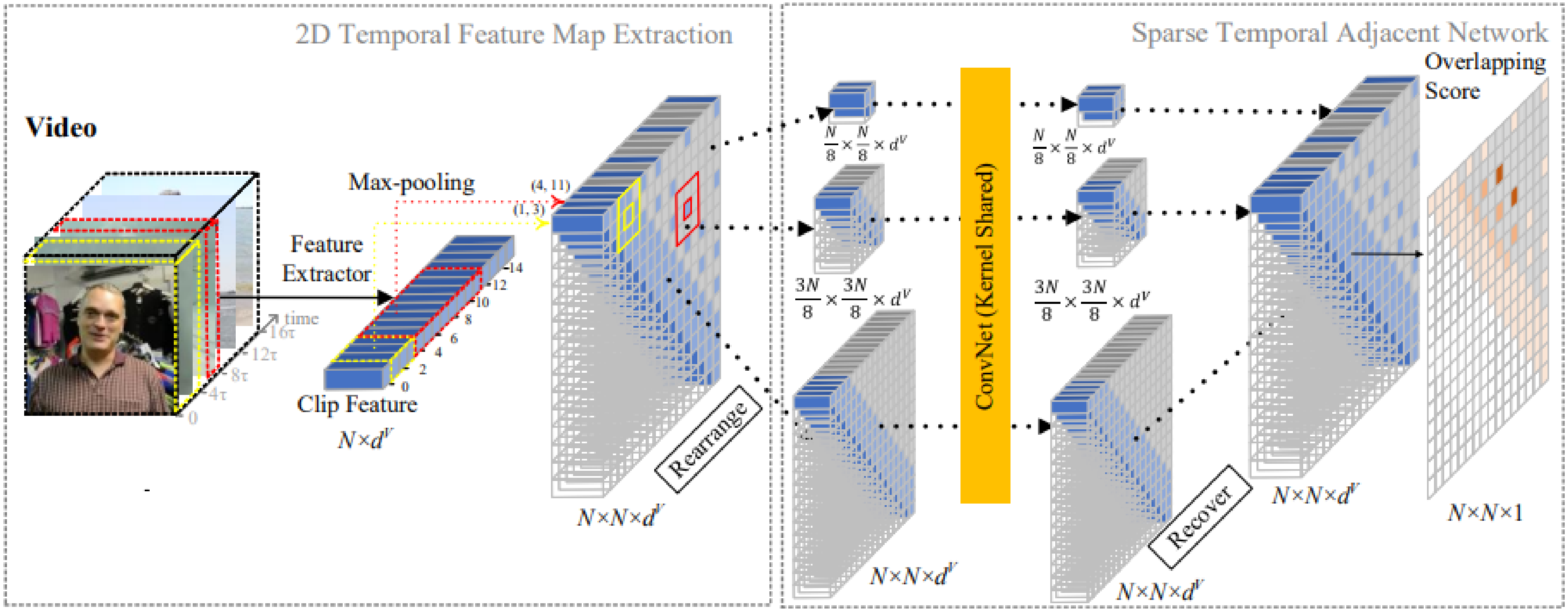
Inference time

- Classifier determines the Background/Action Class
- Regression refine the window generated by anchor
- Post-processing: NMS

2D-TAN [AAAI'20]



2D-TAN [AAAI'20]

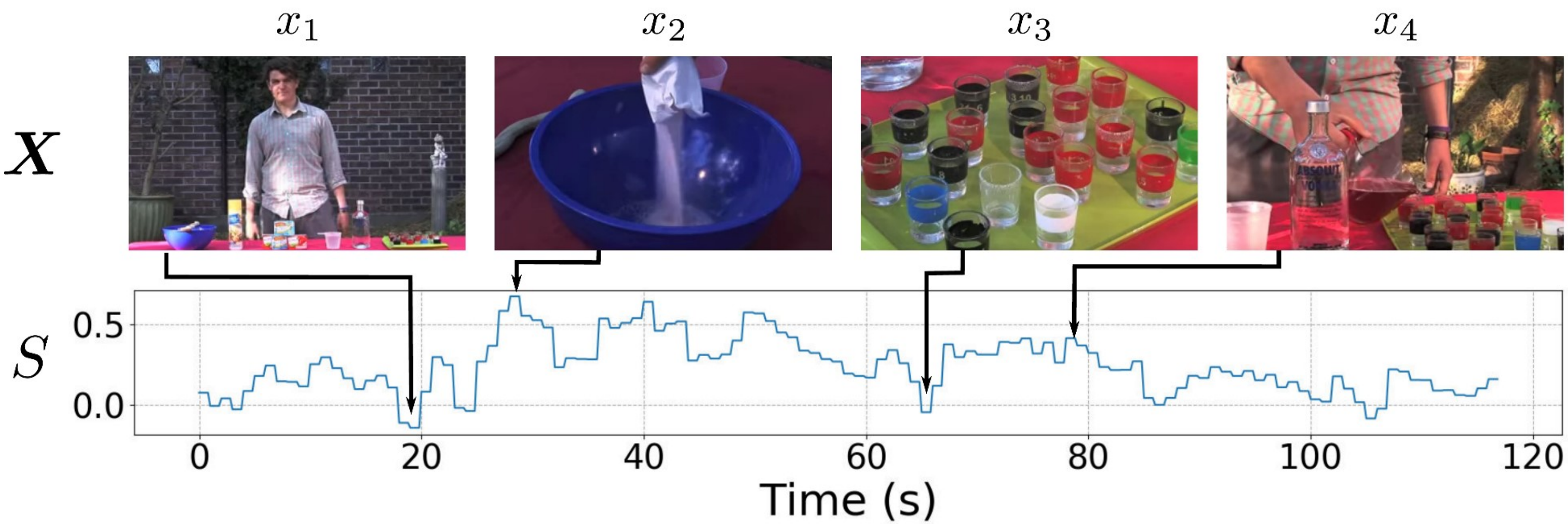


Section 4.3

Actioness based

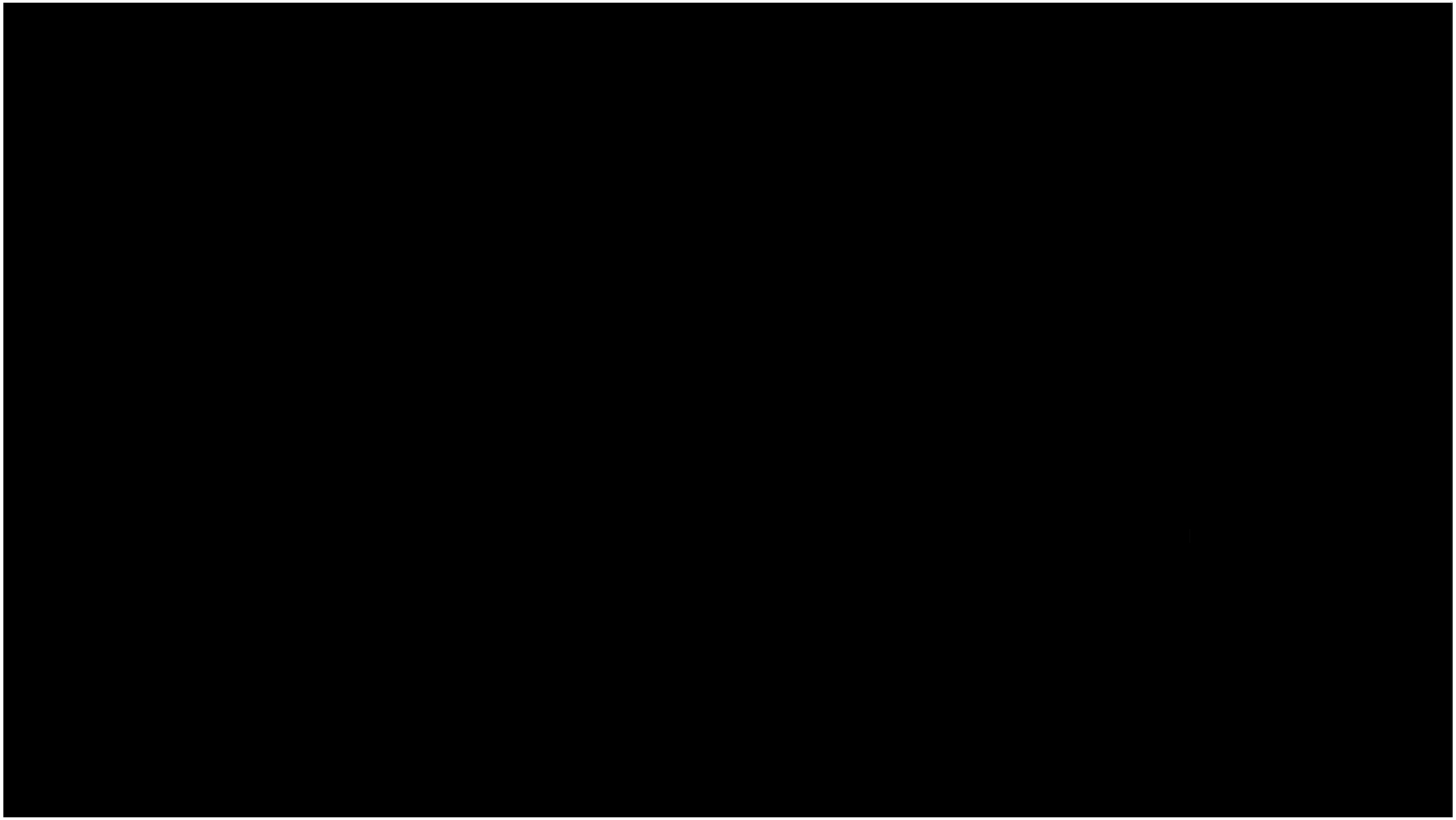
Actionness

- The signal makes salient actions stand out against the background and we term these the "actionness"



Actionness

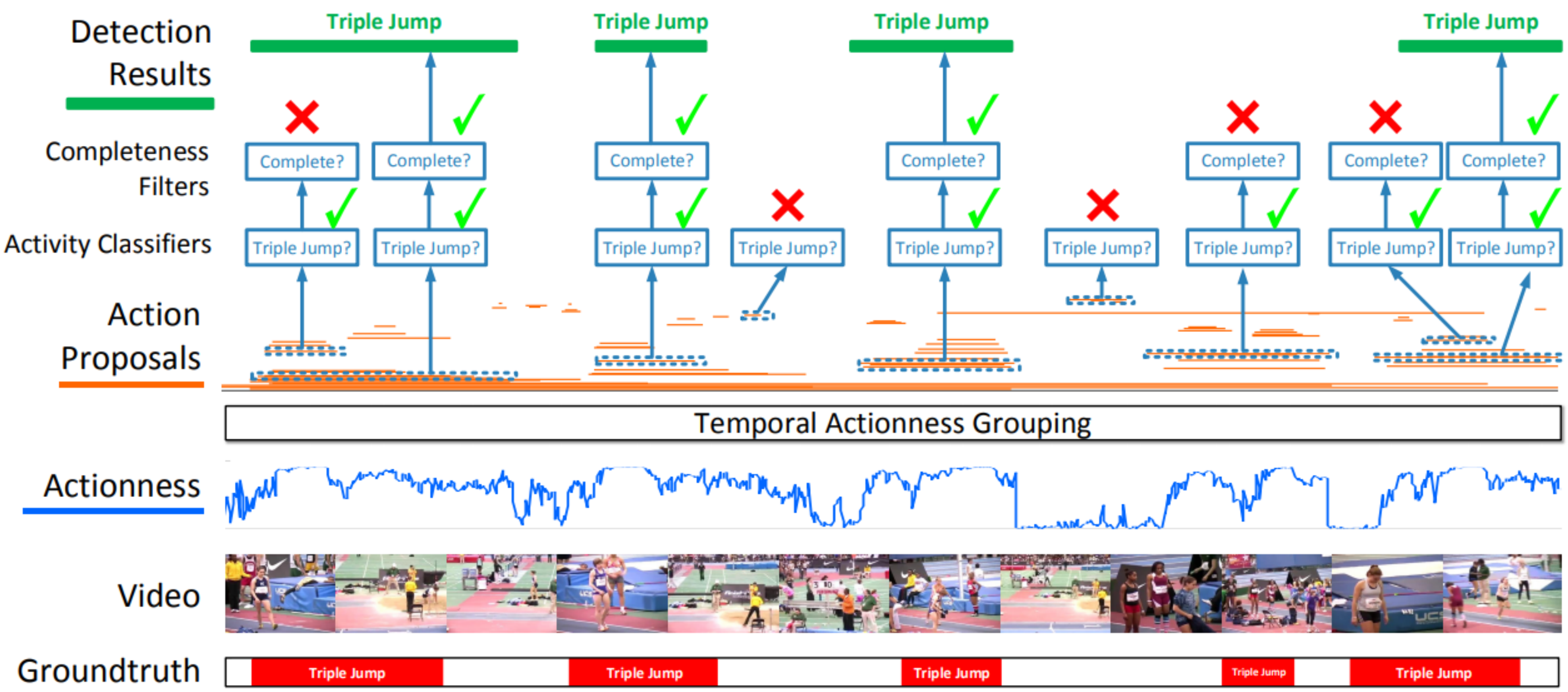
Actionness



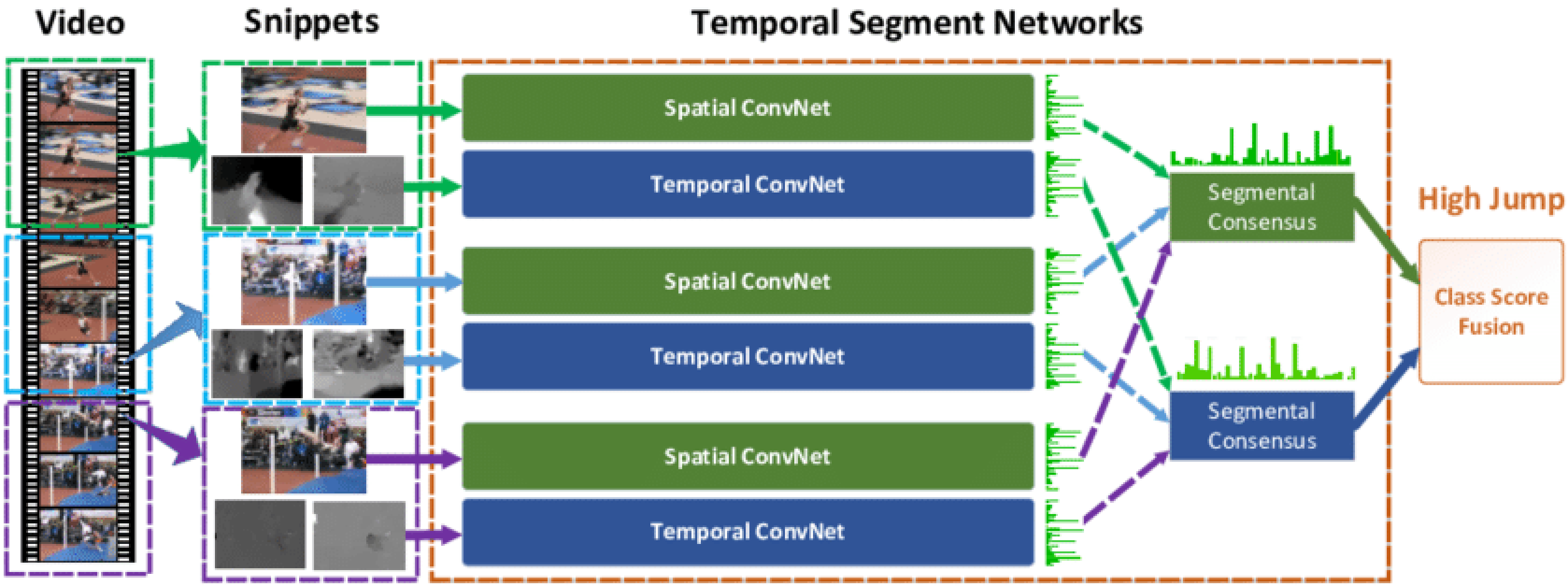
Temporal Actioness Grouping (TAG)

Generating temporal proposals (Actionness from TSN)

Classifying proposed candidates (TSN)



TSN

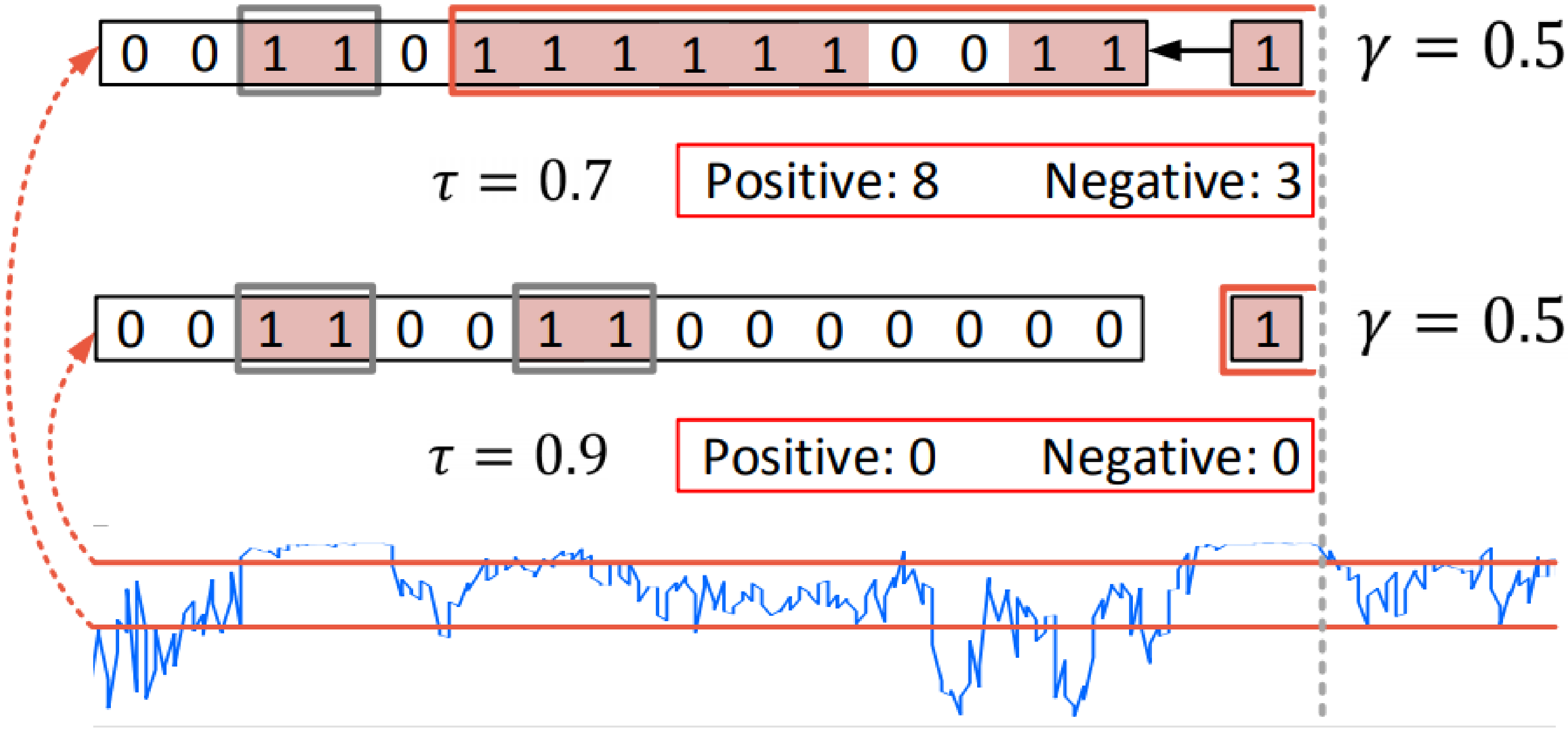


Video
Snippets
Temporal Segment Networks
Spatial ConvNet
Temporal ConvNet
Spatial ConvNet
Temporal ConvNet
Spatial ConvNet
Temporal ConvNet
Segmental Consensus
Segmental Consensus
High Jump
Class Score Fusion

Actionness

- Different threshold generate different proposals
- NMS remove overlapping ones
- Threshold: actionness, tolerence

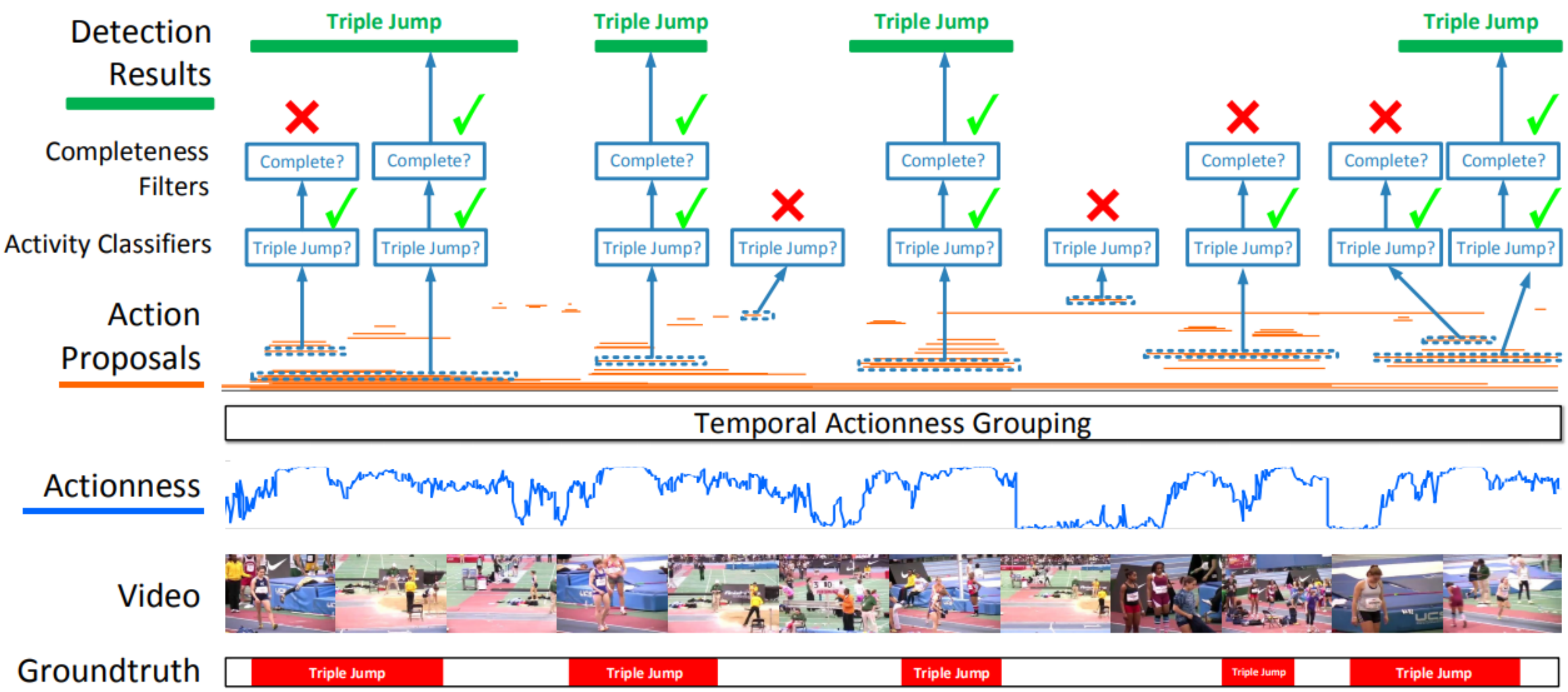
$$\gamma = \frac{\#1}{\#total}$$



Temporal Actioness Grouping (TAG)

Generating temporal proposals (Actionness from TSN)

Classifying proposed candidates (TSN)



Completeness

- Sc: Complete Score, Pa: Action Score

$$S_{Det} = P_a \times \exp(S_c)$$

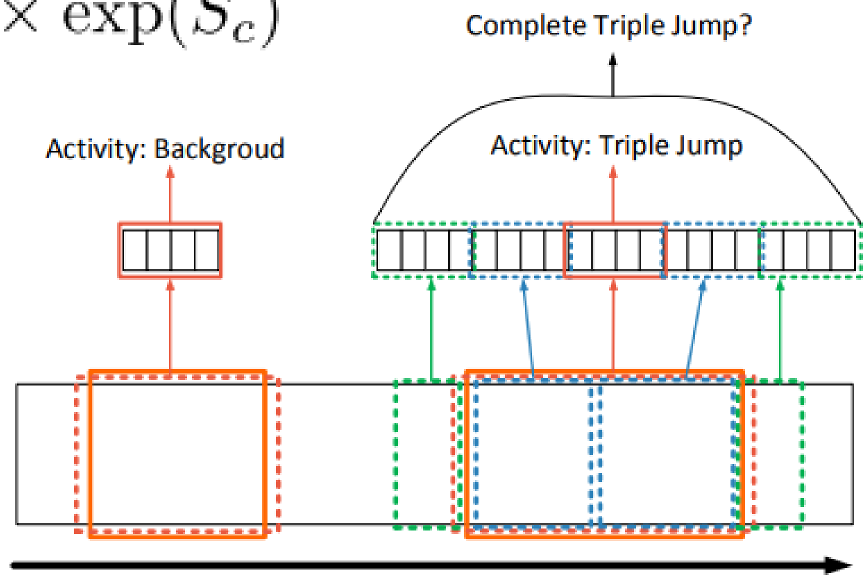
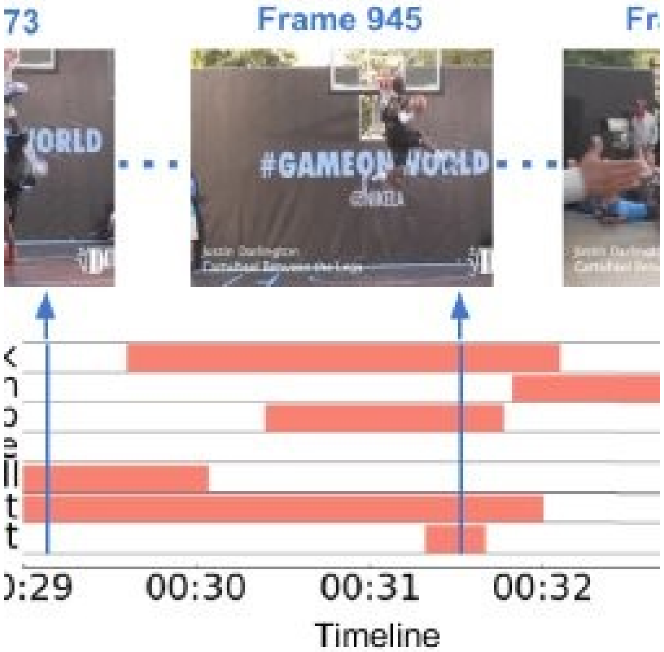


Figure 4. The proposal classification module. The activity classifiers first remove background proposals and classify the proposals to its activity class. Then the class-aware completeness filters evaluate the remaining proposals using features from the temporal pyramid and surrounding fragments.

Drawbacks

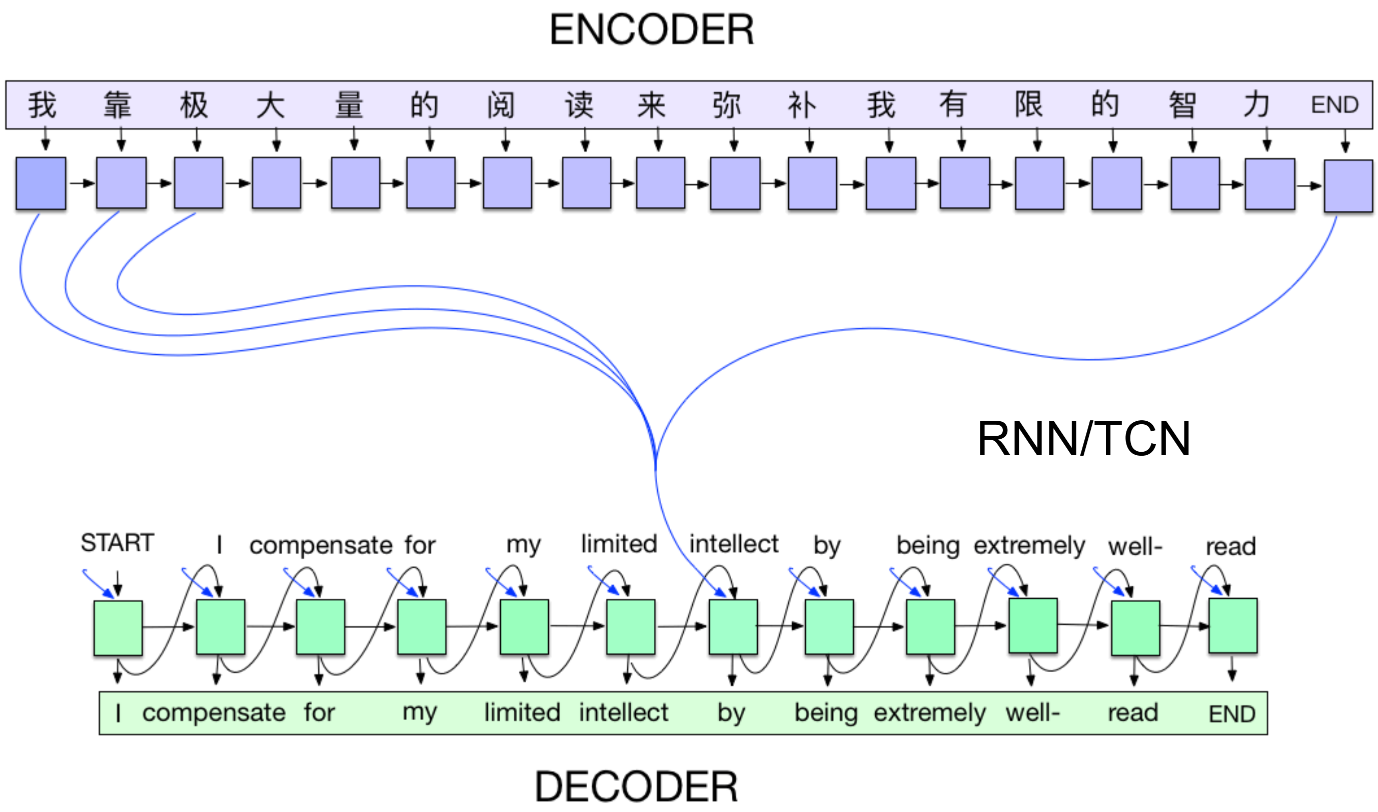
Hard to handle densely annotated videos



Section 4.3

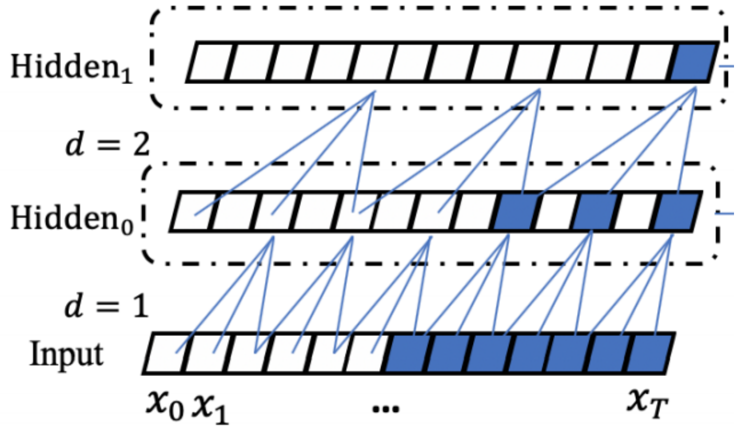
Seq-to-Seq

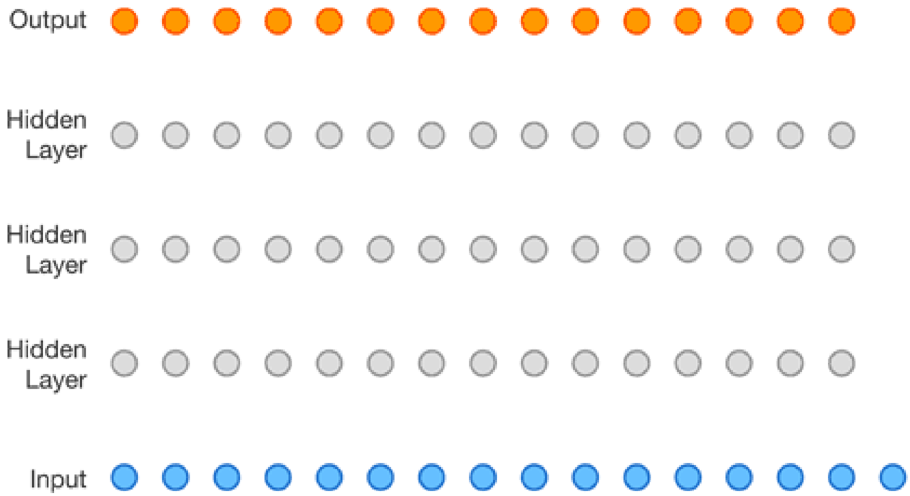
Seq2Seq


ENCODER
我 靠 极 大 量 的 阅 读 来 弥 补 我 有 限 的 智 力 END
RNN/TCN
START I compensate for my limited intellect by being extremely well- read
I compensate for my limited intellect by being extremely well- read END
DECODER

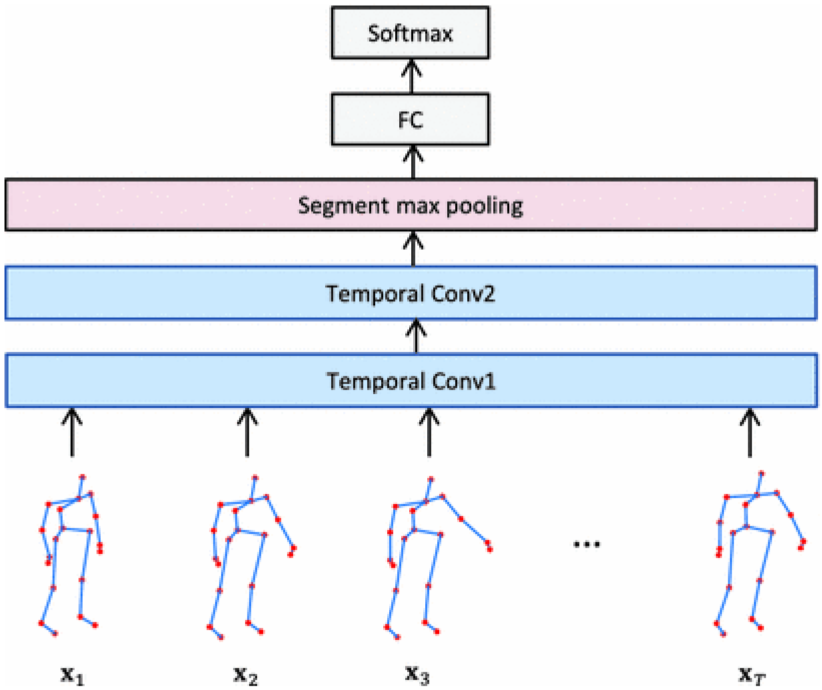
Temporal Convolution Network (TCN)

1 dimensional-convolution

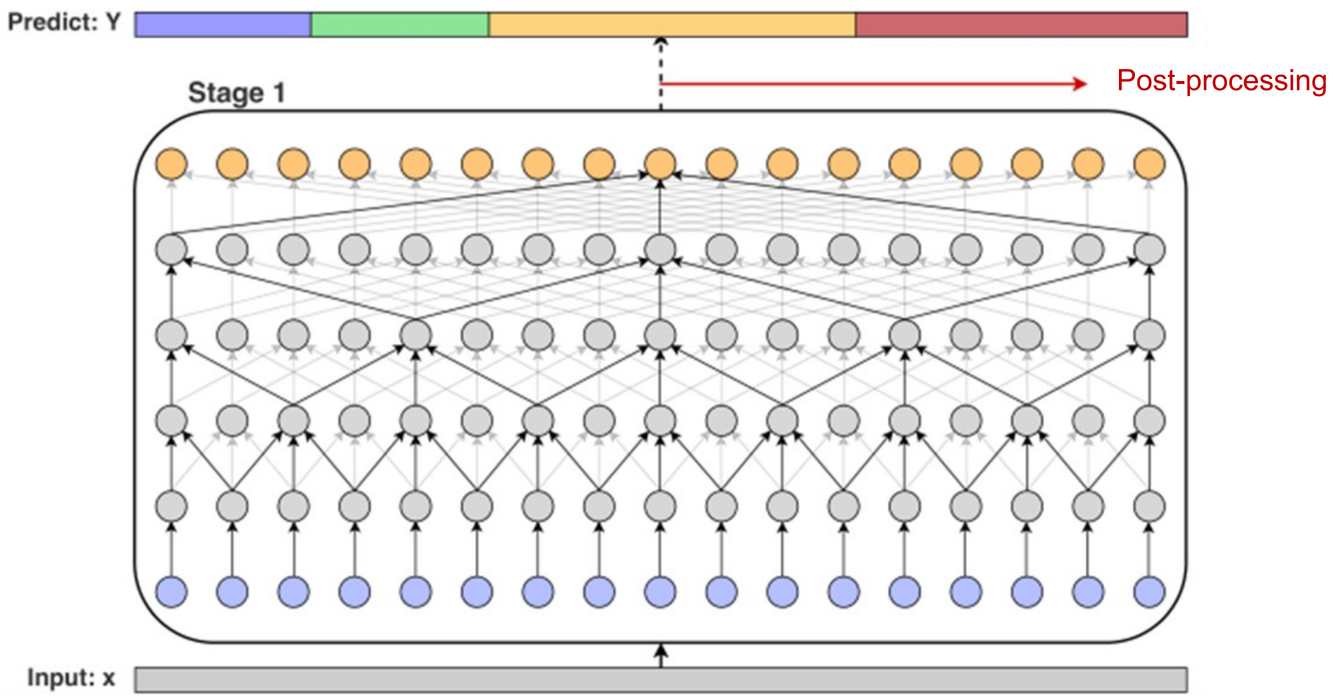




Temporal Convolution Network (TCN)



Temporal Convolution Network (TCN)



Summary

- Sliding window
- Anchor-based
- Actioness
- Seq-to-Seq

Travaux Pratiques

Practice

Sliding window

https://github.com/dairui01/TP_Sliding_window

Practice

SCNN:

https://github.com/zhengshou/scnn

TAG:

https://github.com/yjxiong/action-detection

TURN:

https://github.com/jiyanggao/TURN-TAP

Thanks!

E-mail: rui.dai@inria.fr