Action Recognition and Detection by Combining Motion and Appearance Features

Limin Wang, Yu Qiao, and Xiaoou Tang

The Chinese University of Hong Kong, Hong Kong

Shenzhen Institutes of Advanced Technology, CAS, China

Outline

- Introduction
- Method
- Result
- Conclusion

I. Introduction

Introduction

- Action recognition from short clips has been widely studied recently (e.g. HMDB51, UCF101)
- Action recognition and detection from temporally untrimmed videos received less attention.
- THUMOS 14 challenge focuses on this more difficult problem.

State of the art results

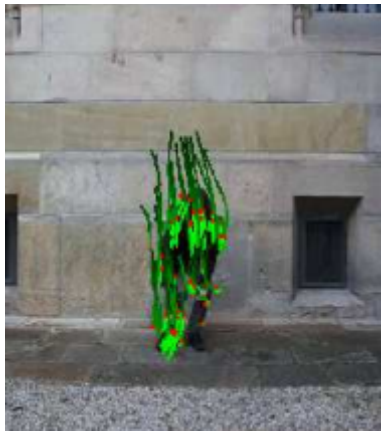
UCF 101 Recognition	THUMOS 14 Recognition	THUMOS 14 Detection
87.9%	71.0%	33.6%

Introduction

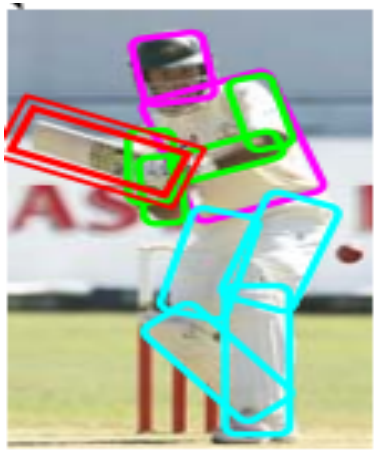
- There are two key problems in this challenge:
  - How to conduct temporal segmentation of continuous video sequence.
  - How to represent the video clips for action recognition.
- In our current method, we only try a simple segmentation method.
- We mainly focus on how to extract effective visual representation.

Introduction

- What kinds of information are important for action understanding from video.



Motion trajectory



Pose

Interacting object

Scene category

Dynamic motion cue

Static appearance cue

Related Works

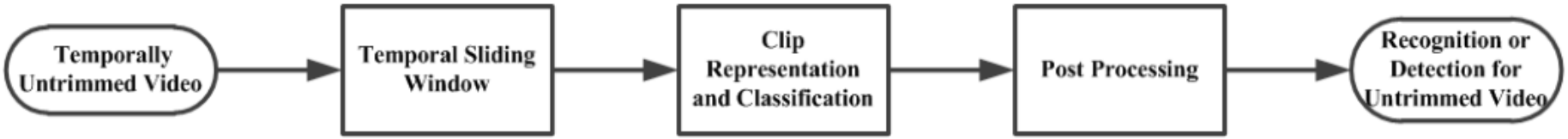
- Dynamic motion cues:
  - Low-level features: STIPs [Laptev 05], Improved Trajectories [Wang13] etc.
  - Mid-level representation: Motionlet [Wang 13], Discriminative patches [Jain etc 13], Motion atoms and phrases[Wang13] etc.
  - High-level representation: Action Bank [Sadanand12] etc.
- From THUMOS 13, it is known that Fisher vector of improved dense trajectories is very effective.
- From our experience, the mid-level representation is complementary to FV of IDT.

Related Works

- Pose: Poselet [Bourdev09], Mixture of parts [Yang13] etc.
- Object: Deformable part model [Felzenszwalb10] etc.
- Scene: Gist [Oliva01], Discriminative Patches [Singh12] etc.
- Recently, deep CNN obtains much better results with theses tasks.
  - Deep CNN will need a large number of training samples with supervised labels.

II. Method

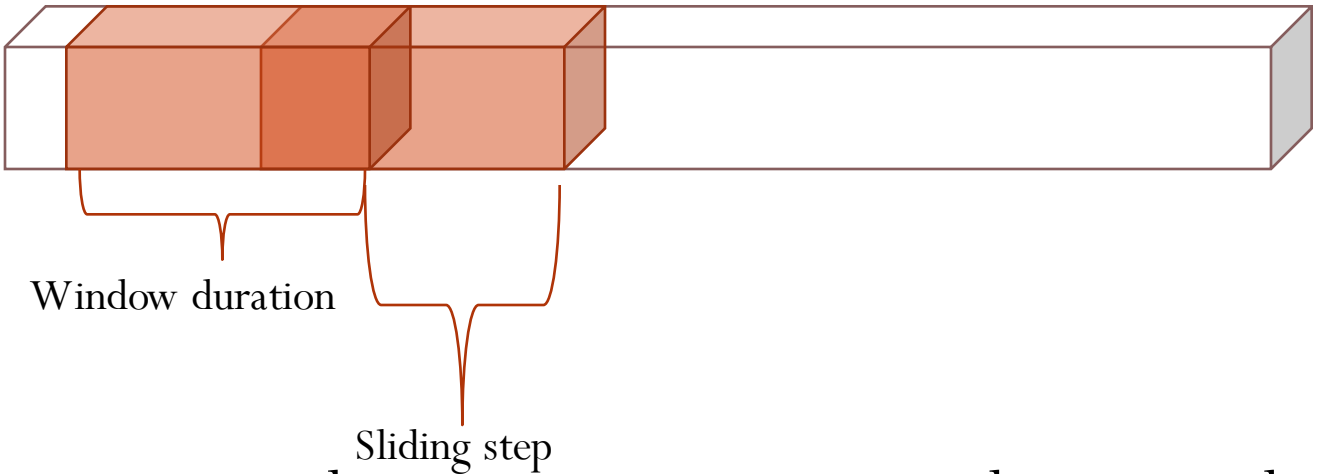
Overview of Our Method



- We propose a simple pipeline for action recognition and detection from temporally untrimmed videos
- It is composed of three steps: temporal segmentation, clip representation and recognition, post-processing

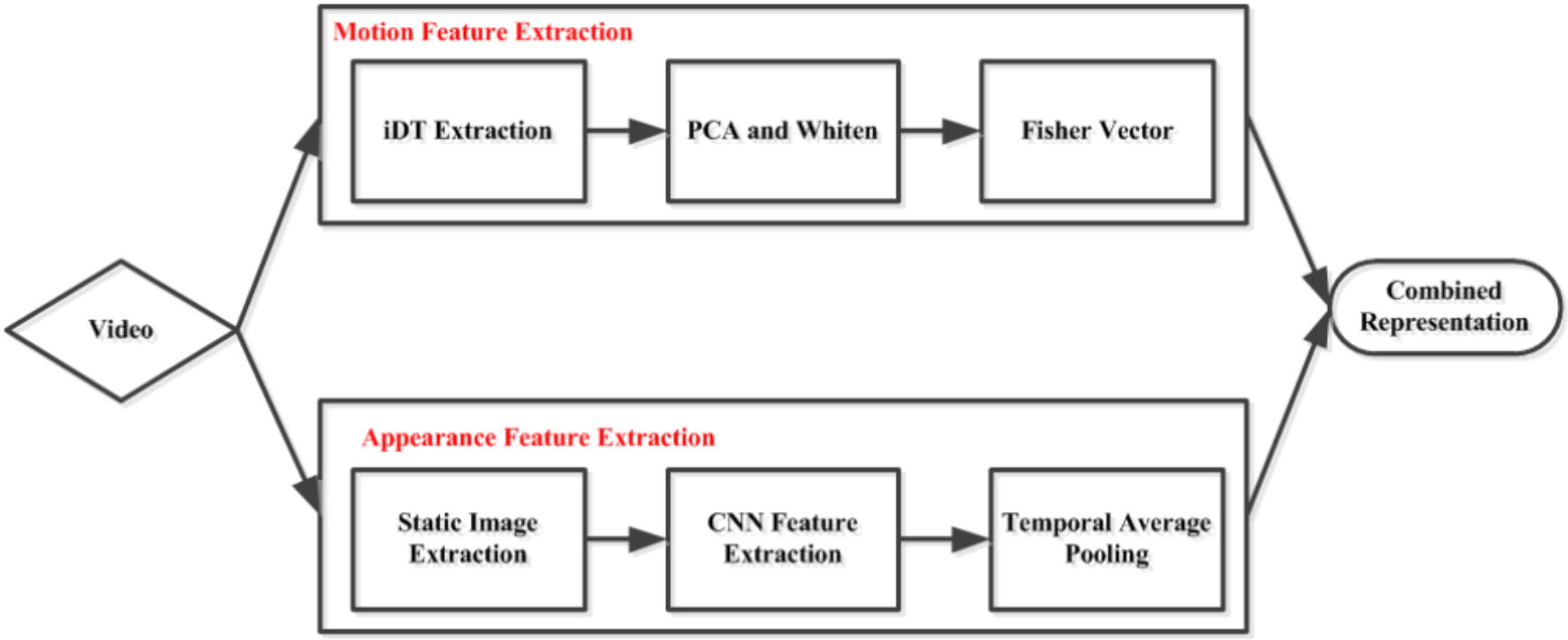
Temporal Segmentation

- We use the simple temporal sliding window method.

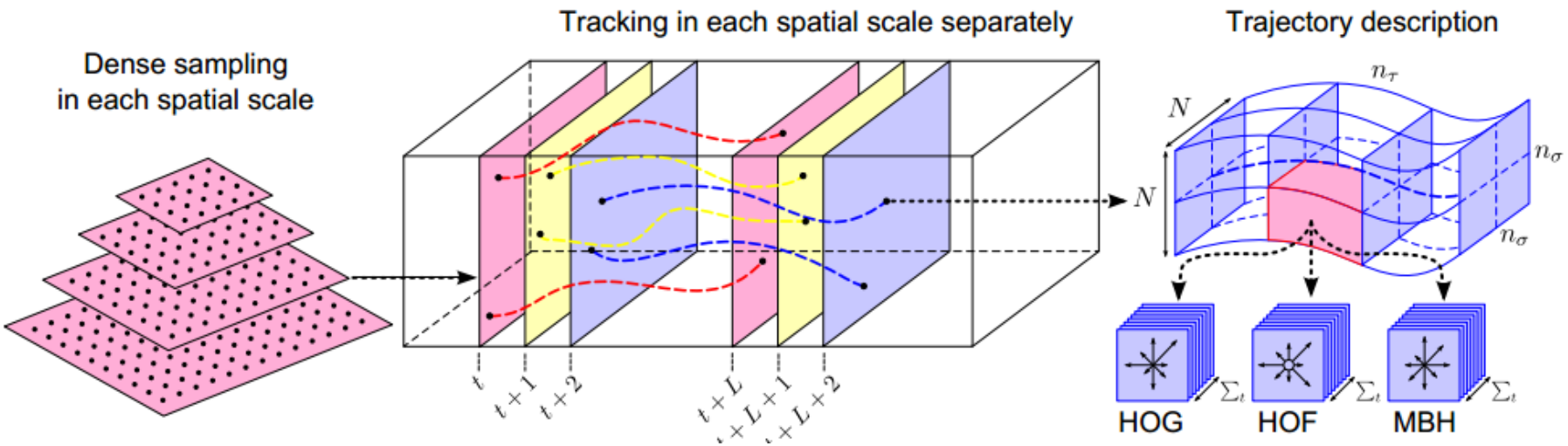


- In current implementation, we just a single temporal scale for sliding window (duration =150 frames, step = 100 frames)

Clip Representation



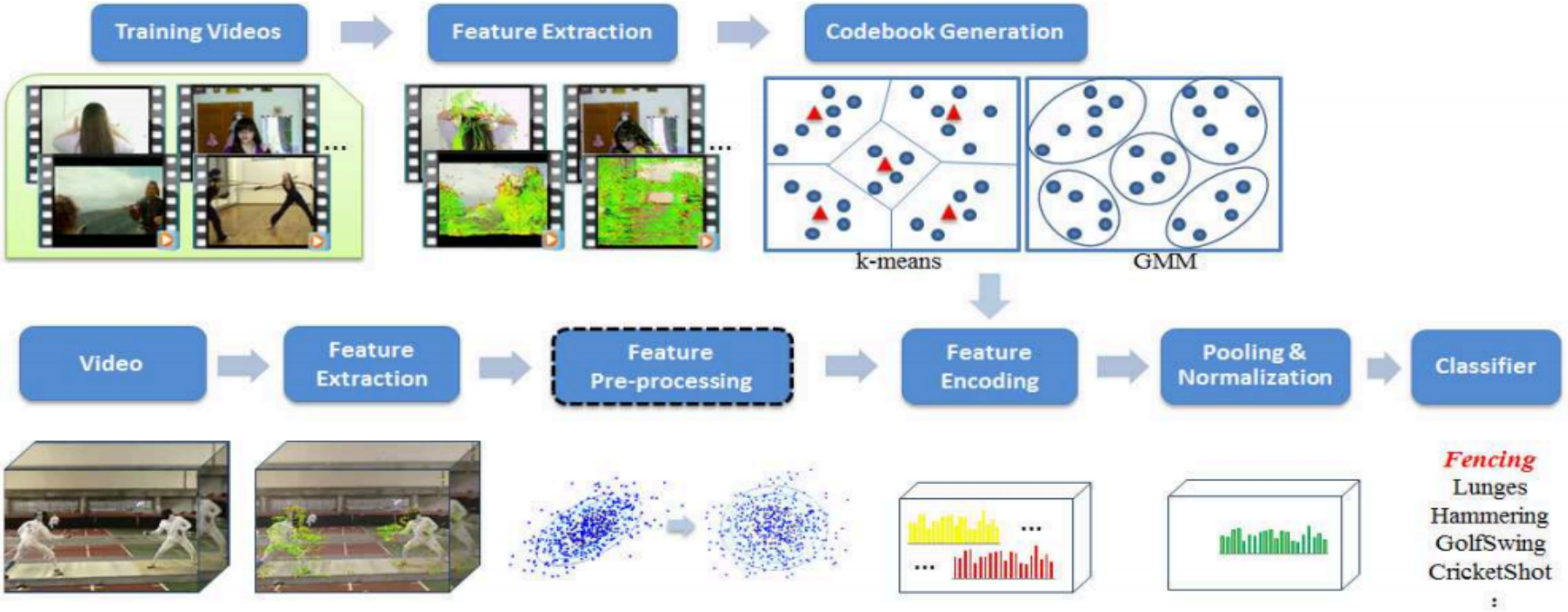
Improved Dense Trajectories



- Improved dense trajectories extract: HOG, HOF, MBHx, MBHy
- Improved dense trajectories firstly estimate camera motion and compensate it.

[1] Heng Wang and Cordelia Schmid, Action Recognition with Improved Trajectories, in ICCV 2013.

Bag of Visual Words



- There many choices in each step of BoVW and implementation details are important.
- Super vector encoding outperforms others.

[1] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. CoRR abs/1405.4506, 2014

Fisher Vector

- Given a set of descriptors: $X = [x_1, x_2, \cdots x_N] \in R^{D \times N}$, we learn a generative GMM: $p(x; \theta) = \sum_{k=1}^{K} N(x; \mu_k, \sum_k)$
- Given the descriptors from video clip, we derive the Fisher vector:

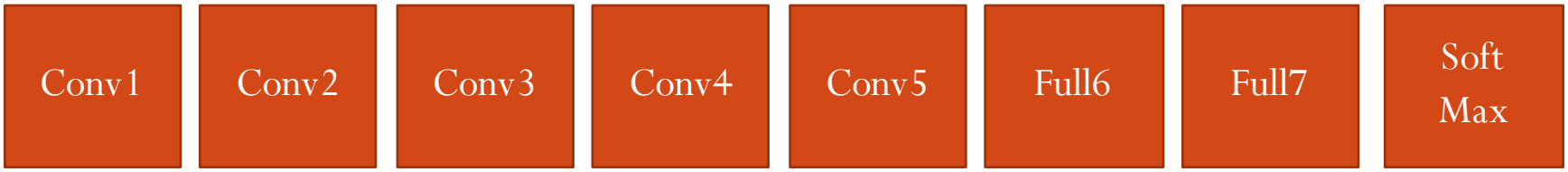
$$\mathcal{G}_{\mu_k}^{\mathbf{X}} = \frac{1}{\sqrt{\pi_k}} \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \left(\frac{\mathbf{x}_i - \mu_k}{\sigma_k} \right),$$

$$\mathcal{G}_{\sigma_k}^{\mathbf{X}} = \frac{1}{\sqrt{2\pi_k}} \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \left[\frac{(\mathbf{x}_i - \mu_k)^2}{\sigma_k^2} - 1 \right],$$

- In our current implementation, we choose power $\ell 2$-normalization ($\alpha = 0.5$) to obtain final representation S :

$$\mathcal{S} = [\mathcal{G}_{\mu_1}^{\mathbf{X}}, \mathcal{G}_{\sigma_1}^{\mathbf{X}}, \cdots, \mathcal{G}_{\mu_K}^{\mathbf{X}}, \mathcal{G}_{\sigma_K}^{\mathbf{X}}], \quad \mathcal{S} = \frac{\text{sign}(\mathcal{S})\sqrt{|\mathcal{S}|}}{\|\sqrt{\mathcal{S}}\|_2}.$$

CNN Activation



- We firstly resize frame as 256*256 and then crop a region as 227*227
- We use the Caffe implementation of the CNN described by [Krizhevsky et al.]
- We extract the activation of Full7 as CNN features (4096 dimensions) and conduct average pooling over different crops.

[1] Jia, Y.: Caffe: An open source convolutional architecture for fast feature embedding. (2013)

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)

CNN Fine-tuning

- We fine tune the parameters of CNN using the UCF101 dataset through 50,000 iterations.
- We extract 10 frames from each video and use the video label as frame label to fine tune the ImageNet training result
- The batch size is set as 256, drop out ratio as 0.5, iteration: 50,000.

Results of Split 1 on UCF 101

Without fine tuning	With fine tuning
65.3%	70.5%

Feature Fusion

- For appearance feature, we extract CNN activations on 15 frames and perform average pooling to get the representation of video clip.
- For motion feature, we use FV for each descriptor of IDT features independently.
- Both appearance and motion features are firstly normalized and then concatenated as a hybrid representation.
- The fusion weight: appearance (0.4), motion (1.0).

Classifier

- We choose linear SVM classifier for action recognition and detection using Training dataset and Background dataset.
- For multi-class classification, we use the one vs. all training scheme.
- To void false detections, we randomly select 4,000 clips from the background dataset as negative examples when training SVM for each action class.

Post Processing

- To obtain the action recognition result for the whole video sequence, we conduct max pooling over the recognition result of video clip.
- To avoid false positive recognition and detection, we simply use three thresholds:
  - Clip level threshold t_1: for each clip, there are at most t_1 action instances.
  - Sequence level threshold t_2: for each sequence, there are at most t_2 action instances.
  - SVM score threshold t_3: elimination of detection instance with confidence score lower than t_3.

III. Experimental Results

Experiment Results 1

Action recognition result of Split1 on UCF 101

Motion Feature	Appearance Feature	Fused Feature
85.3%	70.5%	89.1%

Action recognition result on validation dataset of THUMOS 14

Motion Feature	Appearance Feature	Fused Feature
57.1%	48.2%	65.3%

Experiment Results 2

Action recognition result on test dataset of THUMOS 14

(t_1,t_2)	(1,10)	(1,20)	(5,10)	(5,20)	(101,101)
Result	0.617	0.6177	0.6196	0.6174	0.6201

Action detection result on test dataset of THUMOS 14

(t_1,t_2)	(1,0.5)	(1,0)	(1,-0.5)	(1,-1)
Overlap=0.1	0.1080	0.1373	0.1701	0.1818
Overlap=0.2	0.1042	0.1319	0.1591	0.1700
Overlap=0.3	0.0891	0.1137	0.1306	0.1405
Overlap=0.4	0.0765	0.0975	0.1090	0.1174
Overlap=0.5	0.0563	0.0695	0.0775	0.0834

IV. Conclusions

Conclusions

- We prove that motion features (IDT) and appearance features (CNN) are complimentary to each other.
- For FV of IDT, implementation detail such as descriptor pre-processing, normalization operation has a great influence on final performance.
- For CNN feature, fine-tuning on UCF101 dataset helps to improve recognition performance.
- In the future, we may consider designing more effective segmentation algorithm or performing both tasks simultaneously.

Another Work on Action Detection

- We design a method unifying action detection and pose estimation in ECCV 2014:

Thank You!

lmwang.nju@gmail.com

- Welcome to our ECCV poster presentation:
  - Video Action Detection with Relational Dynamic-Poselets (Session 3B).
  - Action Recognition with Stacked Fisher Vectors (Session 3B).
  - Boosting VLAD with Supervised Dictionary Learning and High-Order Statistics (Session 2B).