CUHK&SIAT Submission for THUMOS15 Action Recognition Challenge

Limin Wang, Zhe Wang, Yuanjun Xiong, Yu Qiao
The Chinese University of Hong Kong
Shenzhen Institutes of Advanced Technology, CAS

Outline

- Introduction
- Our method
- Experiments
- Conclusions

Introduction

- Previous research works focus on short cropped clips.
- From last layer, THUMOS challenge change to temporal untrimmed videos.
- It becomes more challenging and we need to consider both segmentation and classification at the same time.
- For action recognition, there are mainly two types of methods.

Related Works

- The first type of action recognition method:
  - Low-level features: STIPs, Cuboids, Dense Trajectories, Improved Trajectories.
  - BoVW representations: Histogram encoding, Sparse Encoding, VLAD, Fisher Vector.

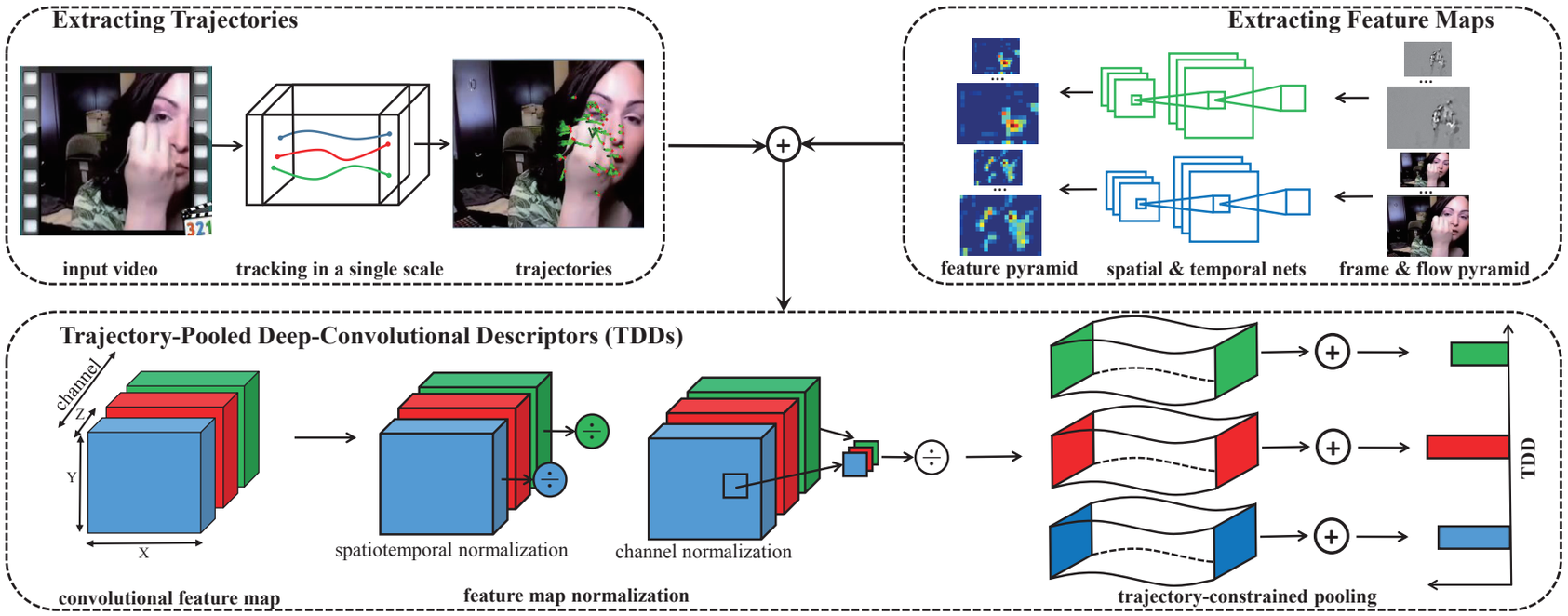
Related Works

- The second type of action recognition method:
  - Training deep neural networks on action recognition in an end-to-end manner.
  - Typical examples: 3D CNN, Two-Stream ConvNets

Related Works

- Trajectory-pooled Deep-Convolutional Descriptors(TDD)
  - Combination of Improved Trajectories and Two-Stream ConvNets.
    - Replace HOG, HOF, MBH descriptors with convolutional activations of CNNs.
    - Fisher vector meeting CNN features.
- **However, due to our late start for this challenge, we did not have the chance to try TDD on this.**

Overview of TDD



.

Algorithm	HMDB51	UCF101	Algorithm	HMDB51	UCF101
HOG [1]	40.2%	72.4%	Spatial conv4	48.5%	81.9%
HOF [1]	48.9%	76.0%	Spatial conv5	47.2%	80.9%
MBH [1]	52.1%	80.8%	Spatial conv4 and conv5	**50.0%**	**82.8%**
HOF+MBH [1]	54.7%	82.2%	Temporal conv3	54.5%	81.7%
iDT [1]	**57.2%**	**84.7%**	Temporal conv4	51.2%	80.1%
Spatial net [2]	40.5%	73.0%	Temporal conv3 and conv4	**54.9%**	**82.2%**
Temporal net [2]	54.6%	83.7%	TDD	63.2%	90.3%
Two-stream ConvNets [2]	**59.4%**	**88.0%**	TDD and iDT	**65.9%**	**91.5%**

Code and model is available at https://wanglimin.github.io/tdd.html

Overview of Our Method

- Action Recognition on Temporal Untrimmed Videos:
  - Step1: Temporal segmentation videos into clips.
  - Step2: Classification of each clip.
  - Step3: Obtaining the score of video from clip scores

Step1: Temporal Segmentation

- We first calculate the color histogram for each frame.

- We evaluate the difference over the color histograms of consecutive frames.

- If the difference is larger than a threshold, a shot boundary will be detected.

- Post-processing: clips of less than 20 frames will be eliminated

Step2: Classification of Clip

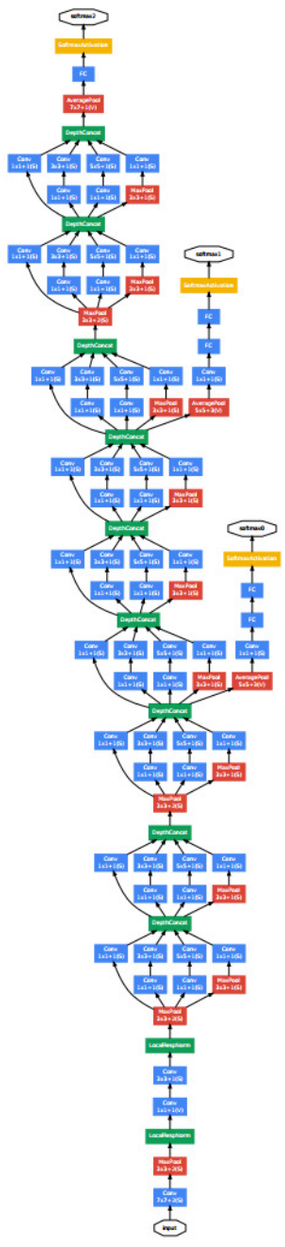
- For each clip, we try two kinds of classification methods:

- We try very-deep two-stream ConvNets

- We use the traditional of IDTs+FV.

Network Architectures

Arch.	conv1	conv2	conv3	conv4	conv5	full6	full7	full8
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop-out	4096 drop-out	1000 soft-max
CNN-M	96x7x7 st. 2, pad 0 LRN, x2 pool	256x5x5 st. 2, pad 1 LRN, x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x2 pool	4096 drop-out	4096 drop-out	1000 soft-max
CNN-S	96x7x7 st. 2, pad 0 LRN, x3 pool	256x5x5 st. 1, pad 1 x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x3 pool	4096 drop-out	4096 drop-out	1000 soft-max



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

AlexNet, Clarifai, OverFeat

deep networks

GoogLeNet

VGGNet

very-deep networks

Very-deep Two-stream ConvNets

- We aim to explore the effect of depth of network on action recognition.

- For spatial nets, we try three architectures: Clarifai net, GoogLeNet, VGG Net (16-layer)

- Due to smaller training dataset (13,320 clips), we pre-train these nets on ImageNet dataset.

Very-deep Two-stream ConvNets

- For temporal net, we also try three different architectures: Clarifai net, GoogLeNet, VGGNet (11-layer).

- We train these nets from scratch on the UCF101 datasets.

- To avoid over-fitting problem, we use strong dropout for fully-connected layers.

- For two-stream ConvNets, we just average the prediction scores of spatial nets and temporal nets.

iDT Features and Fisher Vector

- For hand-crafted representations, we use iDT features with HOG, HOF, MBHx, MBHy descriptors.

- For each descriptors, we choose Fisher vector to aggregate them into a high-dimensional representation.

- We first employ PCA and whiten pre-processing to de-correlate and reduce dimension by a factor of 2.

- We then train a Gaussian Mixture Model (256 mixtures) to Fisher vector representation.

Classifier

- For action recognition, we use linear SVM as classifiers.

- For the fusion of multiple descriptors, we concatenate the Fisher vector representation of them before SVM training.

- For multi-class problem, we resort to the one-vs-all training settings.

Fusion of ConvNets and Linear SVM

- We fuse the prediction results of two-stream ConvNets and linear SVM.

- Before fusion, we use linear transformation to normalize the range of SVM score in [0,1].

Step 3: Recognition for Untrimmed Video

- We have obtained recognition scores for each clip, we design a hybrid pooling method to aggregate clip scores into video score.

- For each class, if the maximum of clip scores is larger than a threshold, then we will choose max-pooling over these scores.

- Otherwise, we will use the average pooling over these scores.

Experiments—Implementation Details

- We train the two-stream ConvNets and linear SVMs on the UCF101 dataset (13,320 clips)

- For spatial nets (fine-tuning), the initial learning rate is set to 0.01 and decreased to 0.001 after 10k iterations, and stop at 20k iterations.

- For temporal nets (from scratch), the initial learning rate is set to 0.01, decreased to 0.001 after 50k, to 0.0001 after 70k, training stop at 90k iterations.

Experiment Results 1

Spatial nets		
ClarifaiNet	GoogLeNet	VGGNet (16-layer)
42.3%	53.7%	54.5%

- We first report the performance of spatial nets on the validation dataset of THUMOS15.

- We notice that deeper architectures achieve higher performance.

Experiment Results 2

Temporal nets		
ClarifaiNet	GoogLeNet	VGGNet (11-layer)
47.0%	39.9%	42.6%

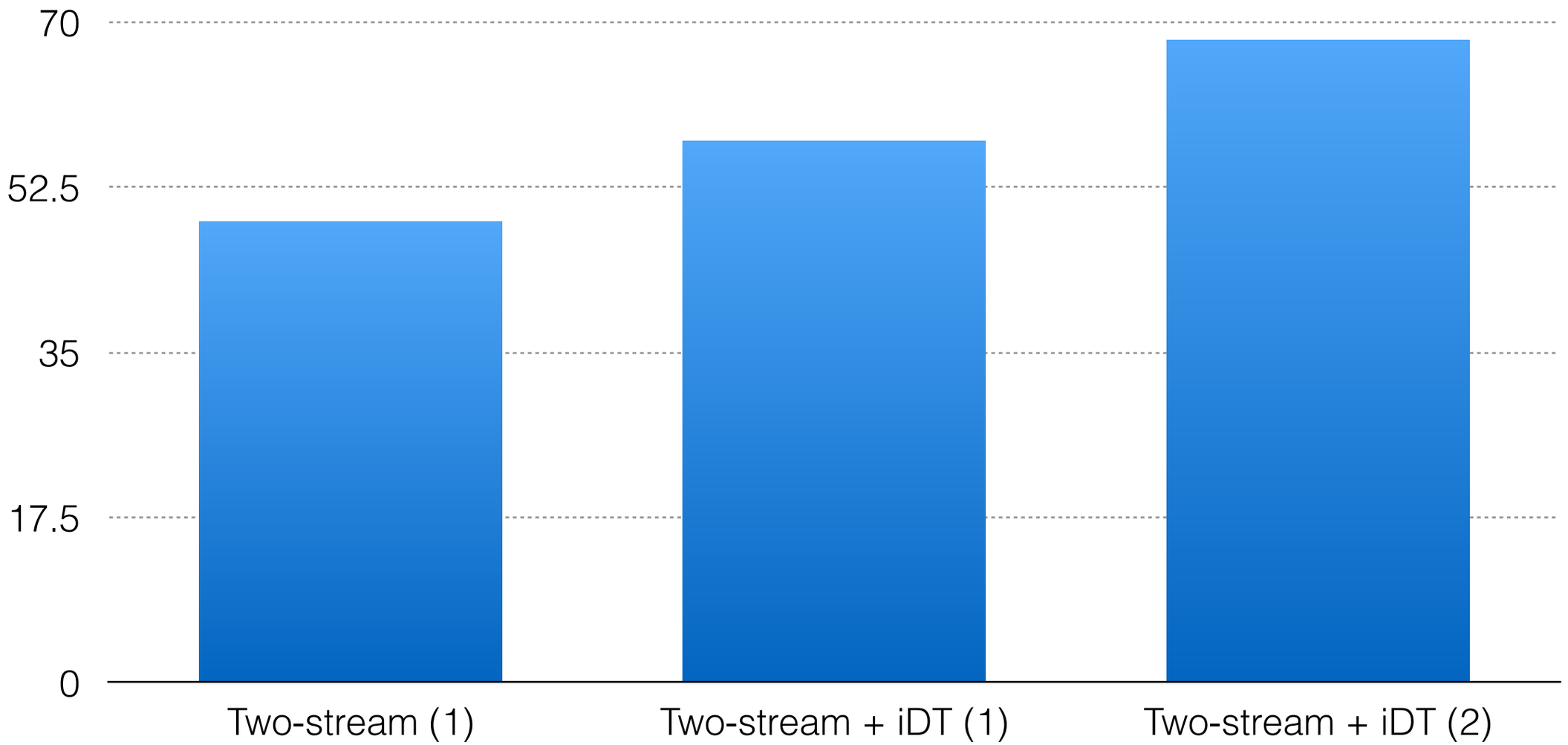
- We second investigate the performance temporal nets on the validation datasets.
- We notice that deeper architectures yet perform worse.

Experiment Results 3

Two-stream	iDTs+FV	Combine
63.7%	52.8%	68.1%

- Finally, we compare the performance of very-deep ConvNets and iDT features on the validation dataset.
- For spatial nets, we choose the architectures of VGGNet and GoogLeNet.
- For temporal nets, we choose the architecture of Clarifai net.

Results on testing dataset



1: training only on UCF101
2: training on both UCF101 and THUMOS15 validation data

Conclusions

- In this submission, we mainly explore very-deep two-stream ConvNets for action recognition.

- For spatial nets, deeper architectures perform better.

- For temporal nets, deeper architecture did not obtain better performance, perhaps due to the smaller training datasets.

- For first time, two-stream ConvNets significantly outperform iDT+FV representation.

What's Next

- Apply TDD features on top of current system.
- Explore the background training data for action recognition.
- Incorporate audio information.

Get TDD at https://wanglimin.github.io/tdd.html