ACTIVITYNET

Large Scale Activity Recognition Challenge

ACTIVITYNET

Schedule

13:30 Opening Remarks

13:40 Grouping process Models in Actor-Action Segmentation, Jason Corso (Univ. of Michigan)

14:10 Human Pose Estimation and Activity Recognition, Bernt Schiele (MPI)

14:40 Challenge Introduction

15:00 Coffee Break

15:15 Classification Task: Results and participant talks

16:15 Detection Task: Results and participant talks

17:15 Closing Remarks

ACTIVITYNET

Organizers

- General Chairs

Cees Snoek

Bernard Ghanem

Juan Carlos Niebles

- Program Chairs

Fabian Caba

Wayner Barrios

Victor Escorcia

Pascal Mettes

Sponsors

2015 Google Faculty Research Award

GPU Sponsorship for Challenge Winners

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

Challenge Hosting and Organization

ACTIVITYNET

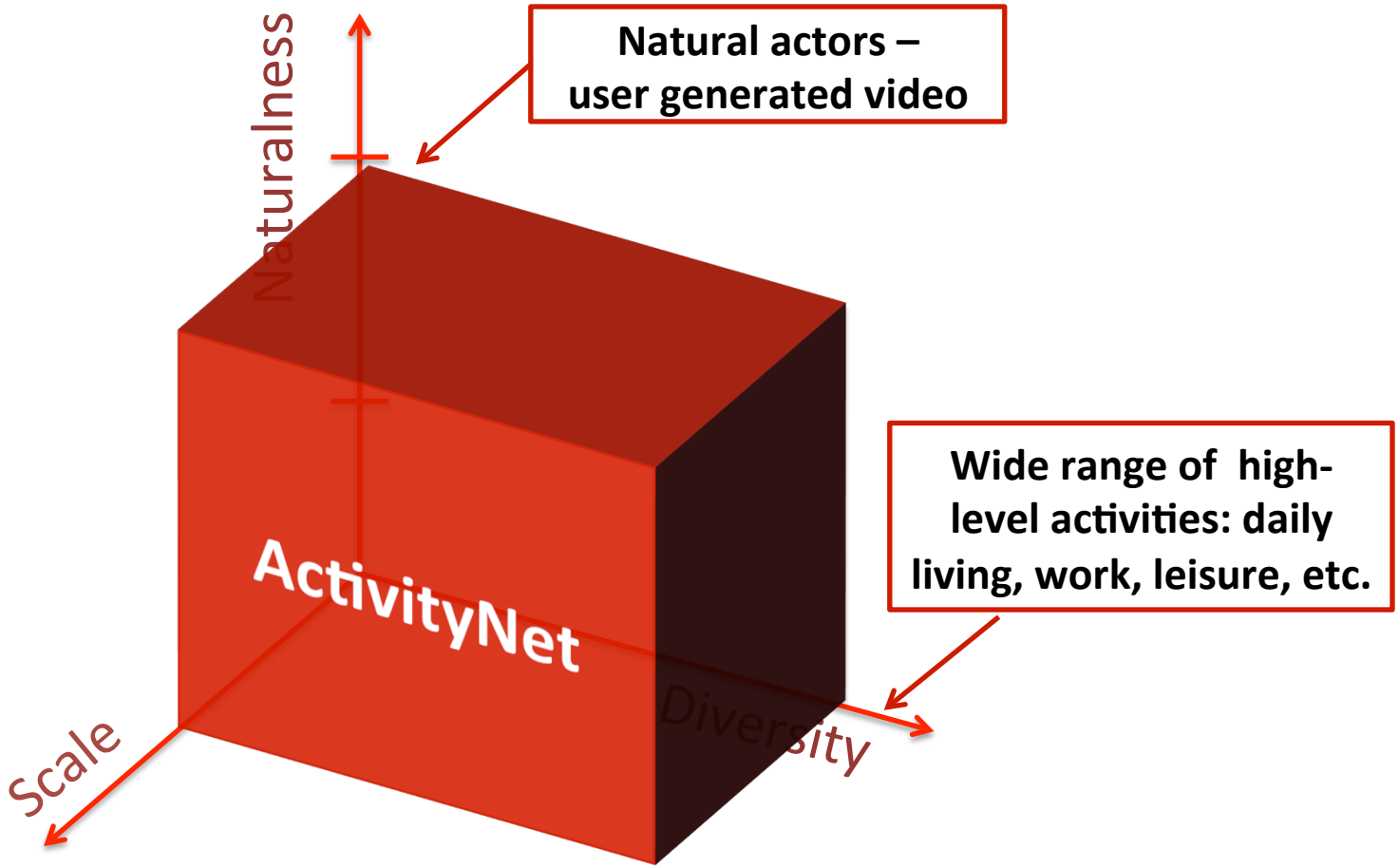
Challenge Introduction

ACTIVITYNET

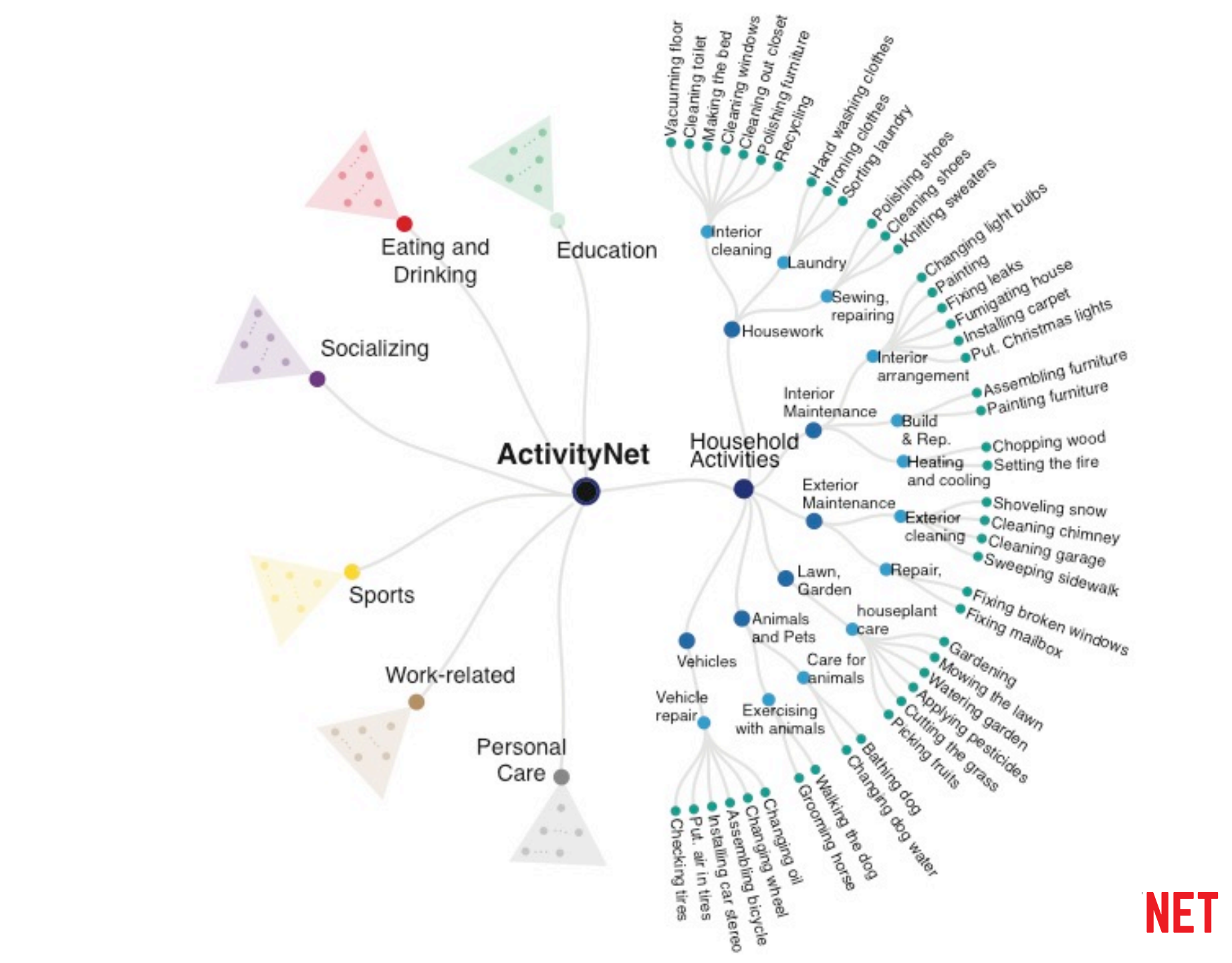
Recognize all activities in daily life

ACTIVITYNET

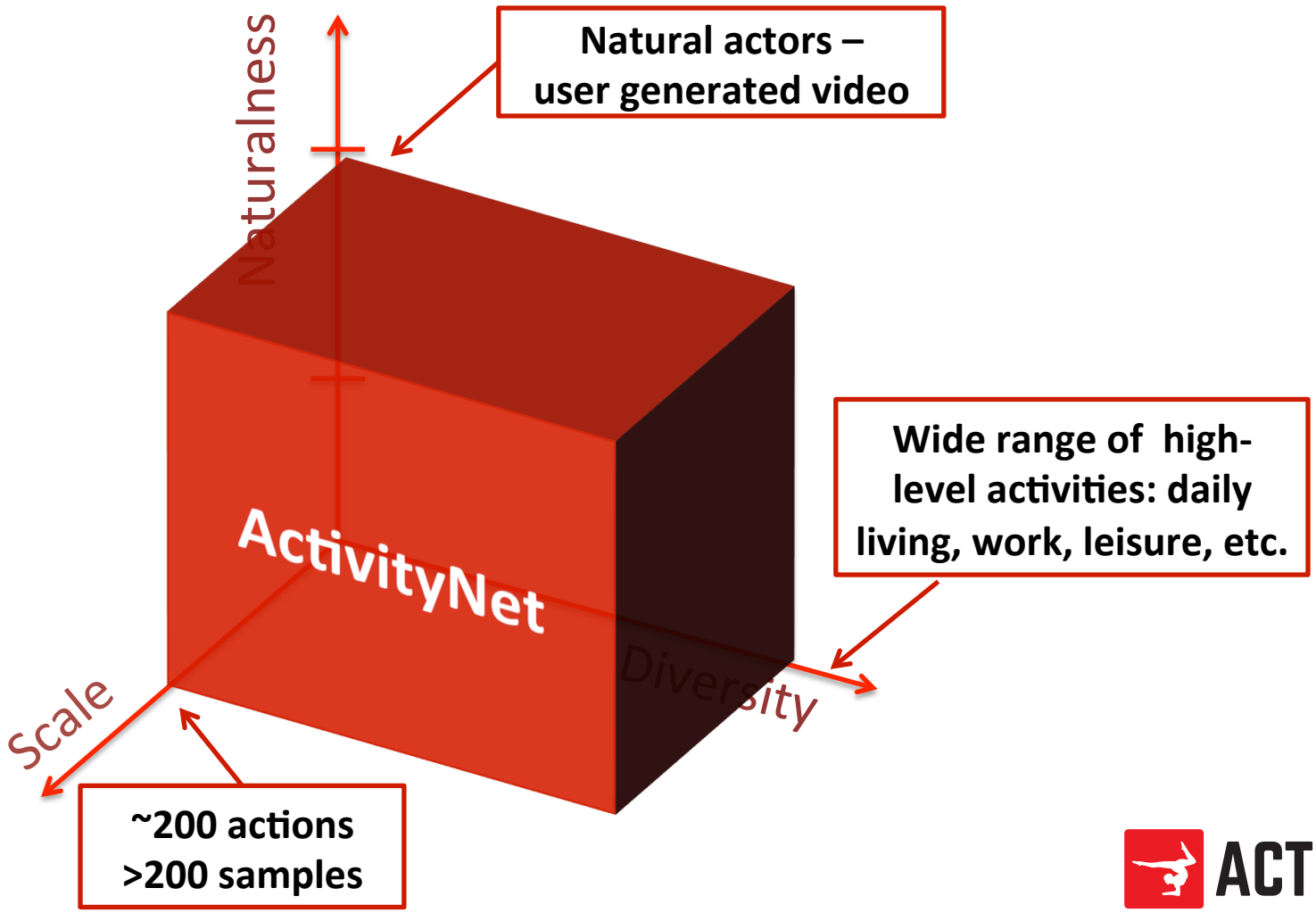
Human Activity Benchmark



ACTIVITYNET

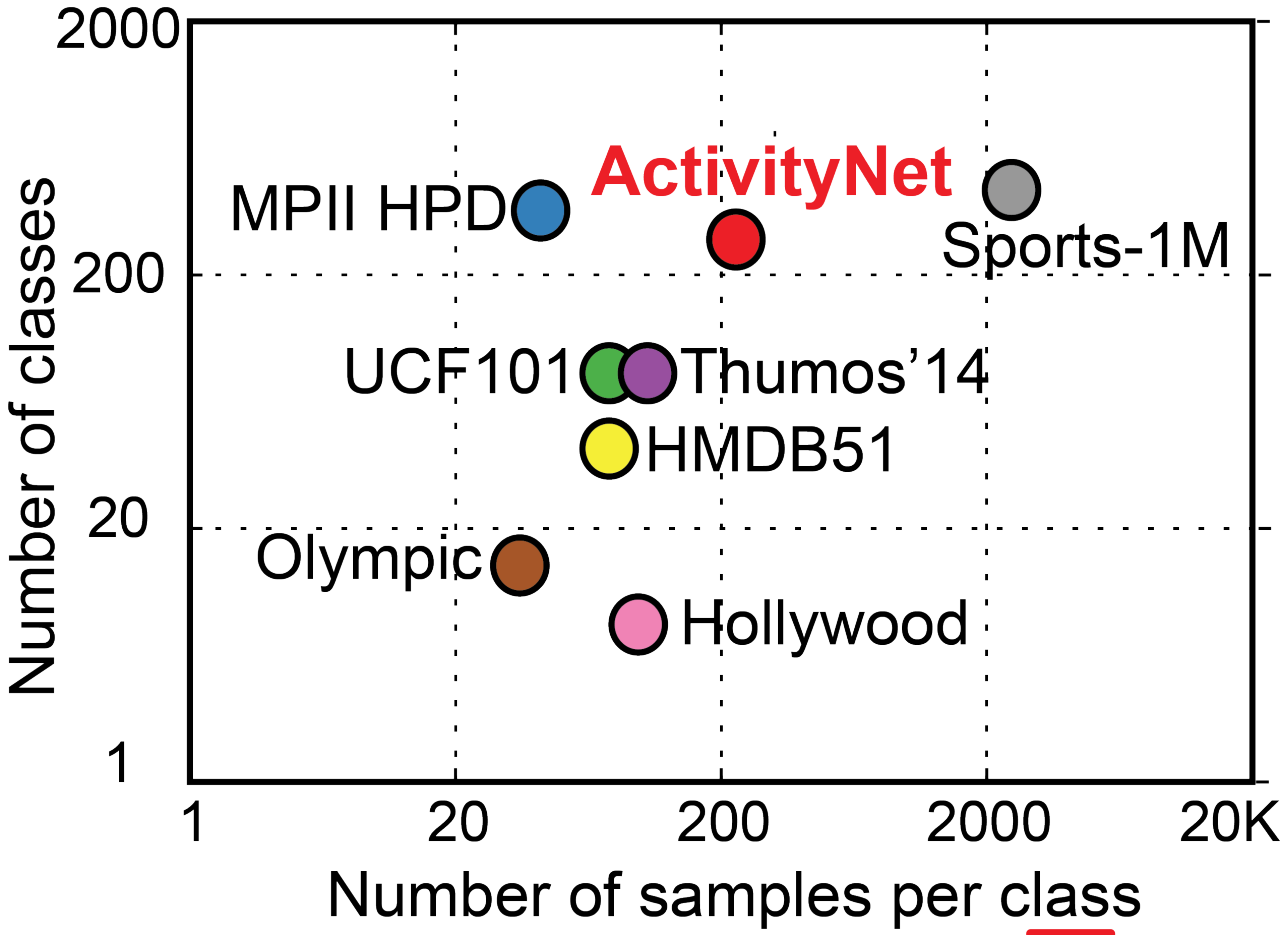

ActivityNet
Eating and Drinking
Education
Socializing
Sports
Work-related
Personal Care
Household Activities
Interior cleaning
Laundry
Housework
Sewing, repairing
Interior arrangement
Interior Maintenance
Build & Rep.
Heating and cooling
Exterior Maintenance
Exterior cleaning
Lawn, Garden
Repair,
houseplant care
Animals and Pets
Care for animals
Exercising with animals
Vehicles
Vehicle repair
Vacuuming floor
Cleaning toilet
Making the bed
Cleaning windows
Cleaning out closet
Polishing furniture
Recycling
Hand washing clothes
Ironing clothes
Sorting laundry
Polishing shoes
Cleaning shoes
Knitting sweaters
Changing light bulbs
Painting
Fixing leaks
Fumigating house
Installing carpet
Put. Christmas lights
Assembling furniture
Painting furniture
Chopping wood
Setting the fire
Shoveling snow
Cleaning chimney
Cleaning garage
Sweeping sidewalk
Fixing broken windows
Fixing mailbox
Gardening
Mowing the lawn
Watering garden
Applying pesticides
Cutting the grass
Picking fruits
Bathing dog
Changing dog water
Walking the dog
Grooming horse
Changing oil
Changing wheel
Assembling bicycle
Installing car stereo
Put. air in tires
Checking tires

Human Activity Benchmark

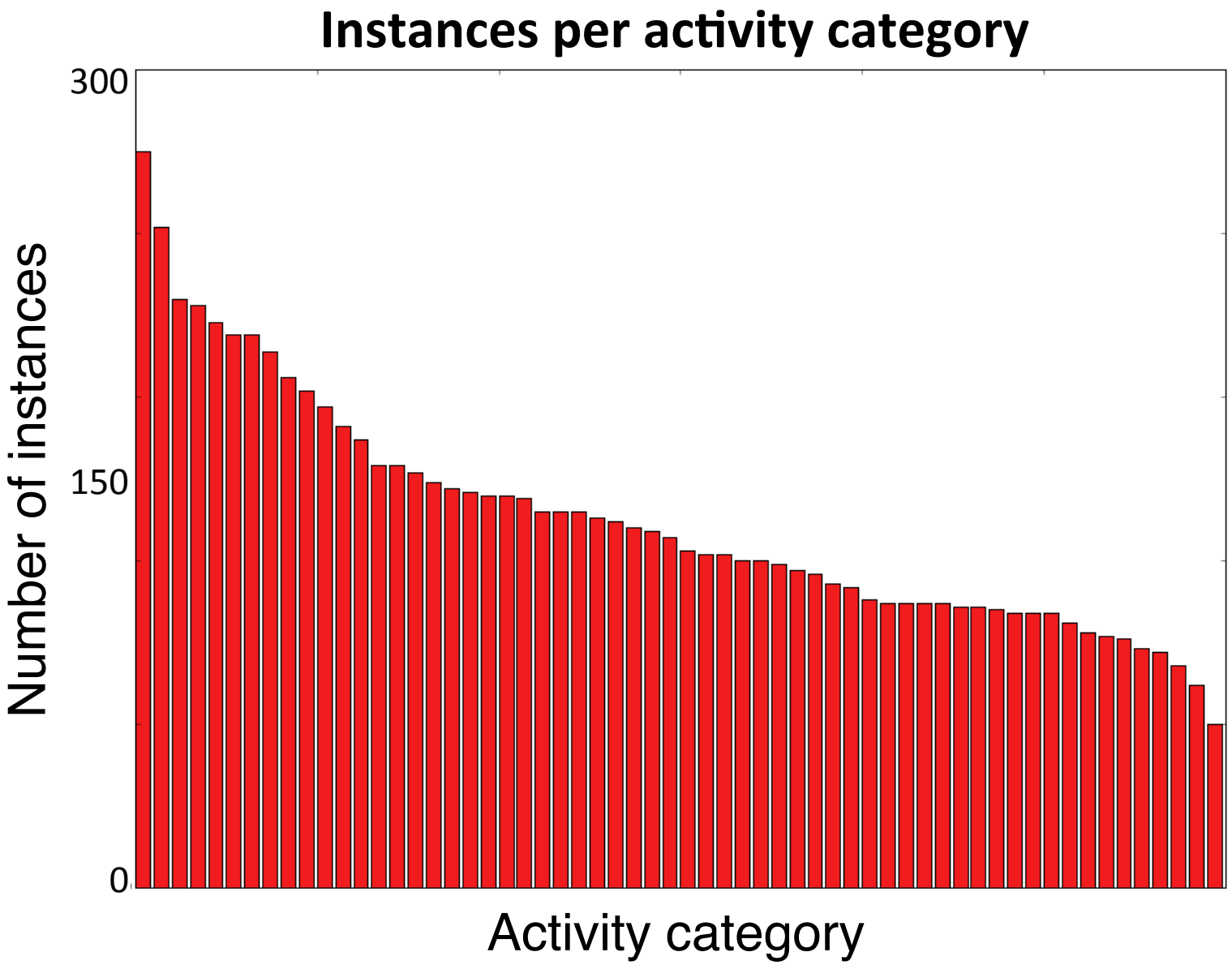


ACTIVITYNET

ActivityNet – A Large scale benchmark

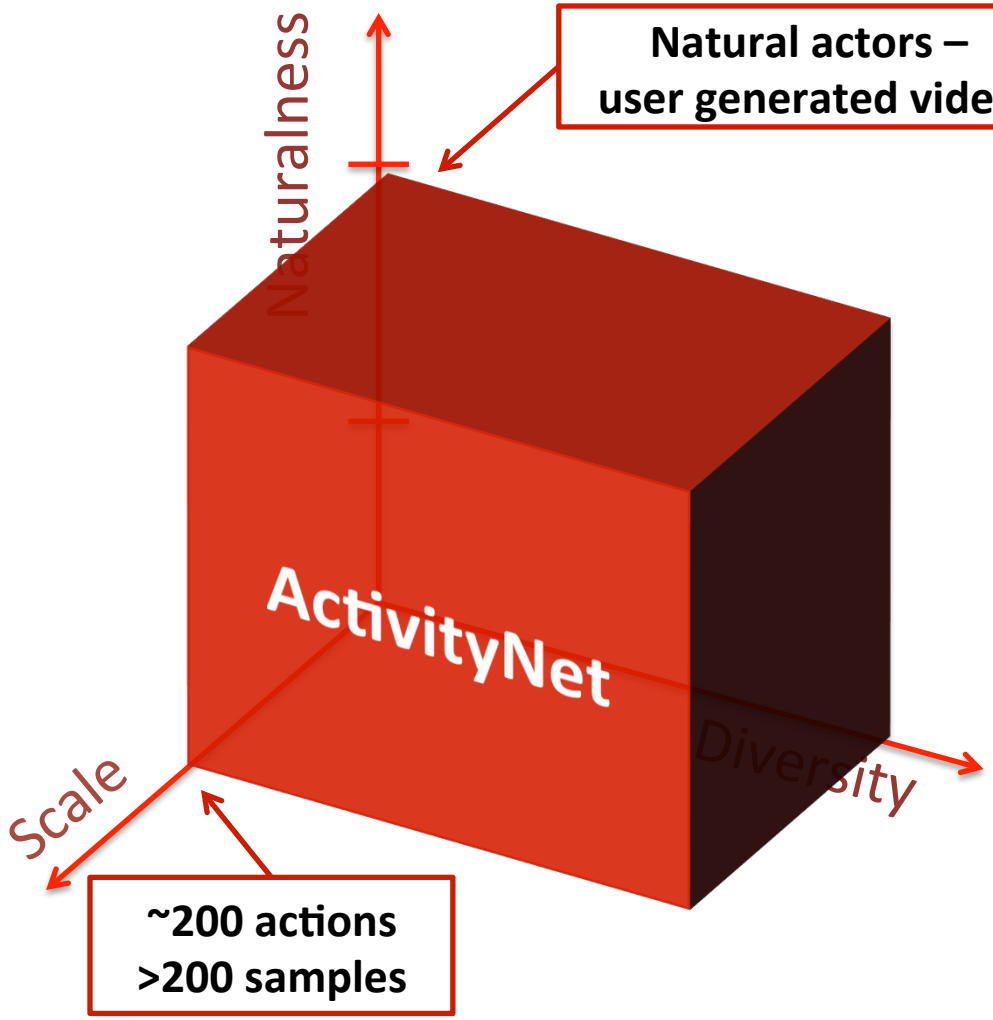


ACTIVITYNET



ACTIVITYNET

ActivityNet



200
Activity Categories

100
Untrimmed Videos per Class

1.65
Activity Instances per Video

600
Storage (GB)

ACTIVITYNET

Challenge Data Statistics

- 200 activity categories
- 20K videos, 32K activity instances

ACTIVITYNET

Challenge Tasks

- Task I: Untrimmed Video Classification

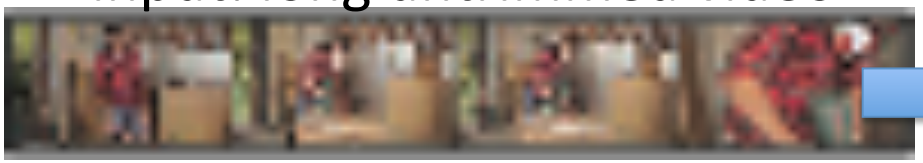
input: long untrimmed video

output

activity presence (binary)

- Task II: Activity Detection

input: long untrimmed video



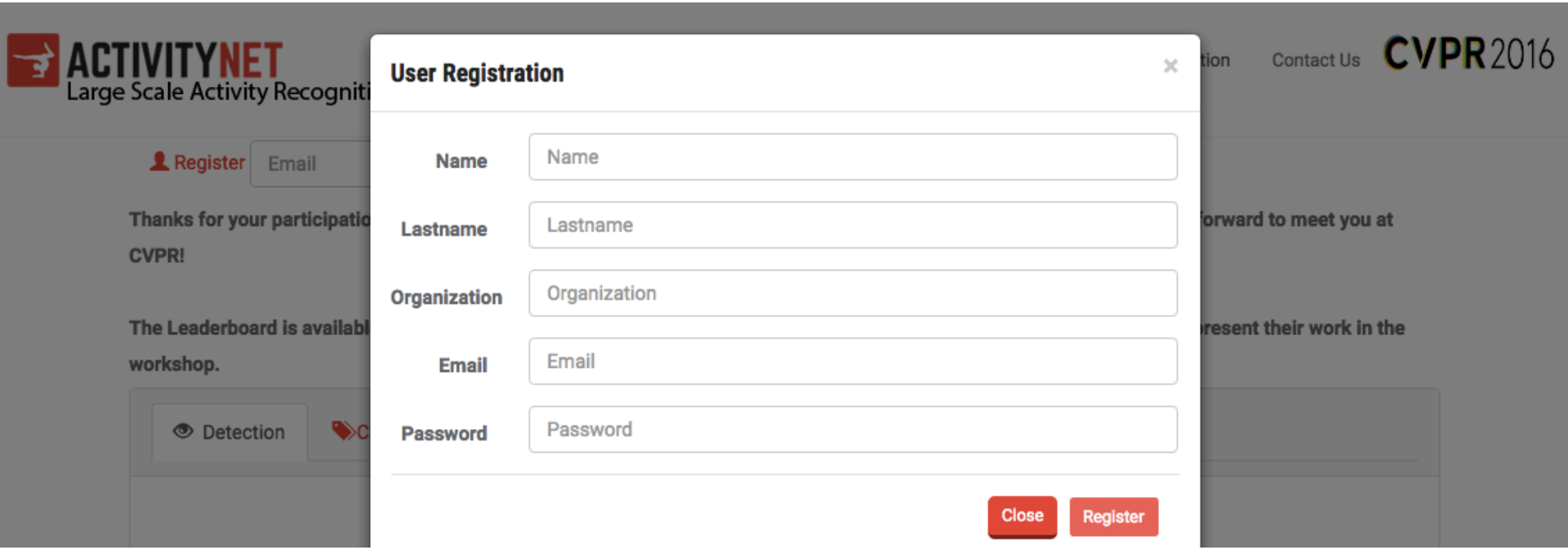
output

activity temporal location

ACTIVITYNET

Submission process

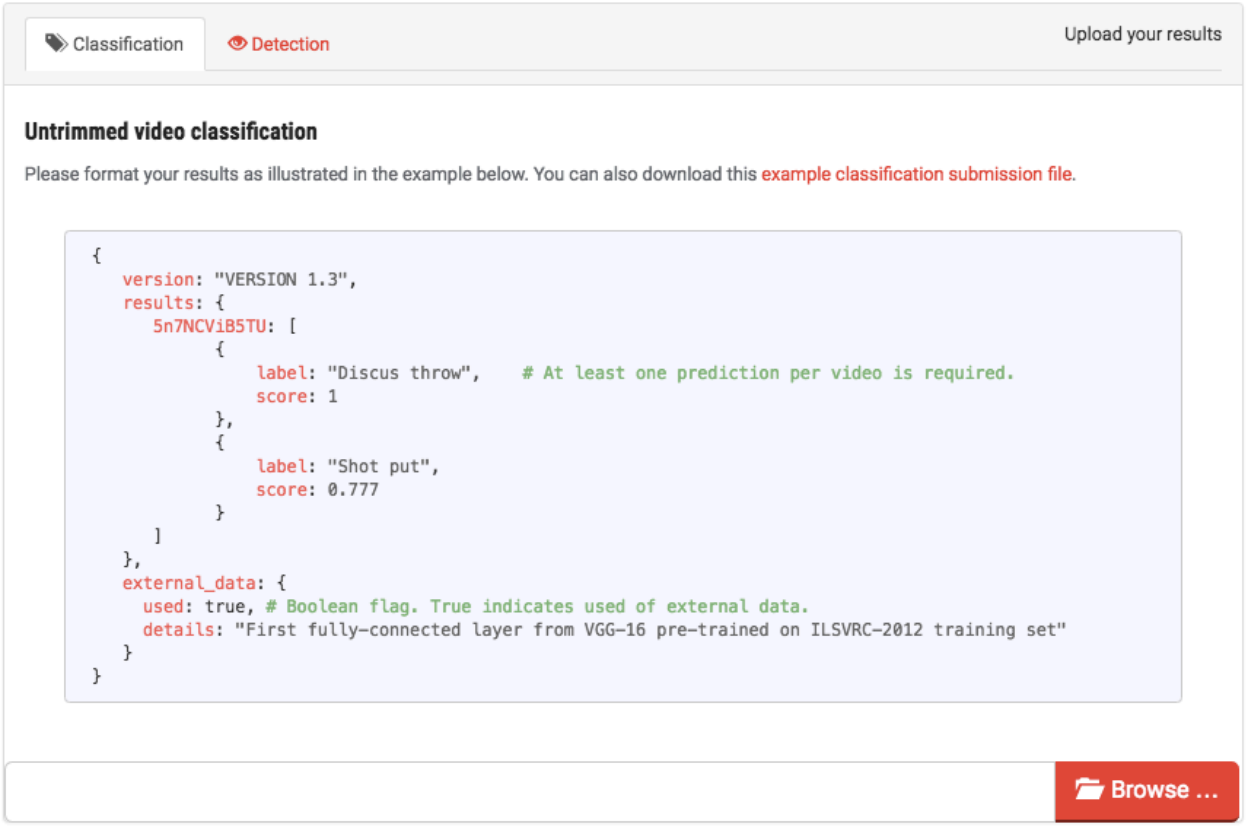
- 1. Register to the ActivityNet evaluation server



ACTIVITYNET

Submission process

- 2. Format and submit your results



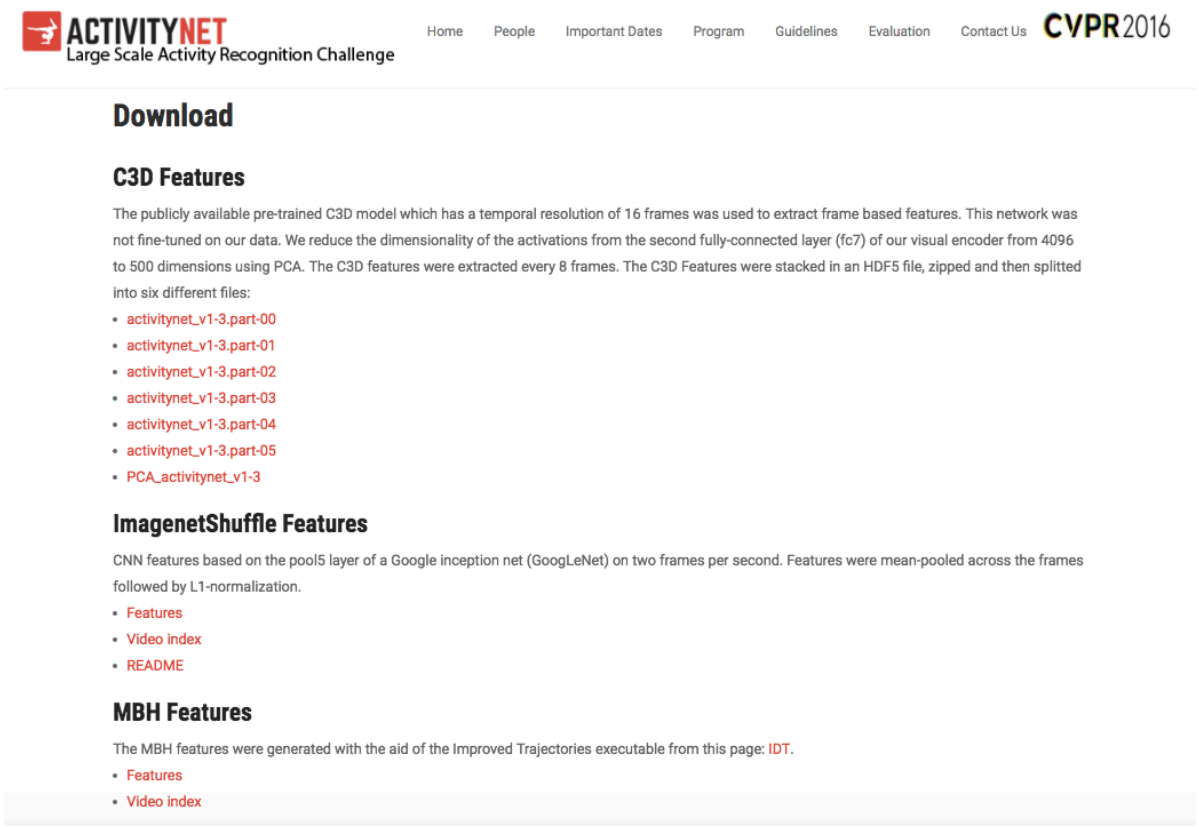
ACTIVITYNET

Policies

- Only one submission per week per team
- Generate results on the testing set by analyzing audio-visual content only
- Not use the test set for training or parameter tuning
- The use of external data is allowed

Provided tools

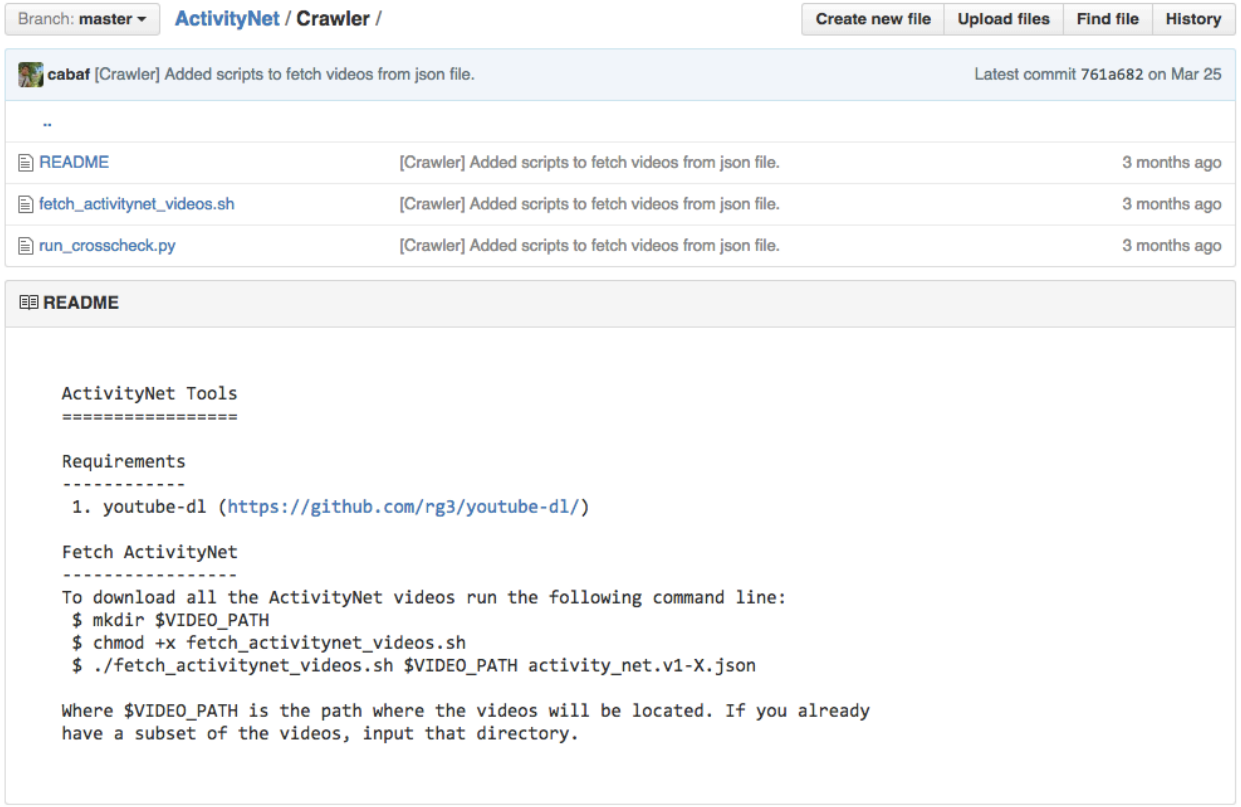
- 1. Features



ACTIVITYNET

Provided tools

- 2. Scripts to download videos



Branch: master ▾ ActivityNet / Crawler / Create new file | Upload files | Find file | History

cabaf [Crawler] Added scripts to fetch videos from json file. Latest commit 761a682 on Mar 25

..

README	[Crawler] Added scripts to fetch videos from json file.	3 months ago
fetch_activitynet_videos.sh	[Crawler] Added scripts to fetch videos from json file.	3 months ago
run_crosscheck.py	[Crawler] Added scripts to fetch videos from json file.	3 months ago

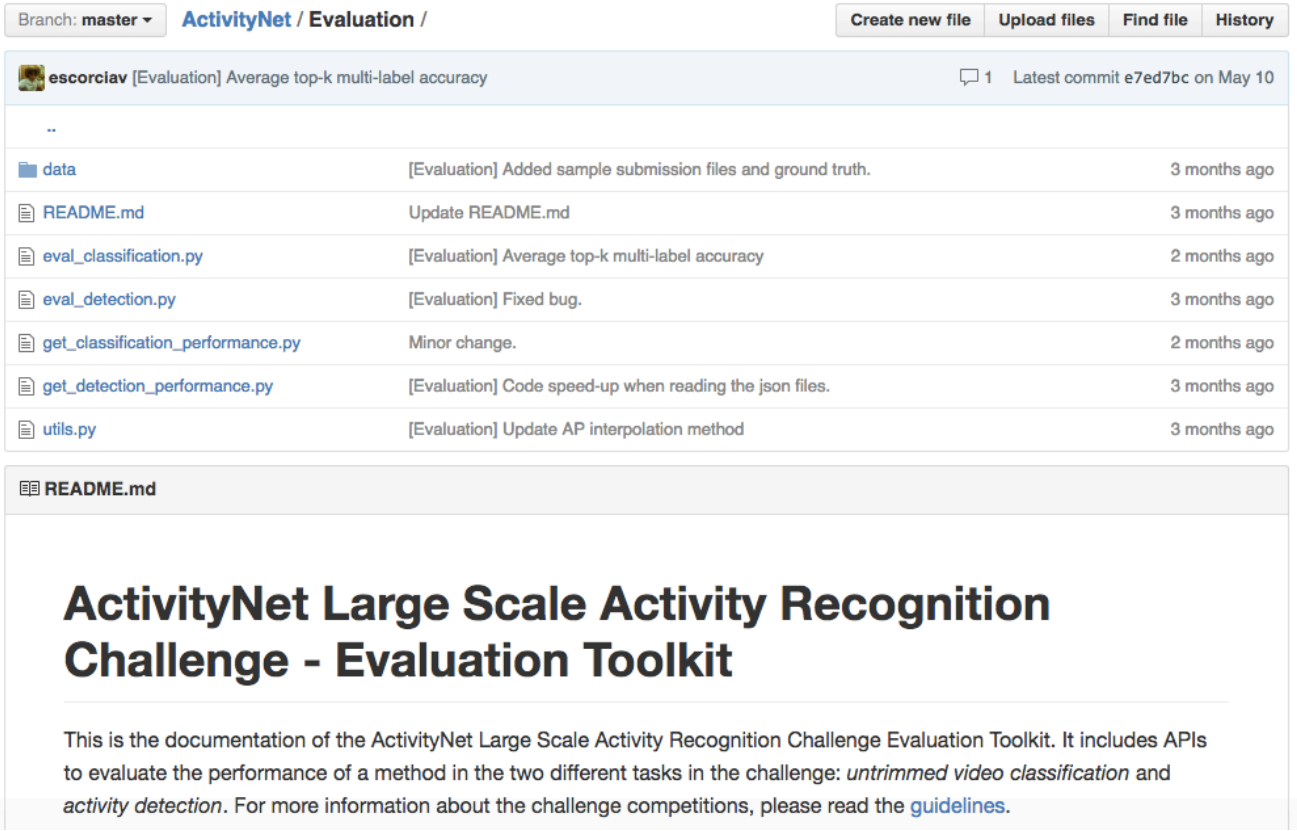
README

```
ActivityNet Tools
=================

Requirements
------------
 1. youtube-dl (https://github.com/rg3/youtube-dl/)

Fetch ActivityNet
-----------------
To download all the ActivityNet videos run the following command line:
 $ mkdir $VIDEO_PATH
 $ chmod +x fetch_activitynet_videos.sh
 $ ./fetch_activitynet_videos.sh $VIDEO_PATH activity_net.v1-X.json

Where $VIDEO_PATH is the path where the videos will be located. If you already
have a subset of the videos, input that directory.
```


ACTIVITYNET

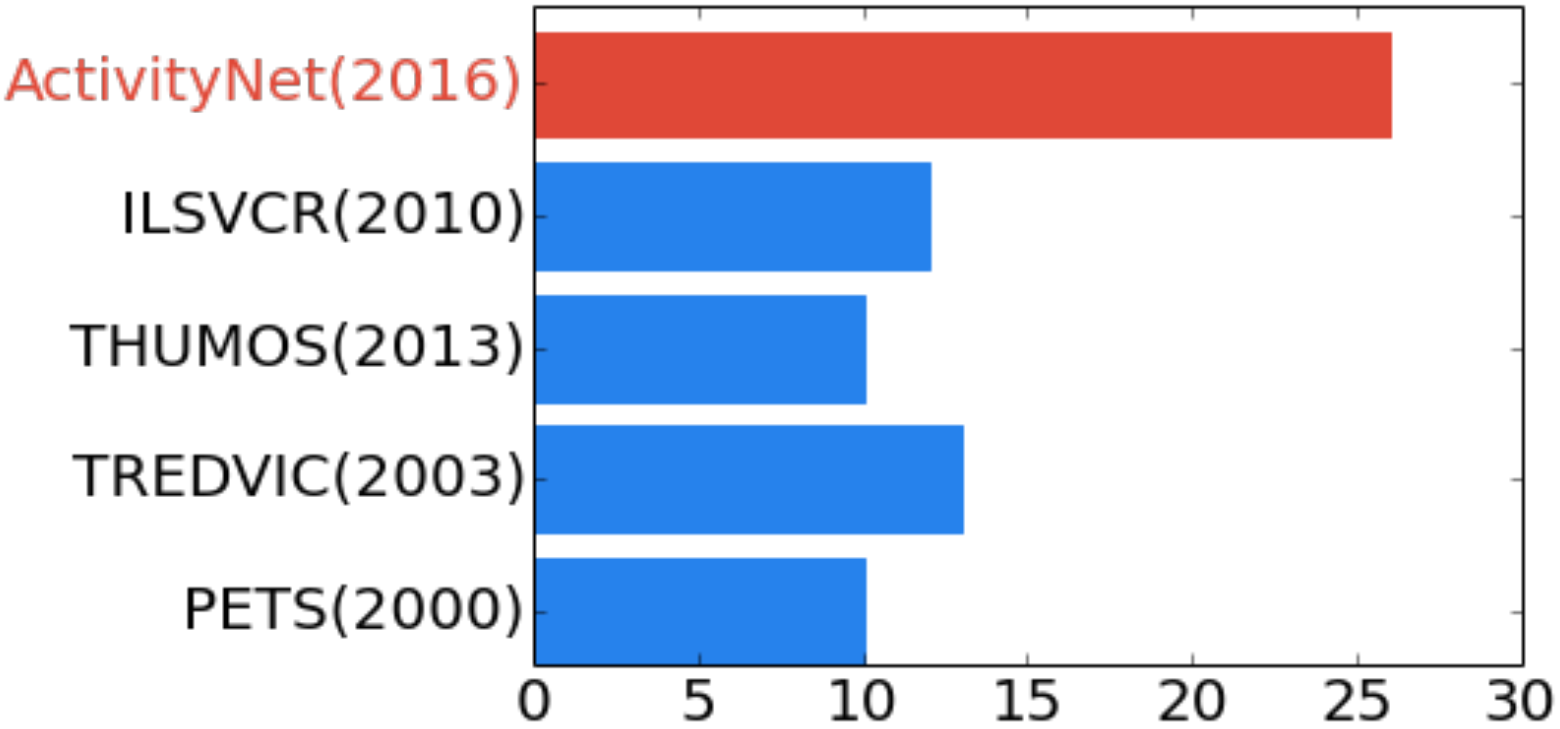
Provided tools

- 3. Evaluation code



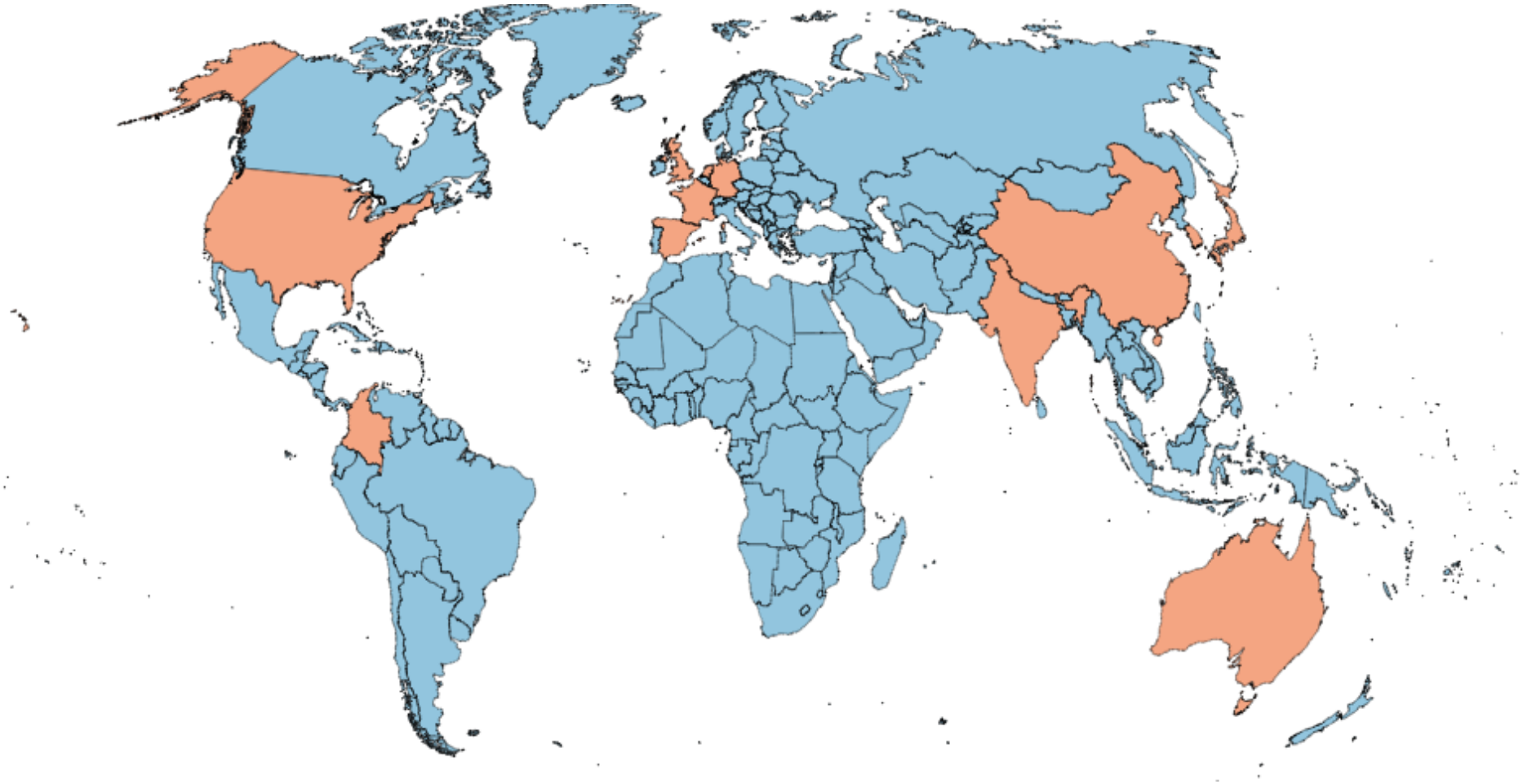
ACTIVITYNET

of participants in vision challenges



ACTIVITYNET

Participants from 14 countries



ACTIVITYNET

Classification results

Task I: Untrimmed Classification

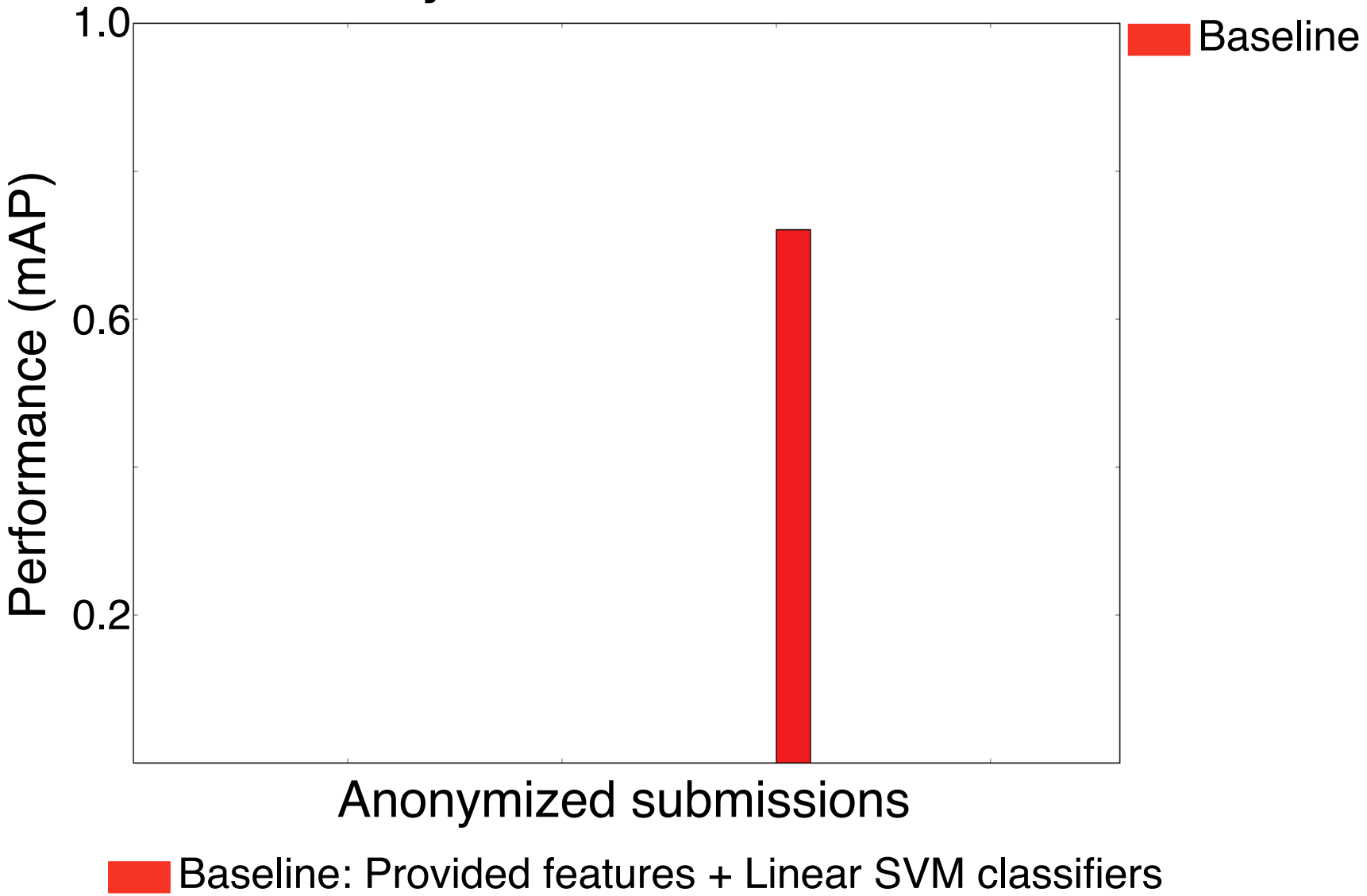
Activity: **Polishing shoes**

ACTIVITYNET

Task II: Activity detection

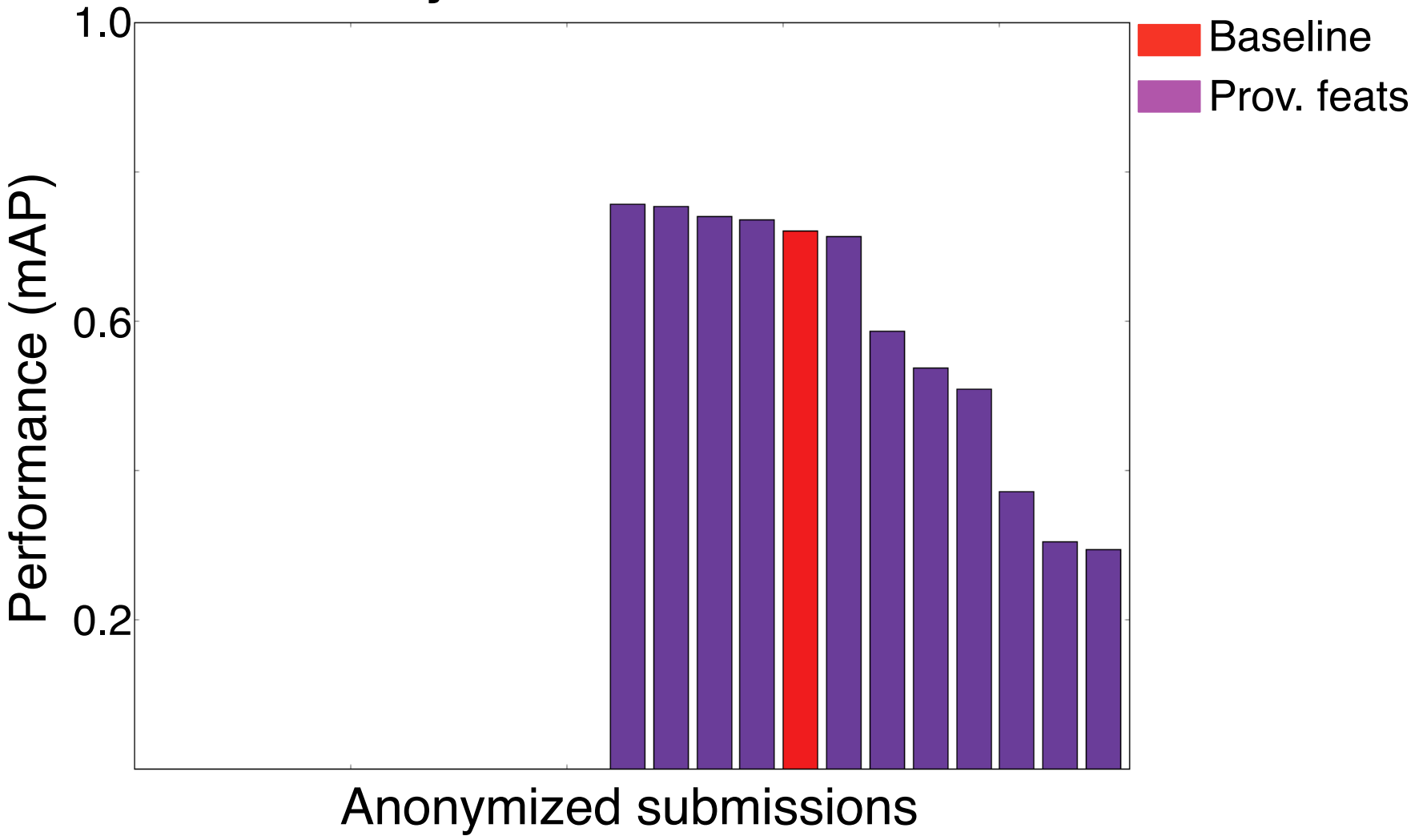
- Metrics:
  - mean Average Precision (determines winners)
  - Top-1 accuracy
  - Top-3 accuracy

Summary of classification results



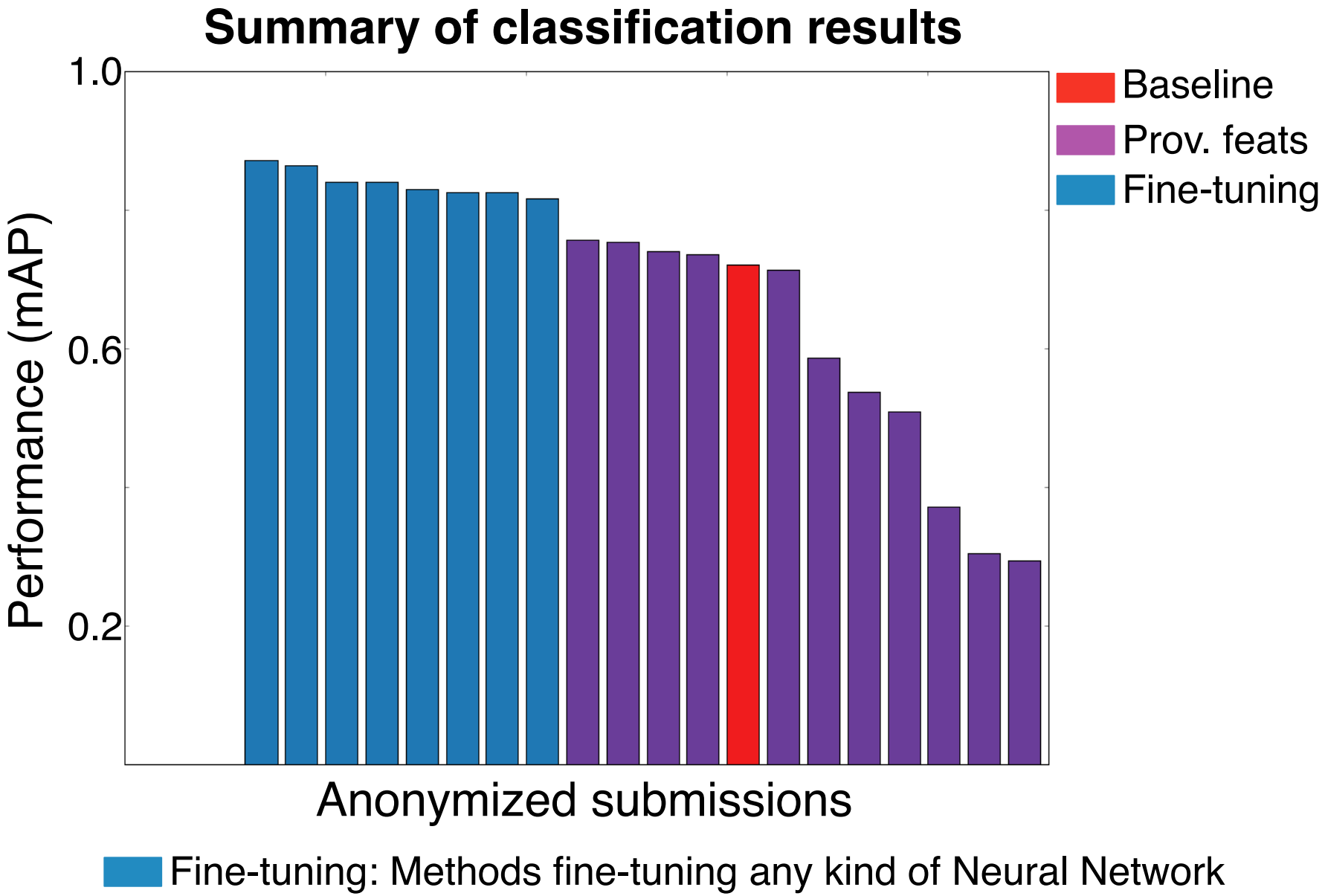
ACTIVITYNET

Summary of classification results



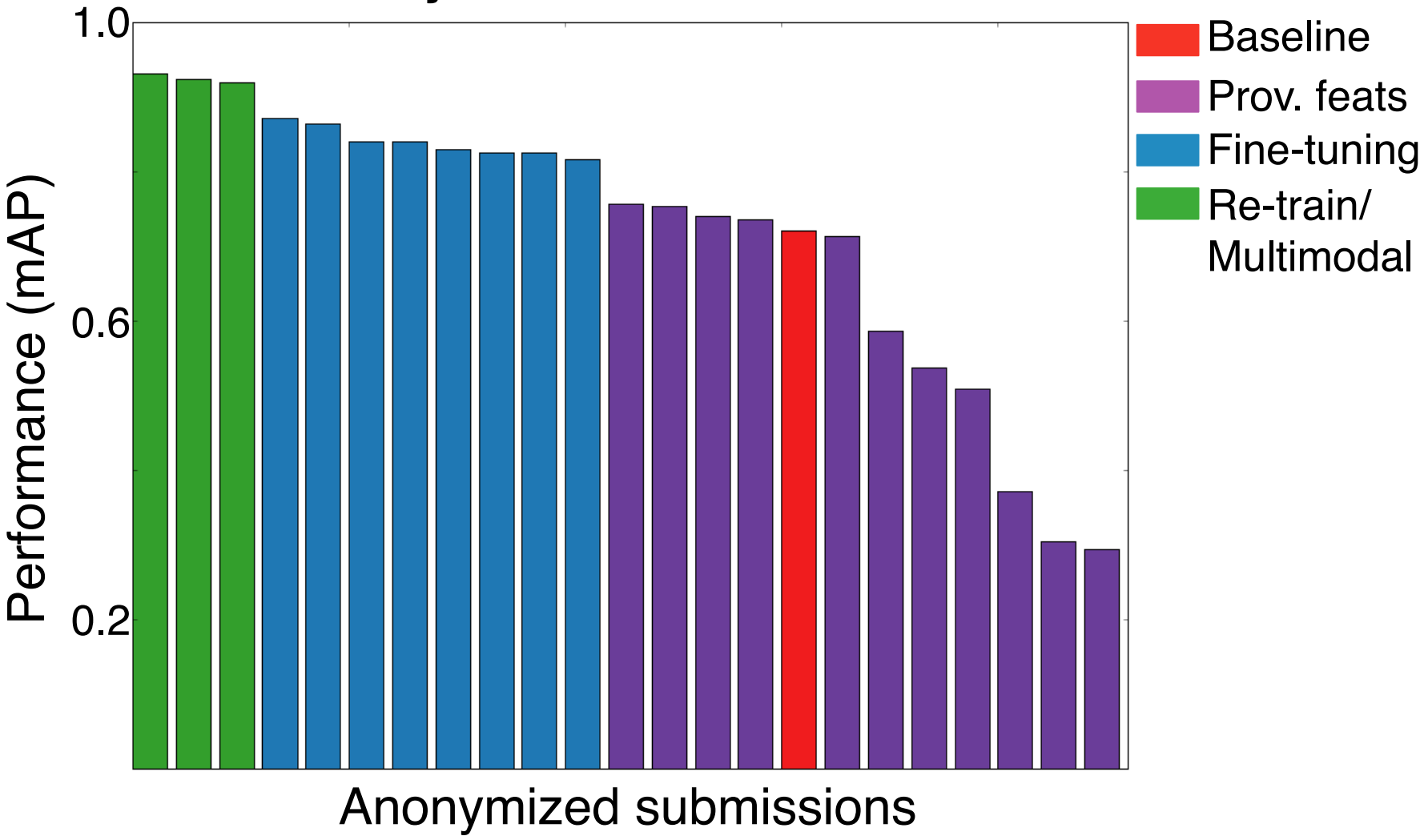
Prov. feats: Methods using only the provided features

ACTIVITYNET



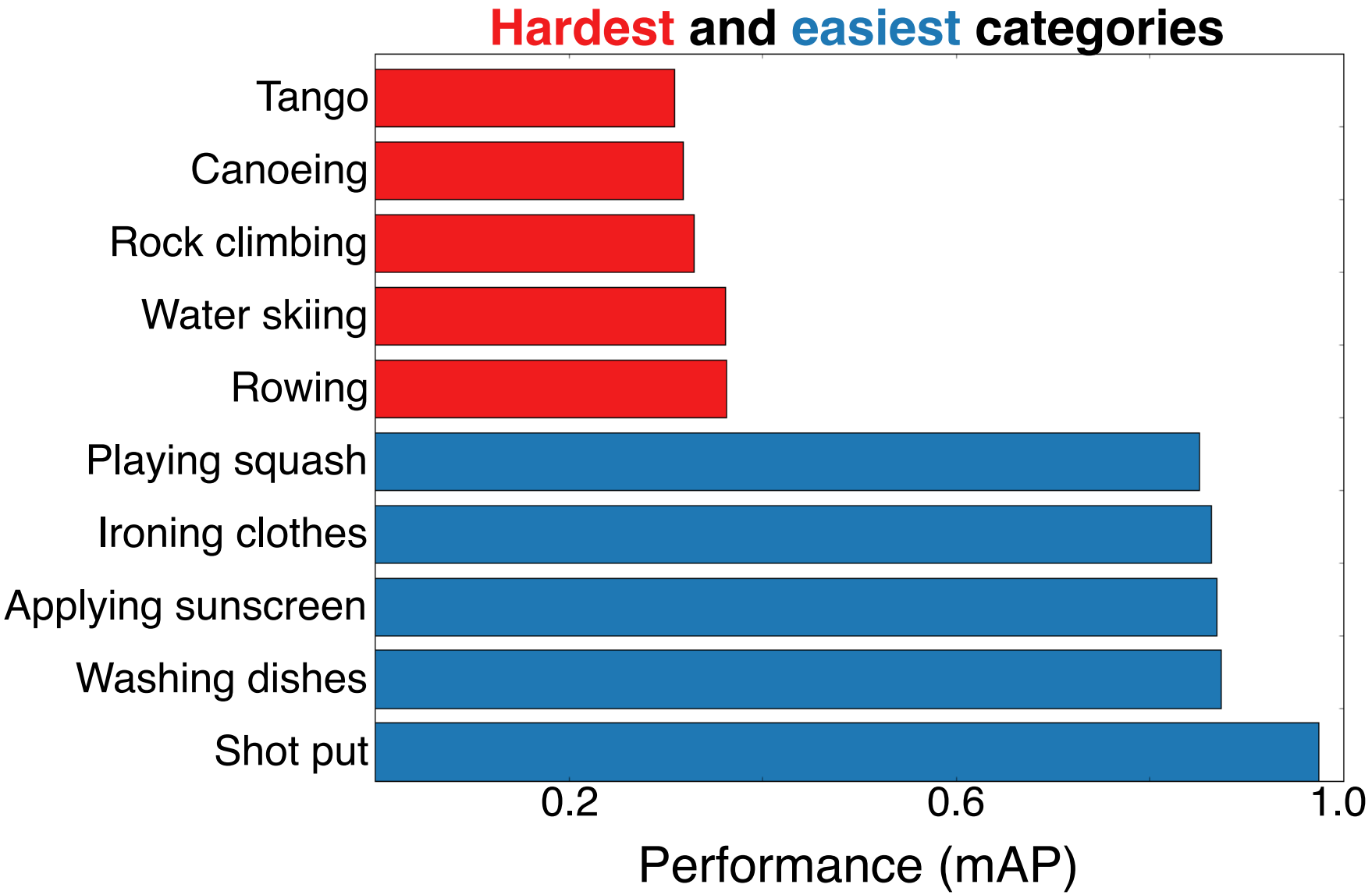
ACTIVITYNET

Summary of classification results



Re-train/Multimodal: Methods re-training a Neural Network or using multimodal representation

ACTIVITYNET



ACTIVITYNET

Qualitative results

- Videos where the methods do well

Washing dishes

Ironing clothes

Shotput

ACTIVITYNET

Qualitative results

- Videos where the methods fail

Water skiing

Rock climbing

Applying sunscreen

ACTIVITYNET

Rank	Organization	mAP	Top-1
10	Oxford Brookes	82.5	76.7

10. Oxford Brookes

- Uses features provided by the challenge organizers (IDT, GoogleLeNet, C3D) to learn one-vs-all SVM classifiers
- A meta-classifier is used to fuse all features

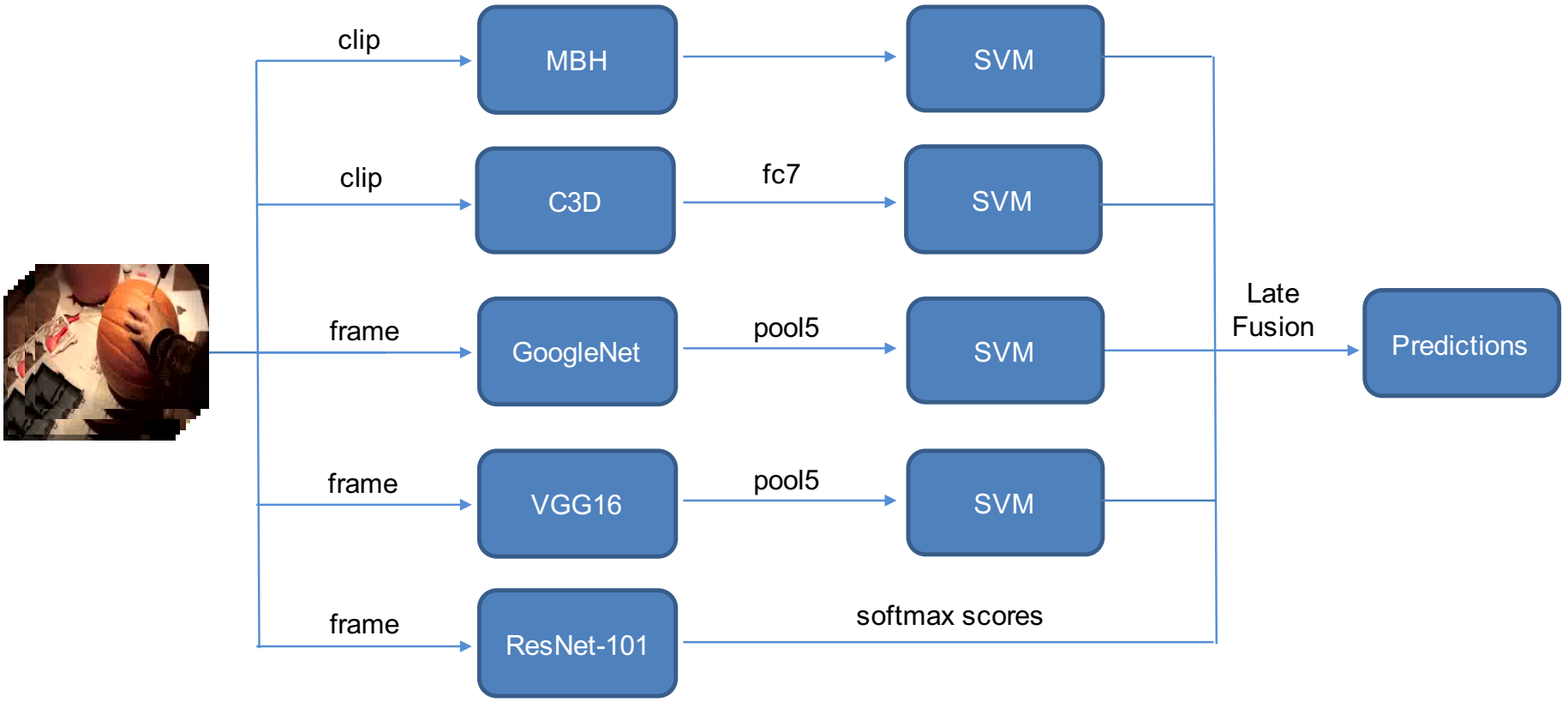
Rank	Organization	mAP	Top-1
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

9. Xerox Research

- Data augmentation
- IDTs and Audio features to represent videos
- SVM classifiers are used to learn the action models

Rank	Organization	mAP	Top-1
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

8. UC Merced

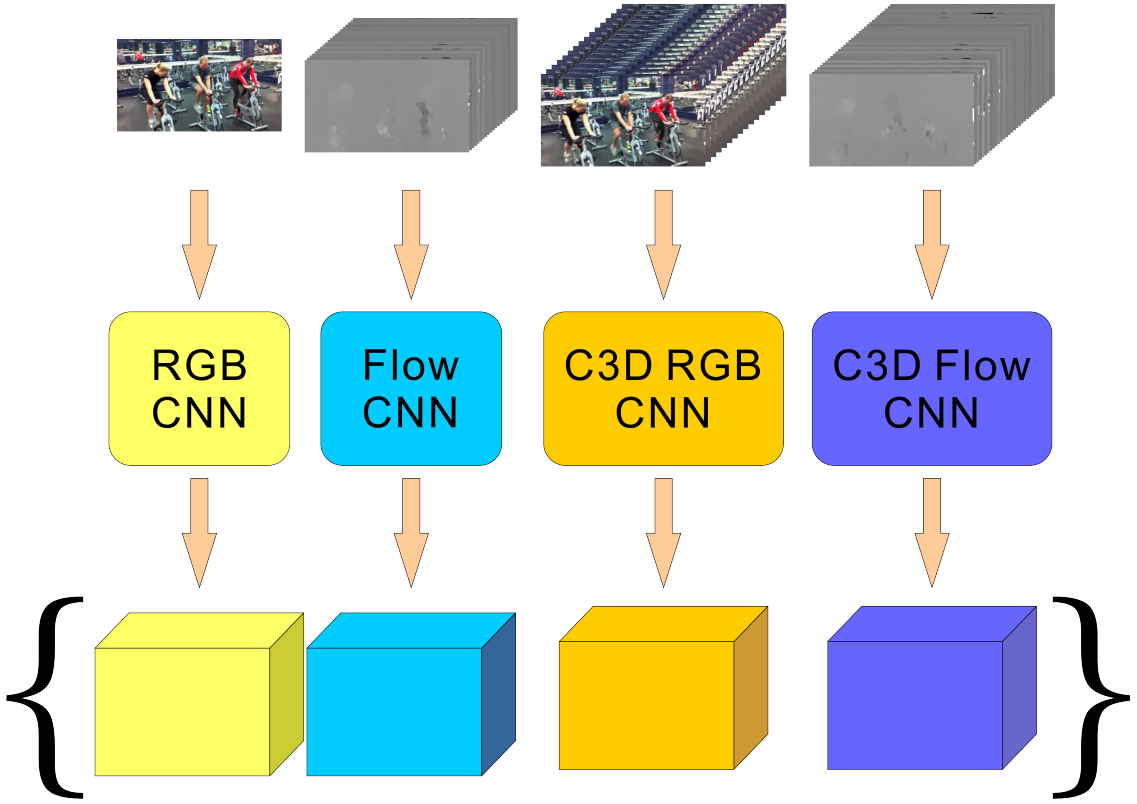


ACTIVITYNET

Rank	Organization	mAP	Top-1
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

Rank	Organization	mAP	Top-1
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

6. Zhejiang University



ACTIVITYNET

Rank	Organization	mAP	Top-1
5	University of Tokyo	86.4	80.4
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

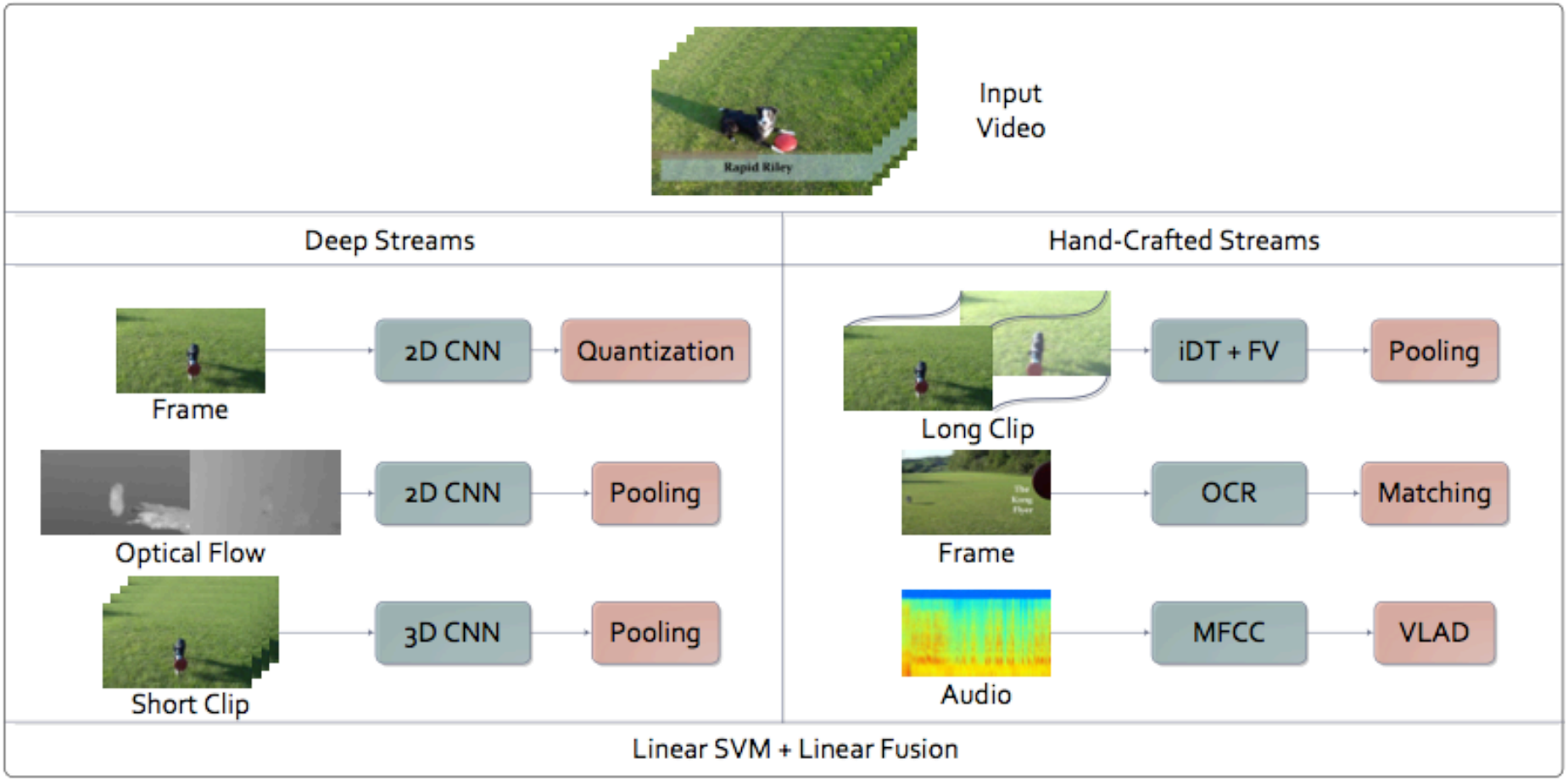
5. University of Tokyo

- Two stream network
- ResNet fine-tuned in ActivityNet
- Only action segments were used in training

Rank	Organization	mAP	Top-1
4	UTS II*	87.1	84.9
5	University of Tokyo	86.4	80.4
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

Rank	Organization	mAP	Top-1
3	MSRA	91.9	86.6
4	UTS II*	87.1	84.9
5	University of Tokyo	86.4	80.4
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

3. MSRA



ACTIVITYNET

Rank	Organization	mAP	Top-1
2	UTS	92.4	87.7
3	MSRA	91.9	86.6
4	UTS II*	87.1	84.9
5	University of Tokyo	86.4	80.4
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

2nd Place – Untrimmed Video Classification Task

- TEAM: UTS
- Prize: 1 Commemorative Plaque

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

ACTIVITYNET

Rank	Organization	mAP	Top-1
1	CUHK&ETHZ&SIAT	93.2	88.1
2	UTS	92.4	87.7
3	MSRA	91.9	86.6
4	UTS II*	87.1	84.9
5	University of Tokyo	86.4	80.4
6	Zhejiang University	84.1	83.3
7	USTC*	84.0	79.6
8	UC Merced	83.1	78.4
9	Xerox Research	82.6	78.5
10	Oxford Brookes	82.5	76.7

1st Place – Untrimmed Video Classification Task

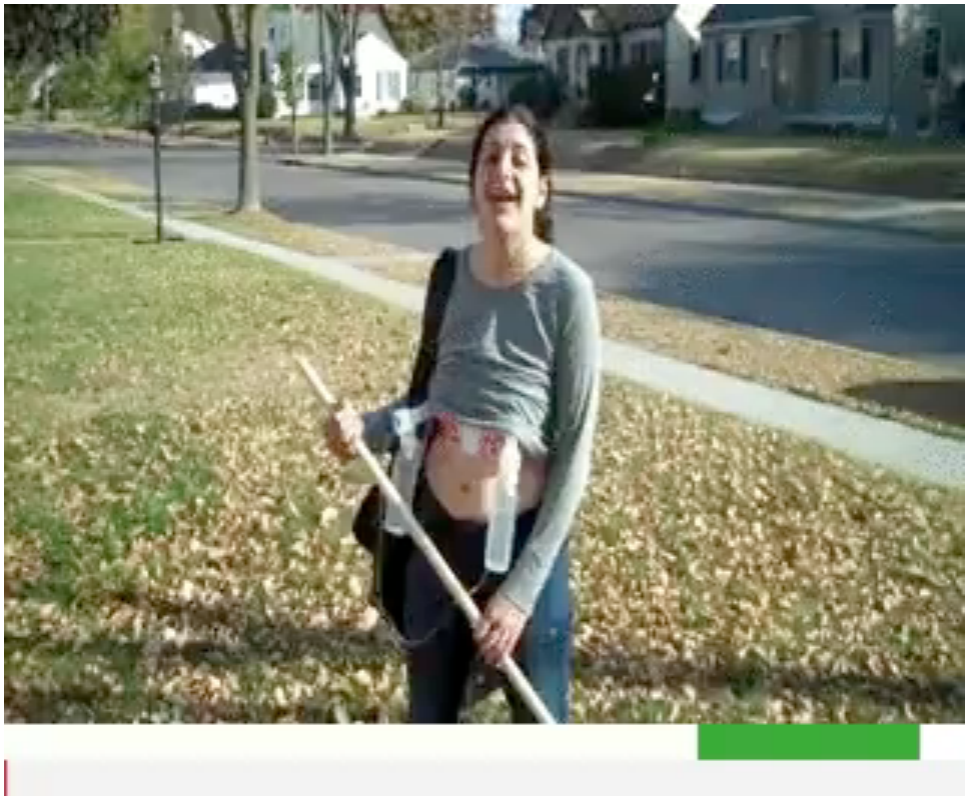
- TEAM: CUHK & ETHZ & SIAT
- Prizes: 1 GTX TITAN X, 1 Commemorative Plaque

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

ACTIVITYNET

Detection Results

Task II: Activity detection



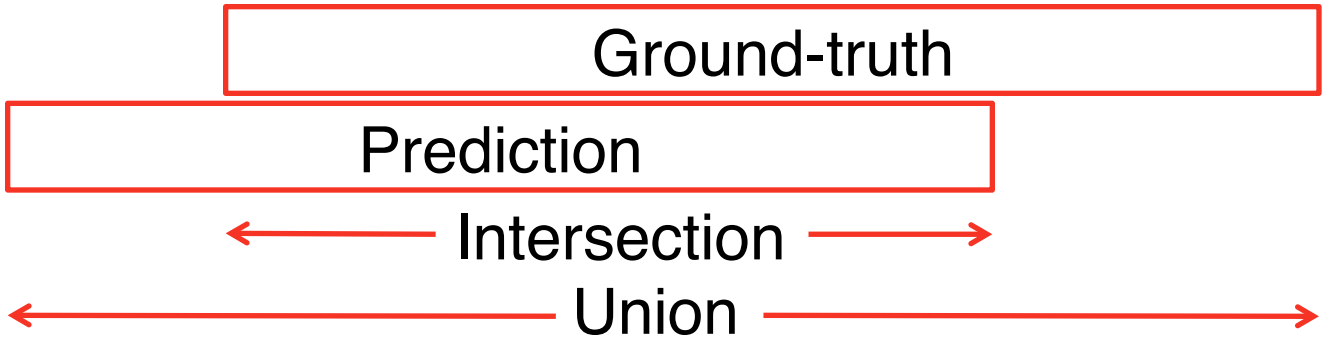
Activity: **Raking Leaves**

Time progress

ACTIVITYNET

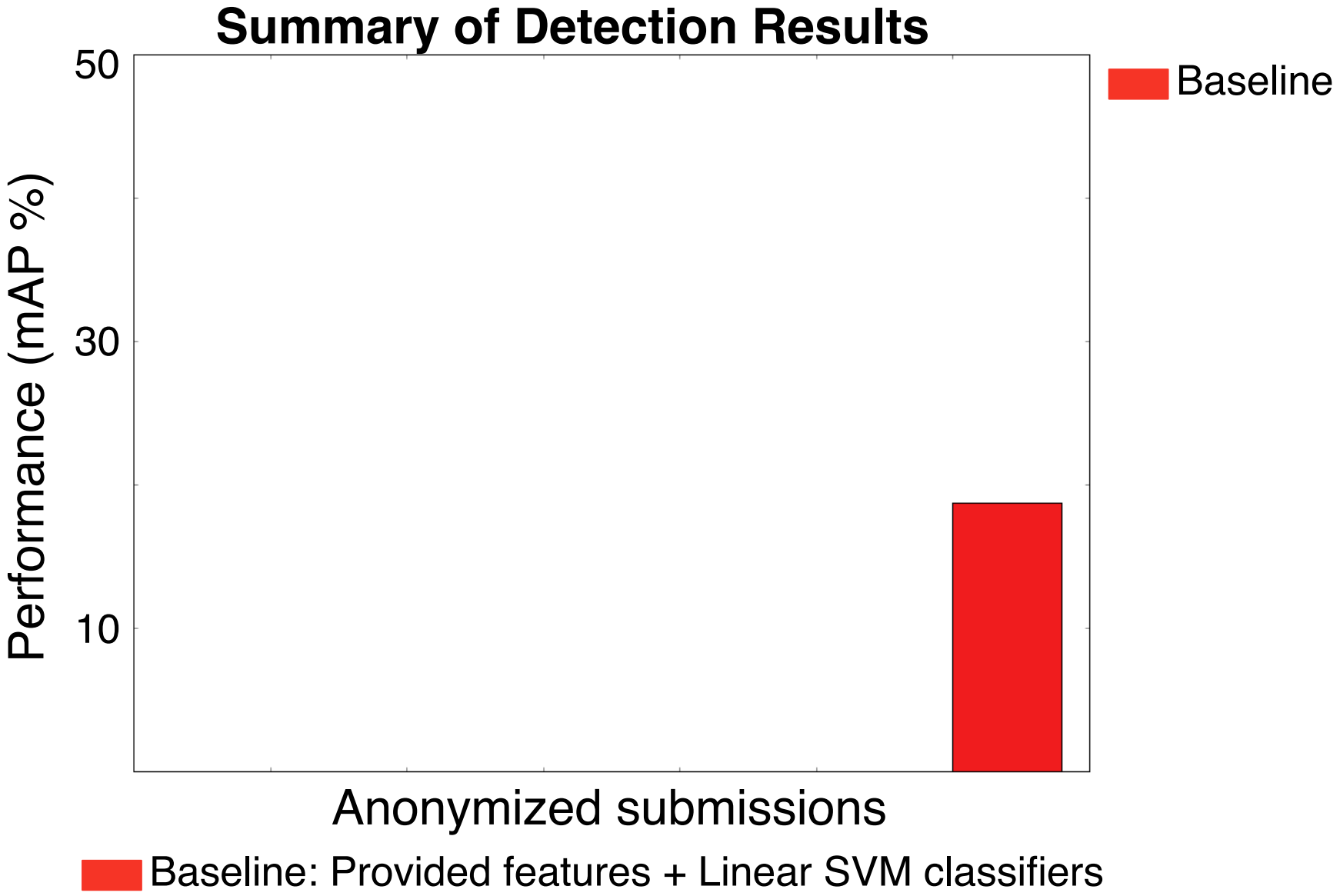
Task II: Activity detection

- Metric: temporal mean Average Precision
  - temporal Intersection over Union (tIoU)

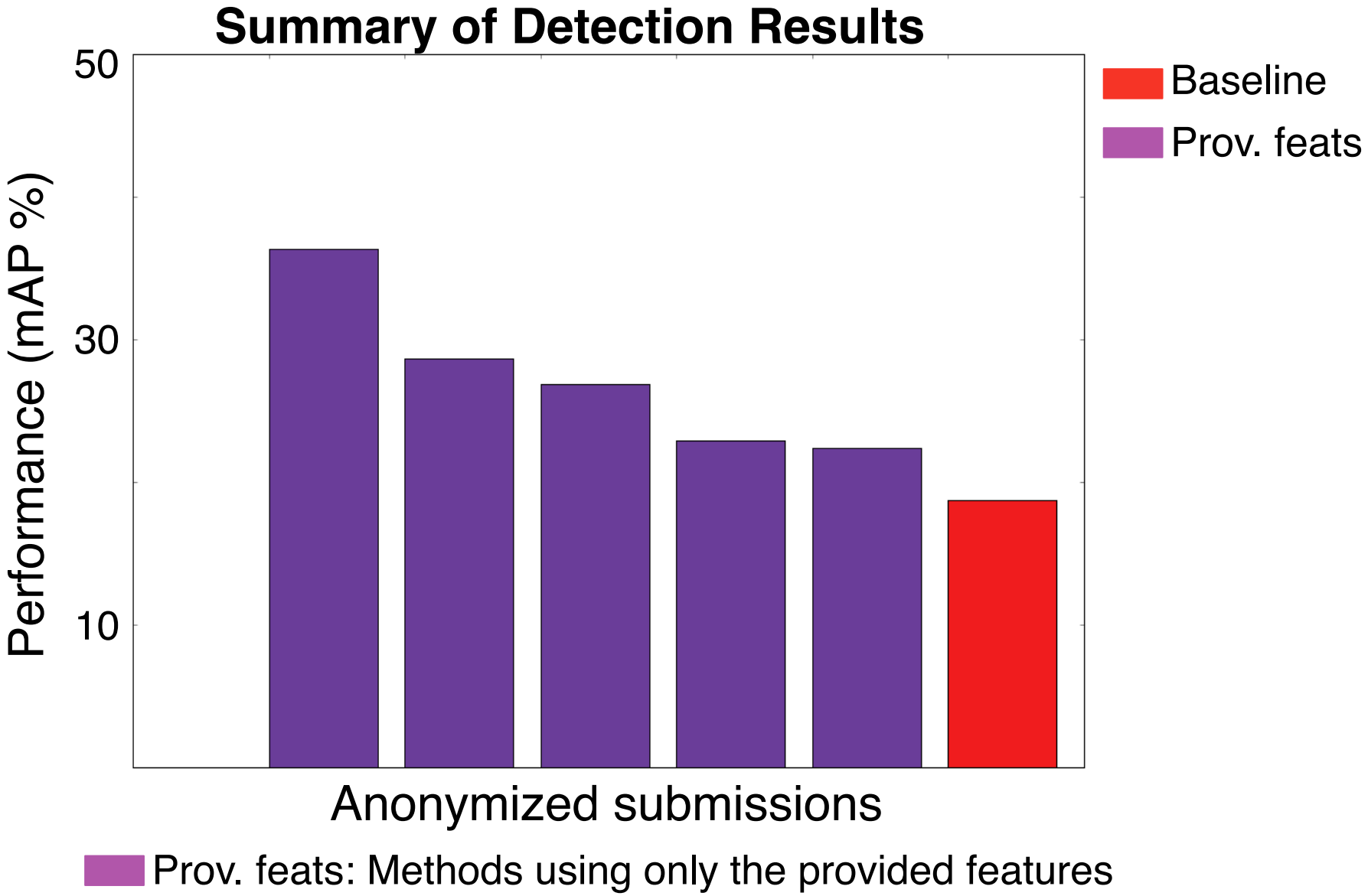


  - Predictions with tIoU>0.5 are marked as true positive
  - Only one prediction must match the ground-truth

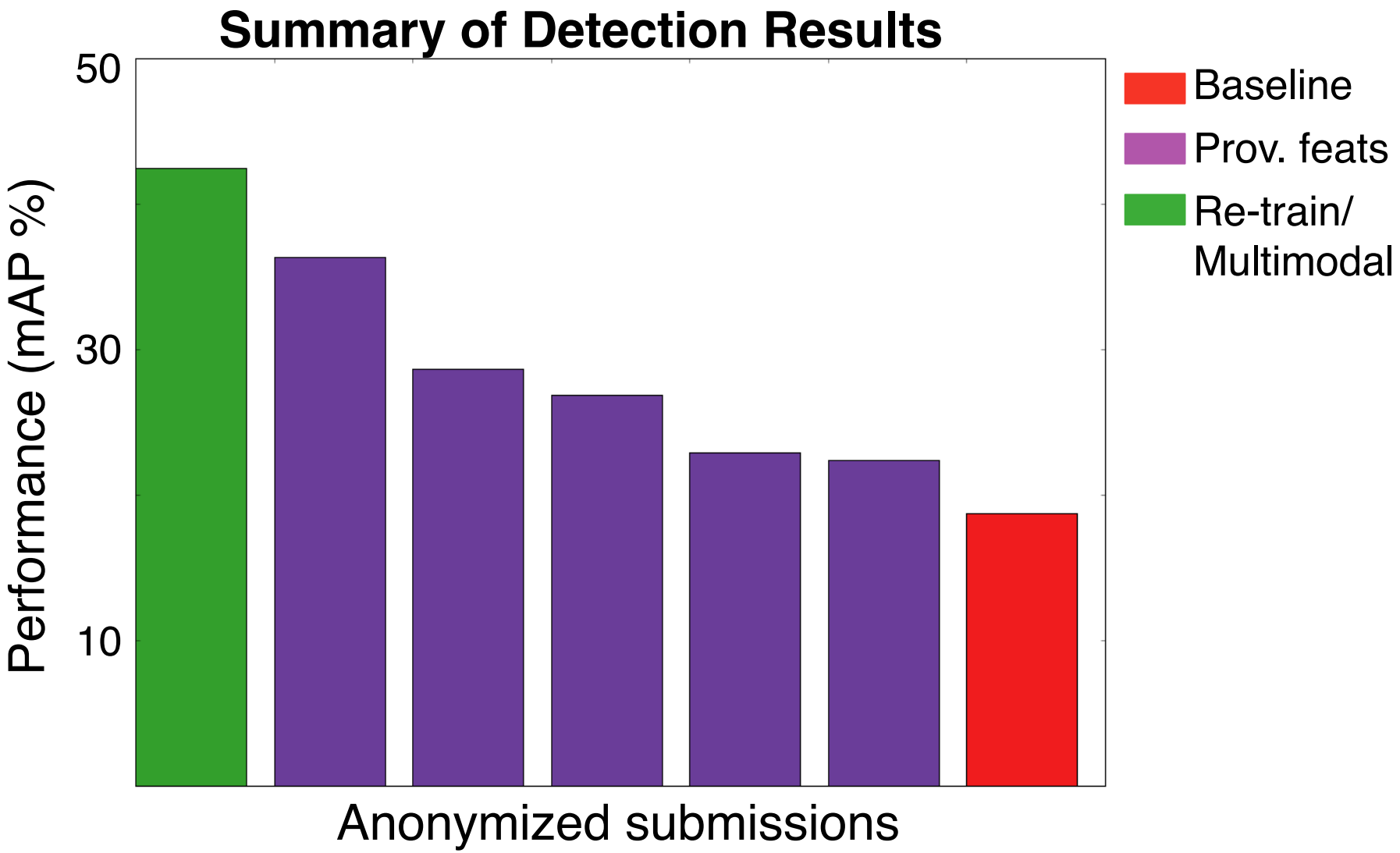
ACTIVITYNET



ACTIVITYNET



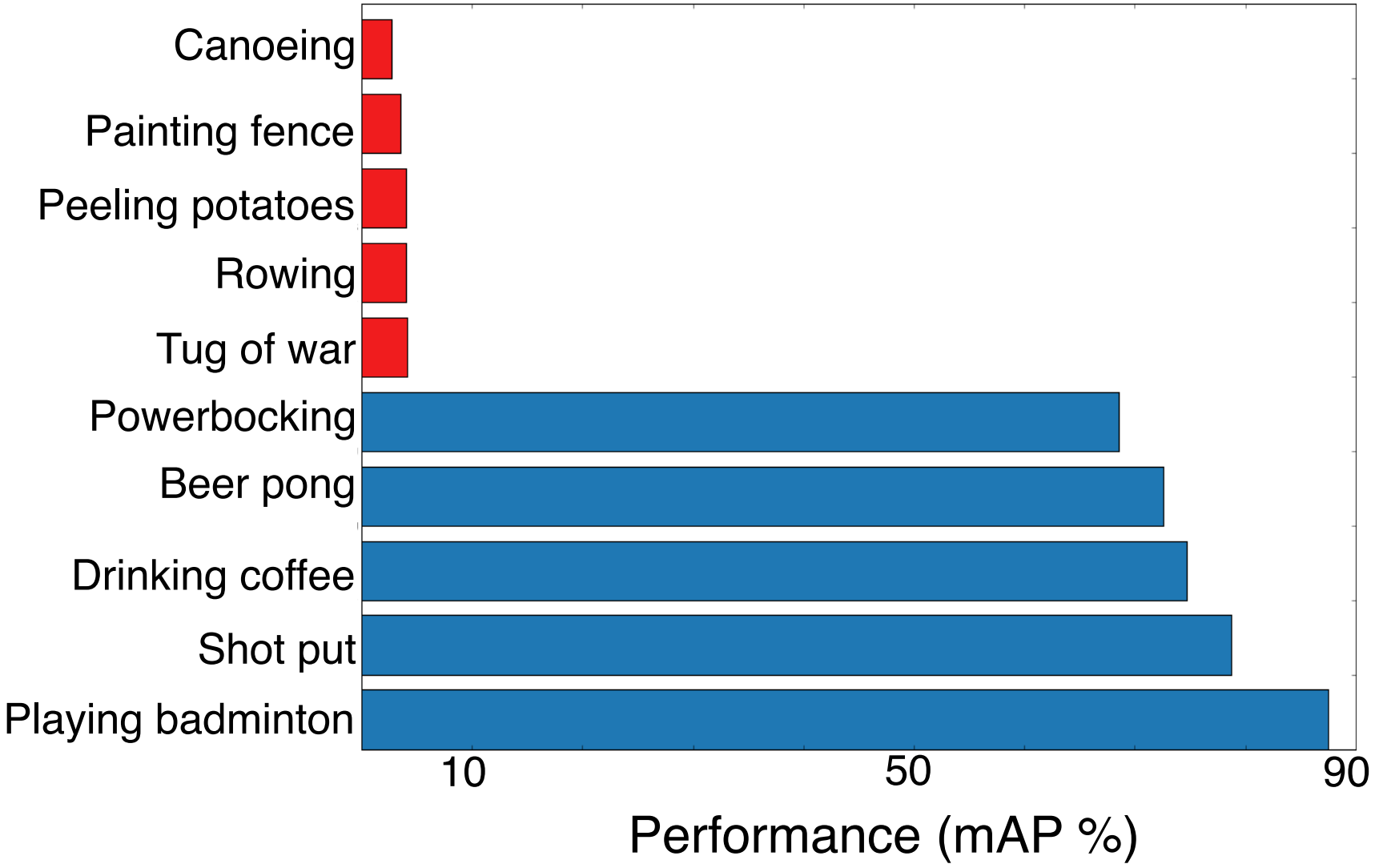
ACTIVITYNET



Re-train/Multimodal: Methods re-training a Neural Network or using multimodal representation

ACTIVITYNET

Hardest and easiest categories



ACTIVITYNET

Qualitative Results

- Videos where all methods do well

Ground-truth Time progress

Badminton

Shotput

Drinking coffee

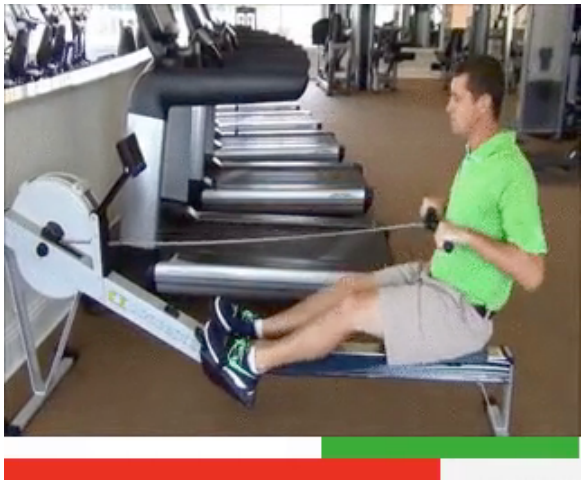
ACTIVITYNET

Qualitative Results

- Videos where all methods fail

Ground-truth Time progress

Peeling potatoes



Rowing

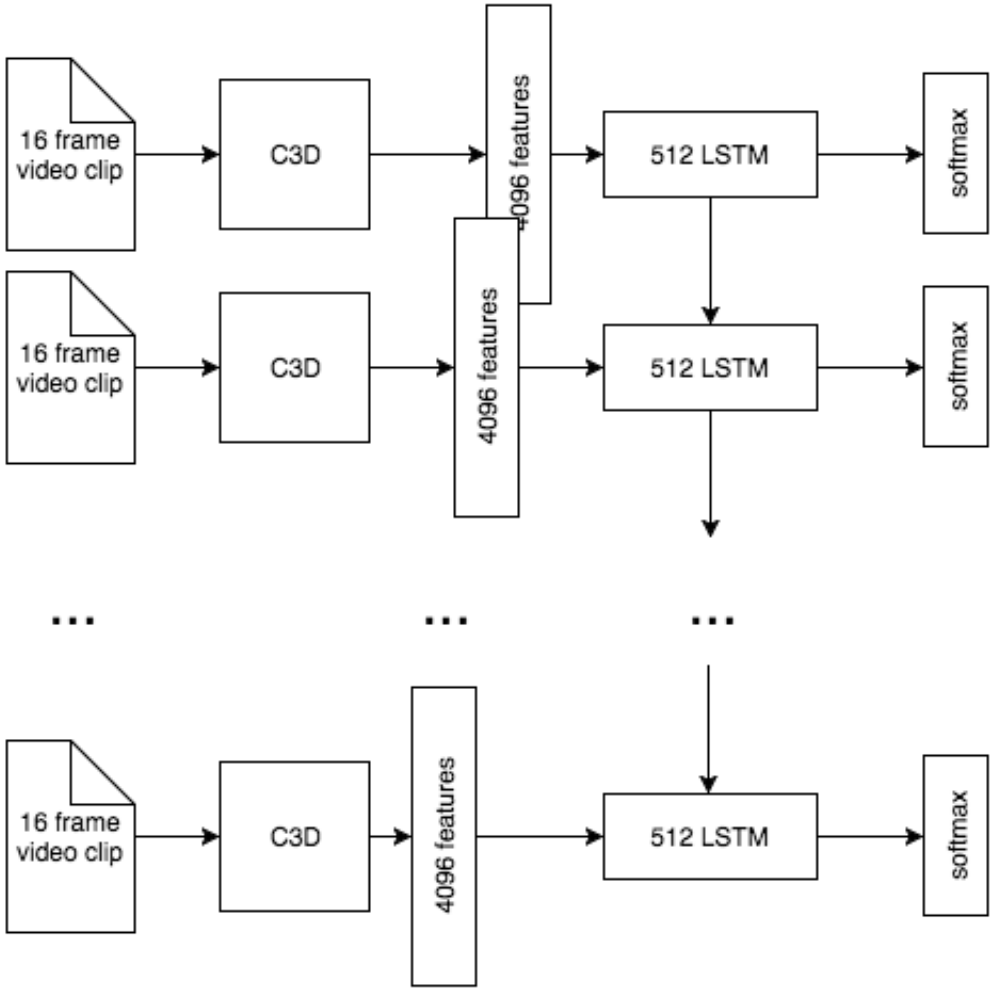
Tug of war

ACTIVITYNET

Rank	Organization	mAP
6	UPC	22.3

ACTIVITYNET

6. UPC



ACTIVITYNET

Rank	Organization	mAP
5	POSTECH*	22.8
6	UPC	22.3

ACTIVITYNET

Rank	Organization	mAP
4	University of Tokyo	26.8
5	POSTECH*	22.8
6	UPC	22.3

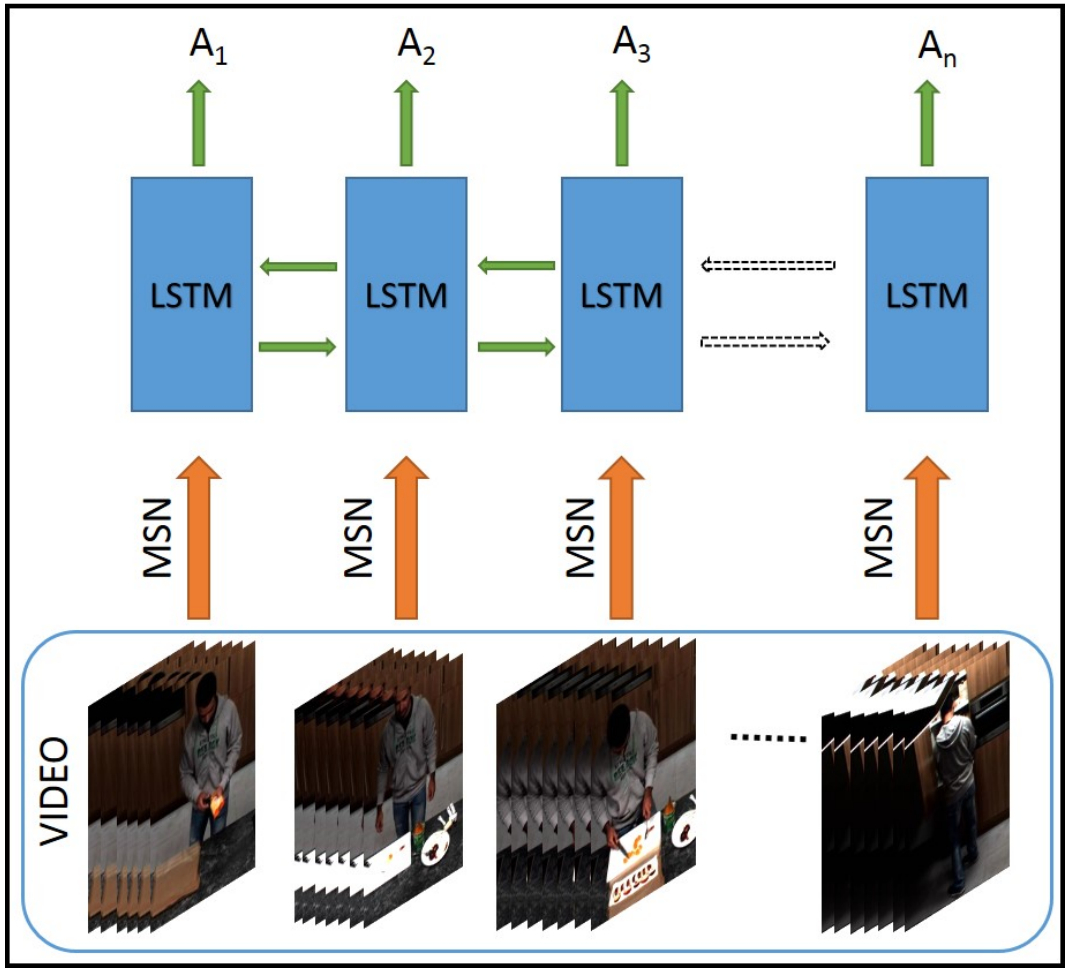
4. University of Tokyo

- Two stream network
- ResNet fine-tuned in ActivityNet
- Only action segments were used in training
- Sliding window

Rank	Organization	mAP
3	University of Maryland	28.8
4	University of Tokyo	26.8
5	POSTECH*	22.8
6	UPC	22.3

ACTIVITYNET

3. University of Maryland



A Multi-Stream Bi-Directional Recurrent Neural Network for Fine Grained Action Detection. Bharat et al. CVPR 2016

ACTIVITYNET

Rank	Organization	mAP
2	Oxford Brookes	36.4
3	University of Maryland	28.8
4	University of Tokyo	26.8
5	POSTECH*	22.8
6	UPC	22.3

2nd Place – Activity Detection Task

- TEAM: Oxford Brookes
- Prize: 1 Commemorative Plaque

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

ACTIVITYNET

Rank	Organization	mAP
1	UTS	42.5
2	Oxford Brookes	36.4
3	University of Maryland	28.8
4	University of Tokyo	26.8
5	POSTECH*	22.8
6	UPC	22.3

1st Place – Activity Detection Task

- TEAM: UTS
- Prizes: 1 GTX TITAN X, 1 Commemorative Plaque

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

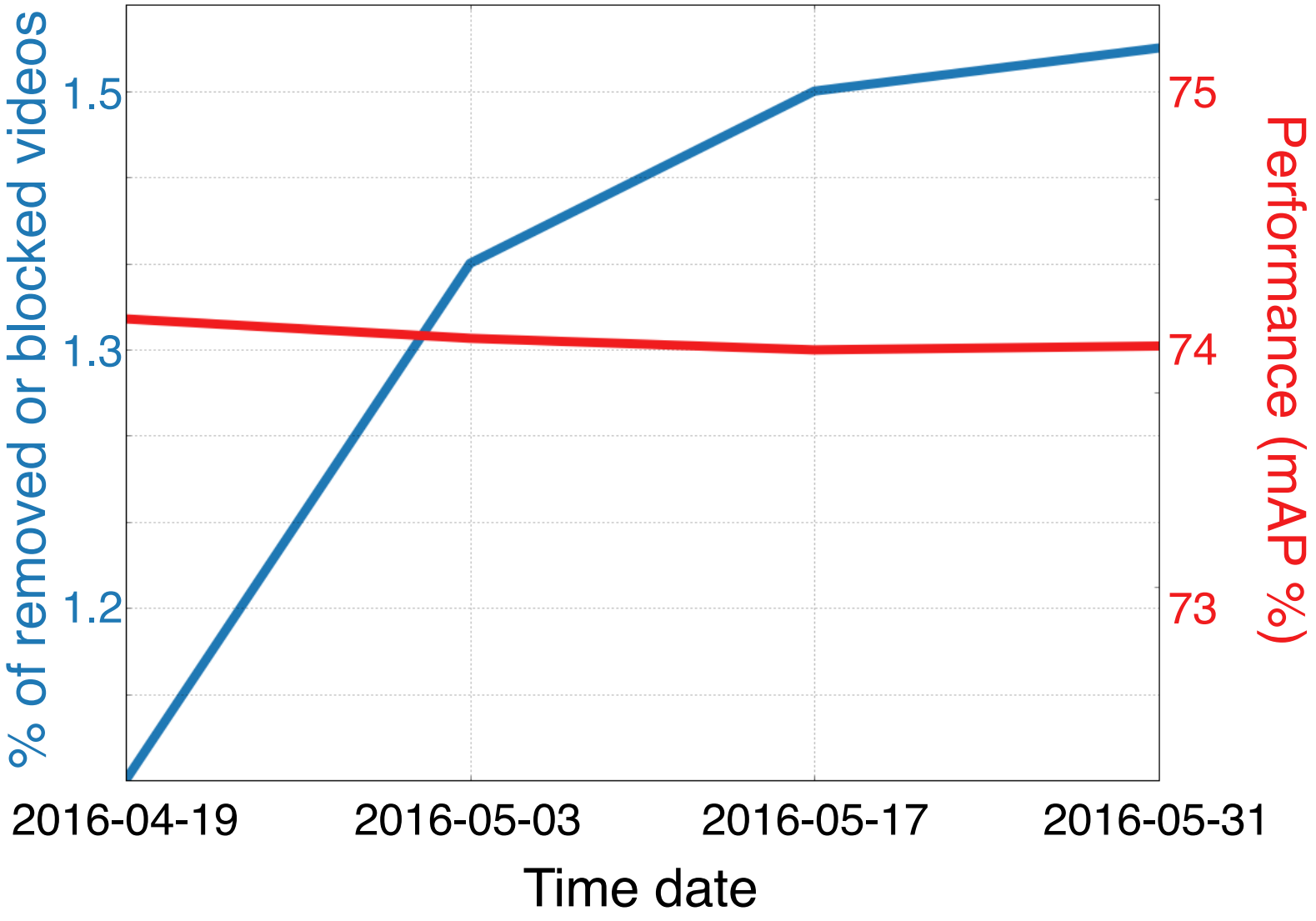
ACTIVITYNET

Conclusions & Future of ActivityNet Challenge

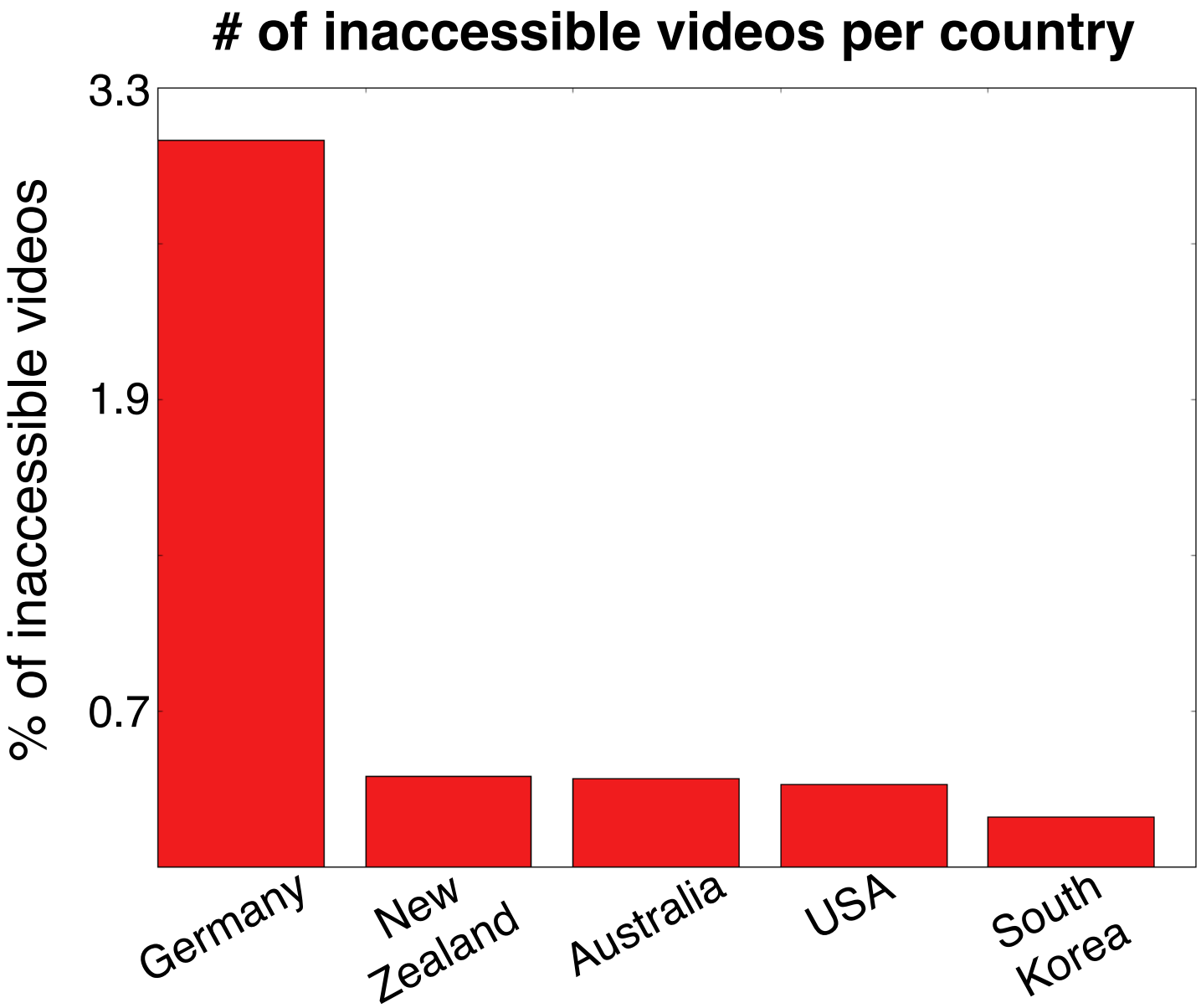
What lessons have we learned?

ACTIVITYNET

Removed or Blocked videos vs. performance

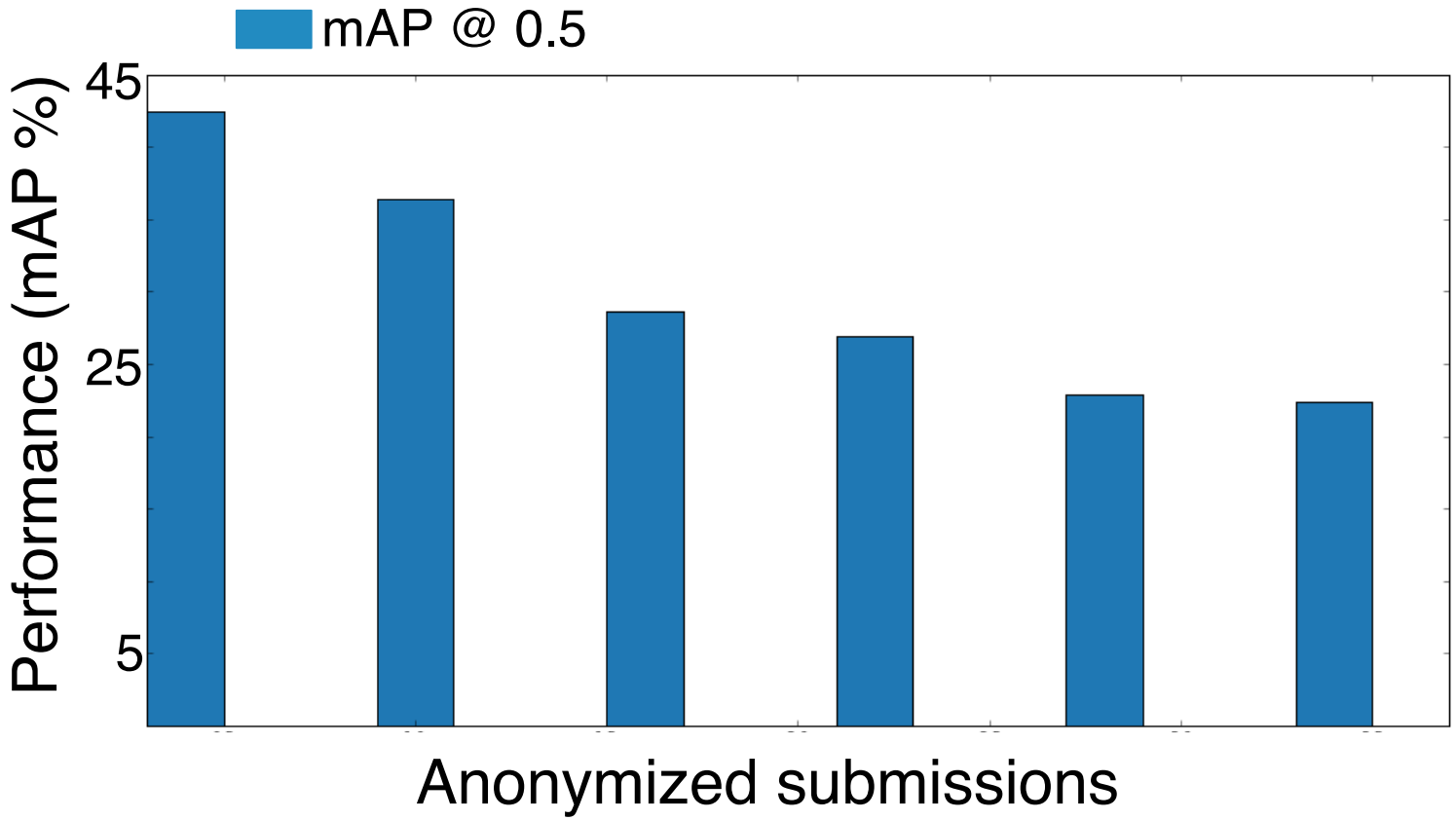


ACTIVITYNET



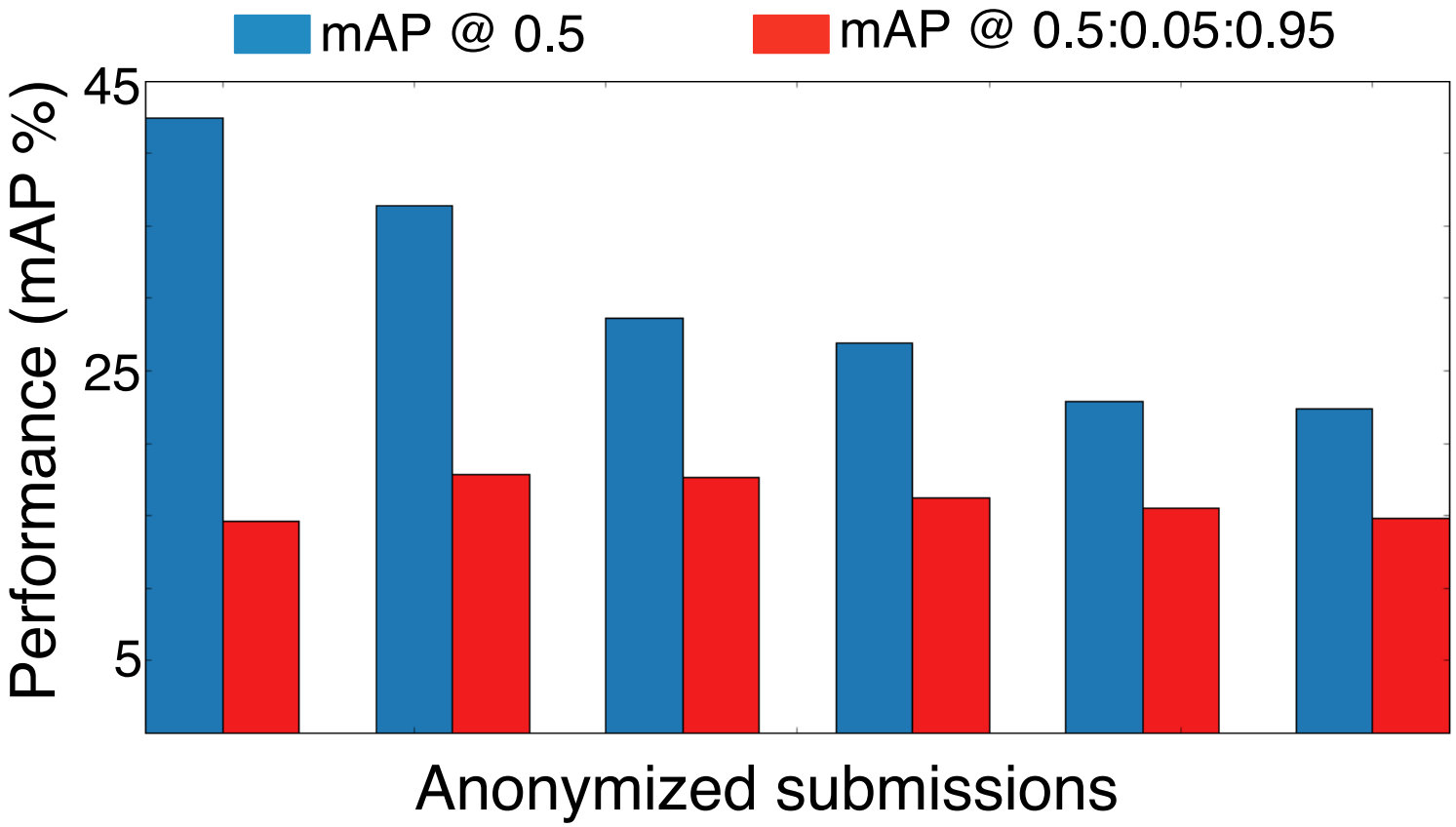
ACTIVITYNET

Metric for Detection



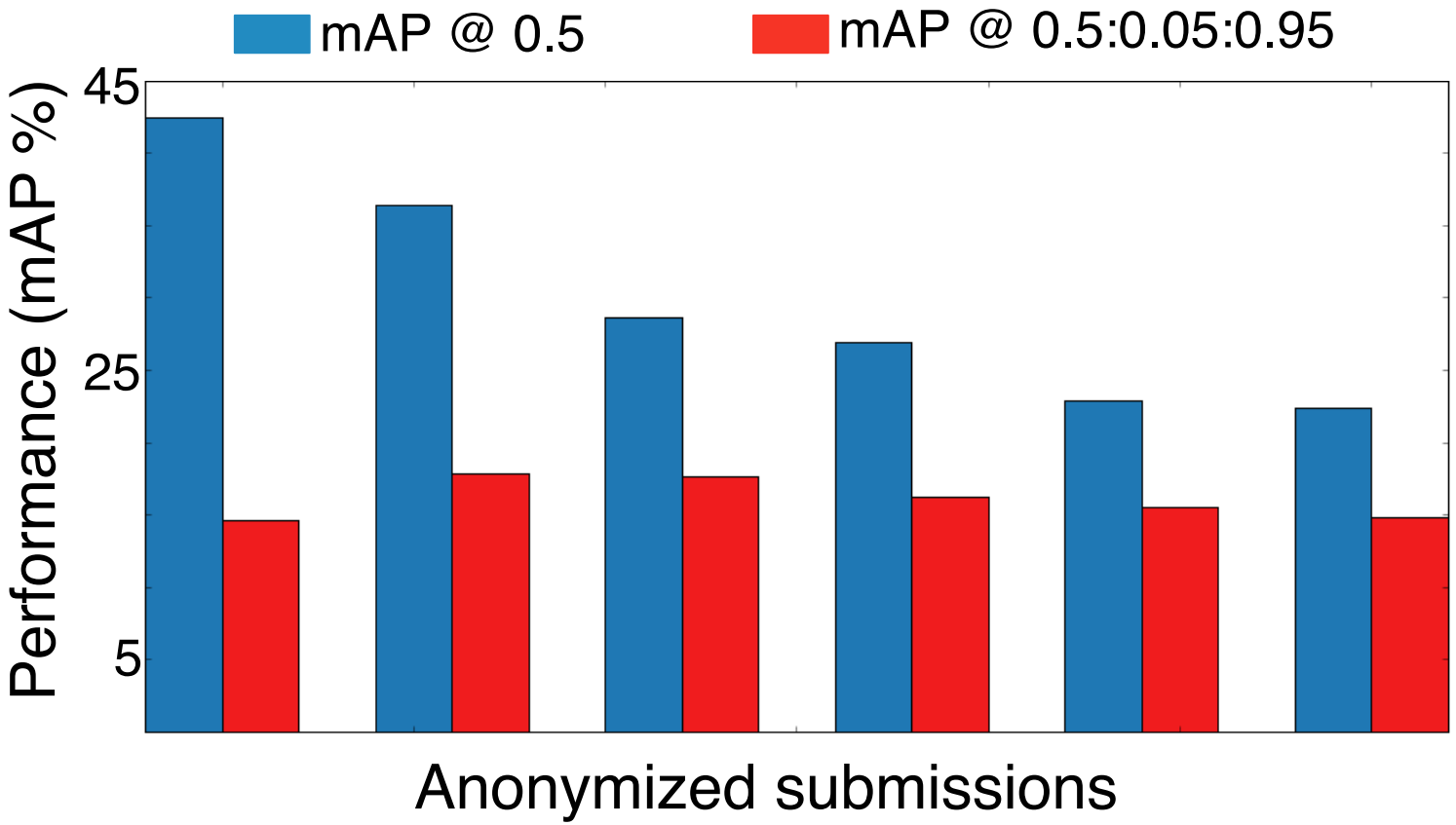
ACTIVITYNET

Metric for Detection



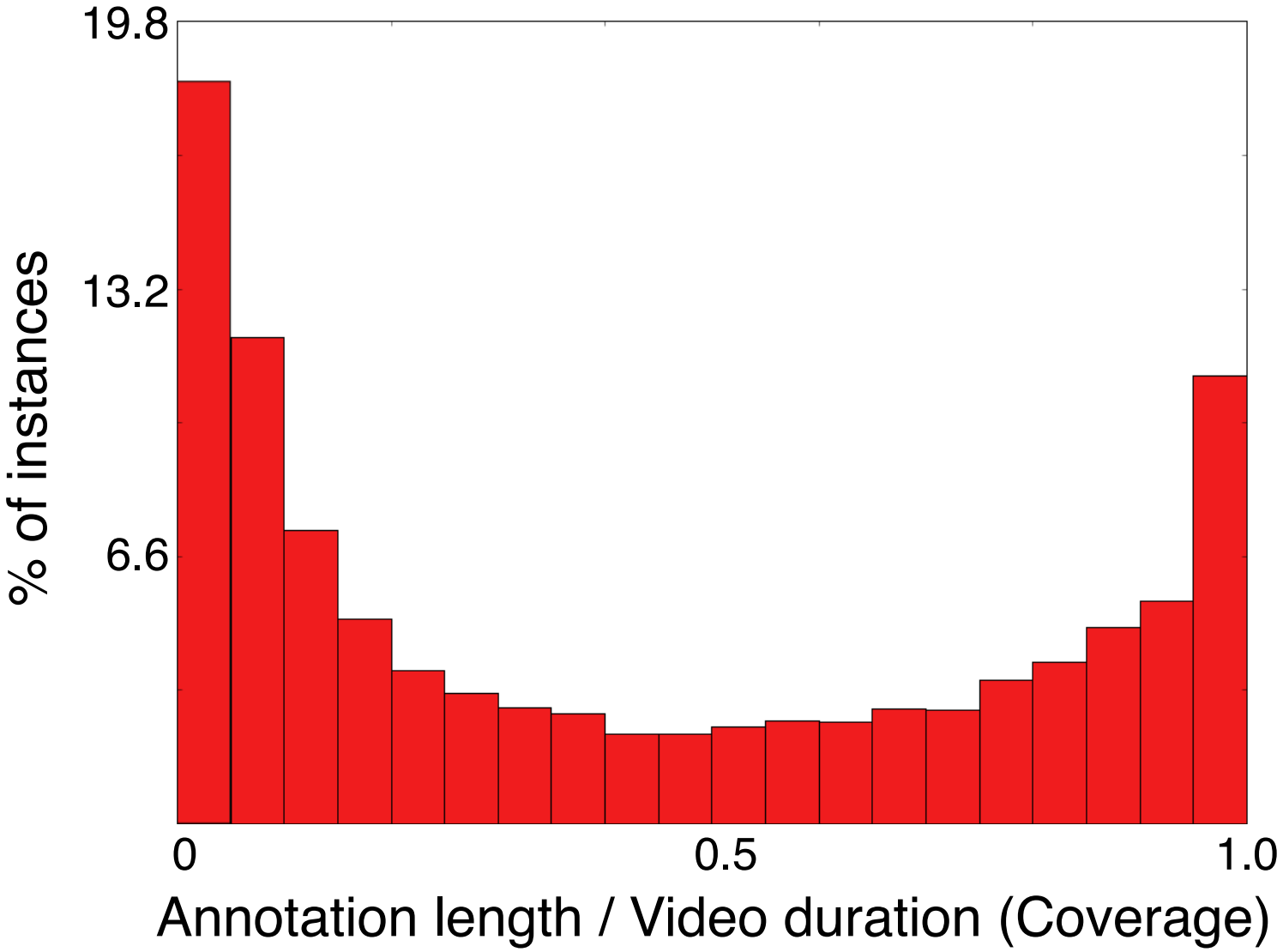
ACTIVITYNET

Metric for Detection



- **In the next challenge, we will average the AP across multiple tIoU thresholds**

ACTIVITYNET



ACTIVITYNET

Ideas for the Future

New Activity Proposal Task

- Retrieve temporal (or spatiotemporal) segments that are likely to contains actions

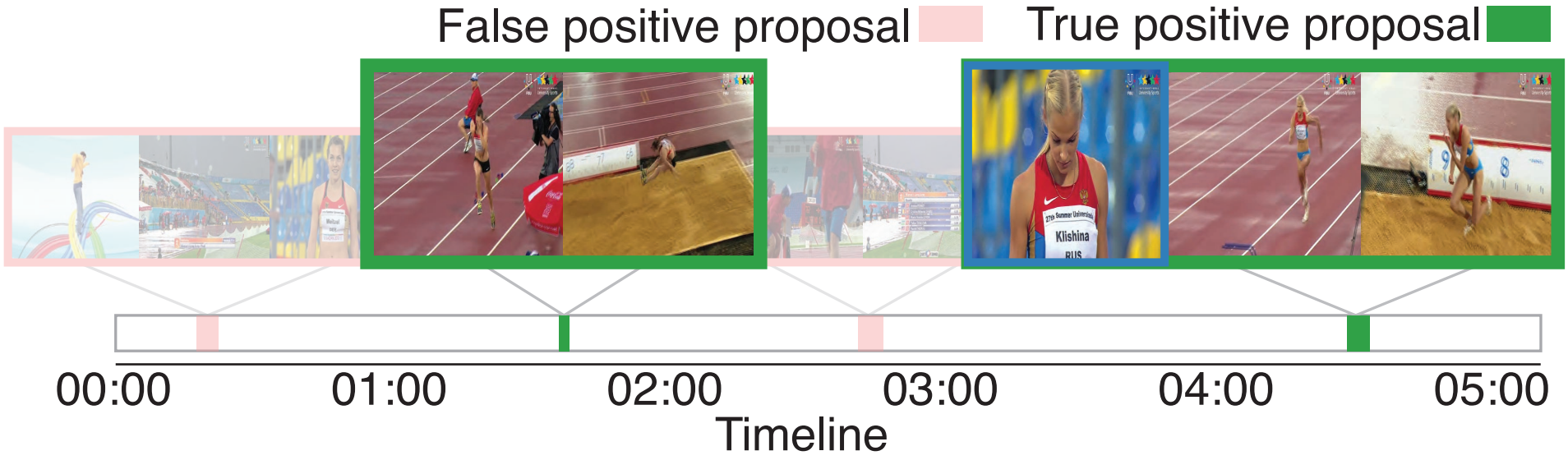
Ground-truth

Temporal Proposal

Time progress

ACTIVITYNET

New Activity Proposal Task



Fabian Caba, Juan Carlos Niebles, and Bernard Ghanem, "Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos", CVPR 2016

ACTIVITYNET

Bigger Classification Task

- Next classification task will include:
  - ~1000 action classes
  - >500 samples per class

Andrew Zisserman

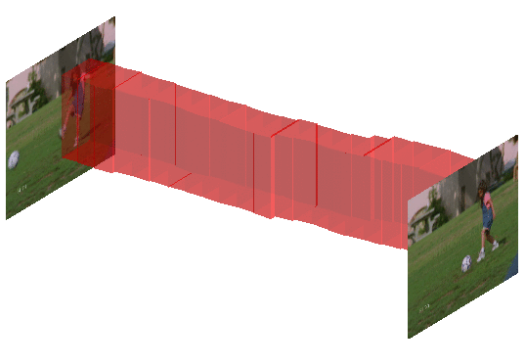
Joao Carreira

Google DeepMind

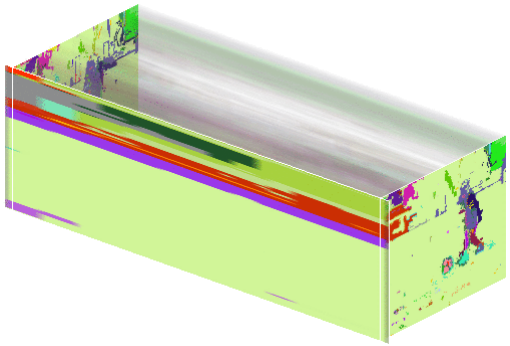
ACTIVITYNET

Extending to Spatiotemporal

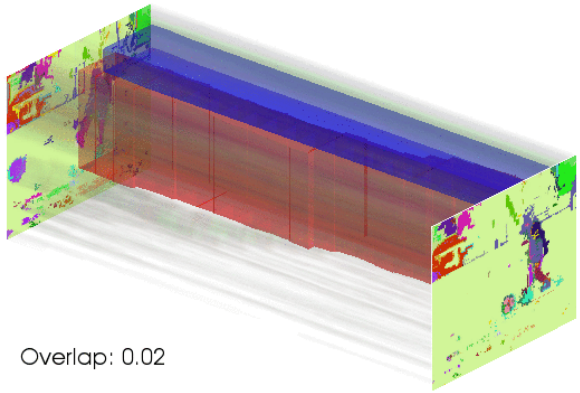
- Possible task: spatiotemporal proposal prediction



Ground truth



Super-voxel segmentation



Proposals from voxels

- Possible task: spatiotemporal detection

ACTIVITYNET

Possible dataset: Hollywood2Tubes

(a) Answer Phone. (b) Drive Car. (c) Eat.

(d) Fight Person. (e) Get out of Car. (f) Hand Shake.

(g) Hug. (h) Kiss. (i) Run.

(j) Sit down. (k) Sit up. (l) Stand up.

- Proposals on test set of Hollywood2
- Covers interaction, co-occurrence and context
- Much harder than UCF sports and UCF 101

P. Mettes, J.C. van Gemert, and C. M. Snoek, "Spot On: Action Localization from Pointly-Supervised Proposals," in ArXiv, 2016.

Feedback

Please feel free to contact us at any time with any feedback you have!

- Additional tools?
- Additional annotations?
- Annotation refinement?
- Visualization code?

ACTIVITYNET

Large Scale Activity Recognition Challenge

www.activity-net.org

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Notebook papers

UPC at ActivityNet Challenge 2016

Alberto Montes, Santiago Pascual de la Puente, Amaia Salvador, Ignasi Esquerra and Xavier Giró-i-Nieto,
Universitat Politècnica de Catalunya
{al.montes.gomez, santi.pdp}@gmail.com, {amaia.salvador, ignasi.esquerra, xavier.giro}@upc.edu

Abstract

This notebook describes our proposed solution for both the classification and detection tasks of the ActivityNet Challenge 2016. We propose a system consisting of two different stages. First, the videos are organized in 16-frame clips, for which we individually extract both audio and visual features. Visual features were extracted from a pre-trained 3D convolutional network (C3D), while MFCC coefficients were extracted for audio. On top of these features, we train a recurrent neural network to predict the activity sequence of each video at the granularity of the 16-frames clip.

1. Introduction

Recognizing activities in videos has become a hot topic over the last years due to the continuous increase of video cameras devices and online repositories. This large amount of data requires an automatic indexing to be accessed after capture. The recent advances in video coding, storage and computational resources have boosted research in the field towards new and more efficient solutions for organizing and retrieving video content.

The techniques described in this document have been tested on the video dataset defined by the ActivityNet Challenge 2016. This dataset contains 640 hours of video and 64 million frames. The ActivityNet dataset offers untrimmed videos, which means that has temporal annotations for the given ground truth class labels. Nearly half of the video hours (311 hours of video) contain a label among the 200 activity classes defined by the dataset. This dataset also give the temporal regions where activities occurs. For the details of the ActivityNet dataset please refer to the dataset description[1].

The architecture proposed is composed of two stages. First, we extract spatio-temporal features with a 3D convolutional neural network, which exploits temporal correlations in short video clips. The second stage of our proposed architecture is a Recurrent Neural Network (RNN), which exploits long term dependencies in the feature sequence. The recurrent neural network generates a sequence of predictions that naturally allows the temporal localization of the activities within a video shot.

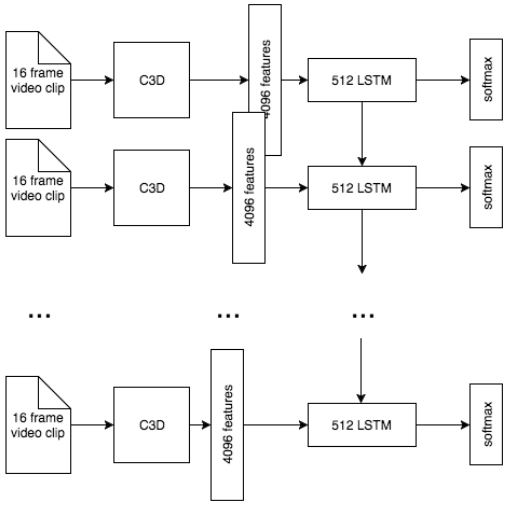


Figure 1. The proposed architecture. The network receives as input the features from the 3DS network, and trains an LSTM to output the class probability for each video clip.

2. Architecture

This section explains in detail the two stages of our proposed architecture, which is depicted in Figure 1. This architecture allows solving both the classification and detection tasks formulated in the ActivityNet challenge.

2.1. Audiovisual Feature Extraction

In order to extract spatio-temporal correlations on short clips of 16 frames, we adopted the C3D features proposed in [5], which have been proven to be well suited for video classification tasks [4] [6]. We use the network proposed in the original C3D network which was trained with the Sports1M dataset[3] and extract the features from the first fully connected layer (fc6), which was chosen based on the previous results reported in [5]. For visual feature extraction, the videos were split in clips of 16 frames each without overlap,

ending up with a total of 4 million clips. Videos clips were resized to 112x112 pixels for feature extraction, in order to match the original input size for which the C3D network was originally trained. This way, for each 16-frame clip, we obtain a visual feature of dimension 4096.

In addition to the video features, audio features were also explored as additional information for activity recognition. The audio features chosen were 40 MFCC coefficients (20 MFCC + 20 Delta-MFCC coefficients) for 20ms window length and 10ms window shift. In addition 8 Spectral coefficients for global audio track where added. The MFCC coefficients were grouped together to match video features in length and duration. The grouping of the MFCC coefficients was made computing the mean and the standard deviation. In total 88 audio features were computed. They where used in addition to the visual features in order to test if this could improve results. When used it, the audio features were concatenated to the video features out of the C3D before training the recurrent neural network.

2.2. Recurrent Neural Network

As a second stage, a Recurrent Neural Network aims at exploiting the long term dependencies in time of the extracted audiovisual features. Our RNN is based on LSTM cells, which control the flow of information that goes through them with gating mechanisms, retaining the necessary information for long periods of time, making them exploit the long-term dependencies better than classic RNNs[2]. We also proposed a sequence to sequence approach, where the model is fit with the video features as a sequence and returns a sequence of the activity class for each clip.

In addition, during our tests we explored an architecture with *feedback*, where the output predicted at the previous time step is added as an input to the LSTM. This approach aimed at smoothing the output sequence of predictions.

3. Experiments

The presented model was trained with the training partition provided by the ActivityNet challenge, and the results reported were obtained based on the predictions over the validation set.

3.1. Classification Task

For the classification task and knowing that each video has a single activity on it, we obtain the activity probabilities for the whole video as the mean of each activity output through the whole video sequence. Then, we get the maximum among all classes (excluding the background) and sort them by probability. Testing different architectures, we obtained the results given on Table 1 and Table 2. The best configuration was obtained with a single layer of LSTM with 512 neurons using only video features as input, without audio features nor feedback from the previous timestep.

Architecture	mAP	Hit@3
3 x 1024-LSTM	0.5635	0.7437
2 x 512-LSTM	0.5492	0.7364
1 x 512-LSTM	**0.5938**	**0.7576**

Table 1. Results for classification task comparing different deep architectures. All values with only video features on the validation dataset.

Features used	mAP	Hit@3
Only video	**0.5938**	**0.7576**
Video w/ audio	0.5755	0.7352
Only video & feedback	0.5210	0.6982
Video w/ audio & feedback	0.5652	0.7319

Table 2. Results for classification task with the model made by one 512-LSTM. Compare between features and feedback on the validation dataset.

α	$k = 0$	$k = 5$	$k = 10$
0.2	0.207324	**0.225138**	0.221362
0.3	0.198542	0.220776	0.221001
0.5	0.190353	0.219376	0.213029

Table 3. mAP with an IOU threshold of 0.5 over validation dataset. Here there is a comparison between values on post processing.

3.2. Detection Task

In order to solve the detection task, we post-process the output of the network with the assumption that videos only contain a single activity. This way, in this task we only focus on detecting the class with the highest probability throughout the video. To achieve this, we compute the activity probability as the sum of probabilities from all the activities except the background. A threshod α was learned and then applied along the sequence of predictions over the 16-frames clips, so that only the predictions with a probability over the threshold were considered. The best results were obtained with $\alpha = 0.2$.

Finally, a post-processing was required to improve the temporal localization of the activities. A mean filter with a window of $k = 10$ at the output of our recurrent network provided the best results, as seen in Table 3. Figure 2 shows an example of the output of our model.

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

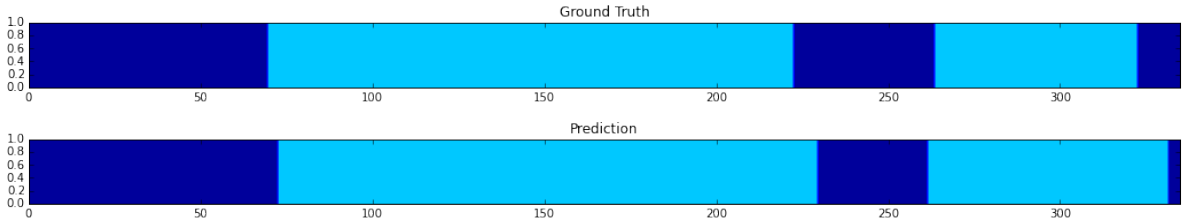


Figure 2. Example of prediction. The dark blue represents background and the light blue represents the *Rafting* activity on video `K3sJnHGHQHM`.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[4] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns.

[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014.

[6] H. Zhang, M. Xua, C. Xu, and R. Jain. Modelling temporal information using discrete fourier transform for video classification. *arXiv preprint arXiv:1603.06182*, 2016.

Action Detection

Bharat Singh
University of Maryland College Park
bharat@cs.umd.edu

1. Overview

This is an implementation of MSB-RNN [1] without tracking. VGG is replaced by ResNet-101. Detection segments within 10 seconds are merged together. Recognition is performed using detection outputs. Score for a class present in a video is obtained by max pooling scores of detected clips. If no clip is detected in the video, an arbitrary class is picked and 0 score is assigned to it. The method was only trained on the training set and parameters were tuned on the validation set.

1.1. References

References

[1] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. Activity detection. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2016.

XRCE/CVC Submission to the ActivityNet Challenge 2016

Cesar R. de Souza[1,2], Adrien Gaidon[1], Eleonora Vig[1], Antonio M. Lopez[2]
{cesar.desouza, adrien.gaidon, eleonora.vig}@xrce.xerox.com, antonio@cvc.uab.es

[1]Xerox Research Center Europe, Meylan, France
[2]Centre de Visio per Computador de la UAB, Barcelona, Spain

June 8, 2016

Abstract

In this document, we describe our method for untrimmed action recognition whose results have been submitted to the ActivityNet Challenge 2016. Our method is based on data-augmentation and feature fusion techniques for video-level Dense Trajectories and C3D features, audio-level MFCC features, and frame-level ImageNet features.

1 Description

Our method is based on the improved Dense Trajectories (iDT) pipeline of Wang & Schmid [1]. However, we employ modifications to both the beginning (data preprocessing) and end (feature fusion) of this pipeline. First, we preprocess the videos from the 1.3 version of the ActivityNet to reduce their size, downscaling them to 244 horizontal lines while keeping the aspect ratio. Then, we generate extra versions of each video using frame-skipping and horizontal mirroring. Afterwards, we proceed to extract information from both their audio and video streams.

From the audio stream, we extract a set of 40-dimensional MFCC audio features for each video. From the video stream, we extract Trajectory shape (Traj) [7], HOG [8], HOF [9], horizontal and vertical MBH components [7] descriptor along trajectories obtained by median filtering dense optical flow, using the same parameters given in [1]. We subsample the trajectories from each transformed version of each video, keeping only 10% of the originally extracted trajectory descriptors. We apply the RootSIFT normalization [3] (ℓ_1 normalization followed by square-rooting) to all video descriptors.

Next, we randomly sample 256,000 trajectories and MFCC vectors from the pool of training videos to learn the vocabularies needed for feature encoding. Before learning the GMMs, we apply PCA to the descriptors, reducing their dimensionality by a factor of two. Afterwards, we concatenate the PCA-transformed video descriptors with their respective $(x, y, t) \in \mathbb{R}^3$ coordinates.

We learn one separate GMM per descriptor channel. Both of those models are learned using the free implementations in the Scikit-learn [13] set of machine learning tools for Python. After the vocabularies have been created, we use them to create Fisher Vector (FV) [14, 15] encodings for each local descriptor in each descriptor channel, combining these encodings into a per-channel, video-level representation using sum-pooling. We then apply Power normalization [15] (signed-square-rooting followed by ℓ_2 normalization) to those per-channel FVs. Next, we concatenate all channels together and reapply this same normalization [2].

Finally, we learn separate probability-calibrated SVMs for a) iDT+MFCC Fisher Vectors b) video-level C3D features b) image-level ImageNet features (the last two being gathered from the challenge website). We concatenate the probability outputs of each of those SVMs and use it as a global feature vector, learning a fourth SVM on top of those features, and use this final SVM to predict the final scores for each video.

References

[1] Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV. (2013)

[2] Lan, Z., Lin, M., Li, X., Hauptmann, A.G., Raj, B.: Beyond gaussian pyramid : Multi-skip feature stacking for action recognition. In: CVPR. (2015)

[3] Arandjelovi, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR. (2012)

[4] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. (2015)

[5] Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. (2010)

[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal on Machine Learning Research **15** (2014) 1929–1958

[7] Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR. (2011)

[8] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)

[9] Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV. (2006)

[10] Sanchez, J., Perronnin, F., De Campos, T.: Modeling the Spatial Layout of Images Beyond Spatial Pyramids. Pattern Recognition Letters **33**(16) (2012) 2216–2223

[11] Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV **103** (2013) 60–79

[12] Wang, H., Oneata, D., Verbeek, J., Schmid, C.: A robust and efficient video representation for action recognition. IJCV (July 2015) 1–20

[13] Pedregosa, F., Varoquaux, G.: Scikit-learn: Machine Learning in Python. J. Mach. ... **12** (2011) 2825–2830

[14] Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: CVPR. (2007)

[15] Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher kernel for large-scale image classification. In: ECCV. (2010)

[16] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv1412.6980 (December 2014)

[17] Chollet, F.: keras. https://github.com/fchollet/keras (2015)

[18] Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints **abs/1605.02688** (May 2016)

[19] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks (2010)

Untrimmed Classification for Activity Detection: submission to ActivityNet Challenge

Gurkirt Singh Fabio Cuzzolin
Artificial Intelligence and Vision research group
Oxford Brookes University
{15056568,fabio.cuzzolin}@brookes.ac.uk

Abstract

Current state-of-the-art human activity recognition is focused on the classification of temporally trimmed videos in which only one action occurs per frame. We propose a simple, yet effective, method for the temporal detection of activities in temporally untrimmed videos with the help of untrimmed classification. Firstly, our model predicts the top k labels for each untrimmed video by analysing global video-level features. Secondly, frame-level binary classification is combined with dynamic programming to generate the temporally trimmed activity proposals*. Finally, each proposal is assigned a label based on the global label, and scored with the score of the temporal activity proposal and the global score. Ultimately, we show that untrimmed video classification models can be used as stepping stone for temporal detection.*

1. Introduction

Emerging real-world applications require an all-round approach to the machine understanding of human behaviour, which goes beyond the recognition of simple, isolated activities from video.

As a step towards this ambitious goal, in this work we address the problem of detecting the temporal bounds of activities in temporally untrimmed videos.

2. Methodology

Whereas (i) video-level features are used for untrimmed video classification task, (ii) frame-level features are used for activity proposal generation and scoring. Finally, (iii) a video's classification score is augmented with the scores of the activity proposals for proposal classification.

2.1. Features

We make use of the features provided on ActivityNet's [2] web page[1].

2.1.1 Video-level features

ImageNetShuffle features are video-level features generated by [4] using a Google inception net (GoogLeNet [5]). CNN features are extracted from the pool5 layer of GoogLeNet [5] at a two frames per second rate. Frame-level CNN features are mean pooled to construct a representation for the whole video. Mean pooling is followed by L1-normalisation.

We train a one-versus-rest linear SVM for each class, and use the resulting SVM scores $S^i = \{s^i_1, ..., s^i_c, ...s^i_C\}$, where C is number of classes, as INS features.

Motion Boundary Histogram (MBH) features are generated with the aid of the improved trajectories [7] executable[2]. We train another battery of one-versus-rest SVMs using a linear kernel on the MBH features, and use the resulting SVM scores $S^m = \{s^m_1, ..., s^m_c, ...s^m_C\}$ as global video features.

2.1.2 Frame level features

C3D Features features are generated at 2 frames per second using a C3D network [6] with temporal resolution of 16 frames. Once again we train a frame level one-versus-rest SVM classifier for each activity class using a linear kernel. The scoring of frame t is defined by the resulting SVM scores: $S^3_t = \{s^3_1, ..., s^3_c, ...s^3_C\}$. Finally, we perform mean pooling along the frames for each class to get another score vector S^3, which is used for video classification.

2.2. Untrimmed video classification

Untrimmed video classification is achieved by fusing all video level scores using a linear SVM as a meta classifier. Video level scores (S^i, S^m and S^3) are stacked up to make a single score vector. A linear SVM is trained on the training set of stacked scores, and evaluated on the validation and testing sets. The output scores S^s outputted by the meta SVM are normalised by dividing them by the sum of the

[1] http://activity-net.org/challenges/2016/download.html

[2] http://lear.inrialpes.fr/people/wang/improved_trajectories

top k scores. The parameter k was cross-validated on the validation set and set to 3 – it contributes to improve the mean average precision metric.

We believe that, since SVM scores are not probabilities, normalisation by top k scores is required to be able to compare them across all videos.

2.3. Activity detection in untrimmed videos

Activity proposals are detected by (i) training a binary random forest (RF) classifier [1] for each class on the frame-level C3D features, and (ii) casting activity proposal generation as an optimisation problem [3], which makes use of these binary decisions.

2.3.1 Binary random forest classification

The binary RF classifies each frame into a negative (i.e. no activity taking place) or a positive bin (i.e. something is happening). The positive score of a frame t is denoted by s_t^r. Temporal trimming is then achieved by dynamic programming as follows.

2.3.2 Activity proposal generation

Given the frame-level scores $\{s_t^r, t = 1, ..., T\}$ for a video of length T, we want to assign to each frame a binary label $l_t \in \{1, 0\}$ (where zero represents the 'background' or 'no-activity' class), which maximises:

$$E(L) = \sum_{t=1}^{T} s_t^r - \lambda \sum_{t=2}^{T} \psi_l \left(l_t, l_{t-1}\right), \qquad (1)$$

where λ is a scalar parameter, and the pairwise potential ψ_l is defined as: $\psi_l(l_t, l_{t-1}) = 0$ if $l_t = l_{t-1}$, $\psi_l(l_t, l_{t-1}) = \alpha$ otherwise, (where α is a parameter which we set by cross validation). This penalises labellings $L = \{l_1, ..., l_T\}$ which are not smooth, thus enforcing a piecewise constant solution. All contiguous sub-sequences form the desired activity proposal (which can be as many as there are instances of activities). Each activity proposal is assigned a global score S_a equal to the mean of the scores of its constituting frames. This optimisation problem can be efficiently solved by dynamic programming [3]. It can easily be extended for simultaneous detection and classification [3].

2.3.3 Overall activity detection

The top (in this case 2) activity proposals in each video are assigned the label of top untrimmed classification class (§2.2). For example, if $c = 10$ is the top class for the video with score S_{10}^s, and a is the top activity proposal with score S_a (§2.3.2), then a detection of class 10 is flagged with the temporal bounds determined by activity proposal a and score $S_{10}^a = S_{10}^s * S_a$.

3. Implementation

We used the precomputed features provided by the competition organisers. We used SciKit-learn[3] for linear SVM and random forest Implementation. We will make the activity proposal generation code available [4].

4. Results

We report results for untrimmed classification and activity detection on ActivityNet [2]. We use the same evaluation setting as described in challenge [2].

4.1. Untrimmed classification

	Validation Set			Testing Set		
Method	TOP-1	TOP-3	mAP	TOP-1	TOP-3	mAP
Caba *et al.* [2]	-	-	42.50%	-	-	42.20%
proposed	76.89%	89.25%	81.99%	77.08%	89.38%	82.49%

Table 1: Untrimmed classification performance on validation and testing set in percentage.

4.2. Activity detection

TIoU threshold δ =	0.1	0.2	0.3	0.4	0.5
Validation-Set Caba *et al.* [2]	12.50%	11.90%	11.1%	10.40%	09.70%
Validation-Set proposed	52.12%	47.94%	43.50%	39.22%	34.47%
Testing-Set proposed	-	-	-	-	36.40%

Table 2: Activity detection performance on validation and testing set. Quantity δ is the Temporal Intersection over Union (TIoU) threshold.

5. Conclusion and Future Work

We show that activity detection can be achieved via untrimmed video classification. Our dynamic programming-based approach is efficient, and has shown a clear potential for generating good quality activity proposal.

The approach can be easily extended for simultaneous detection and classification without requiring classification scores at video level, which open ups the opportunity for online activity classification, detection and prediction.

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2

[2] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1, 2

[3] http://scikit-learn.org/stable/

[4] https://github.com/gurkirt/actNet-inAct

[3] G. Evangelidis, G. Singh, and R. Horaud. Continuous gesture recognition from articulated poses. In *ECCV Workshops*, 2014. 2

[4] P. Mettes, D. Koelma, and C. G. M. Snoek. The imagenet shuffle: Reorganized pre-training for video event detection. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, New York, USA, 2016. 1

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014. 1

[7] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *Proc. Int. Conf. Computer Vision*, pages 3551–3558, 2013. 1

Multi-stream CNNs for video action recognition

Ke Ning[†*] Xiaoming Liu[‡] Fei Wu[†]
[†]College of Computer Science, Zhejiang University, China
[‡]CSE, Michigan State University, USA
{ningke, wufei}@zju.edu.cn, liuxm@cse.msu.edu

Abstract

We describe our system for ActivityNet Challenge 2016 [3] in this report. In general, we use multiple deep learned features to depict the visual information in videos. Each of the convolutional neural networks (CNNs) captures different aspects of visual information. We try to exploit the internal structural information across different CNNs. In the end, the one-vs-rest linear SVM is used as the final classifier.

1. Introduction

Action recognition is becoming an attracting problem in computer vision, especially recognizing human actions in untrimmed real-world videos. The research about video action recognition also helps in other tasks of video analysis and video understanding, such as video retrieval and video recommendation.

Hand-crafted features like improved dense trajectories [16] with HOG, HOF and MBH achieved good performance and are being widely used in video analysis. With the success of convolutional neural networks (CNNs) on image recognition [7], deep learning methods have become popular in various areas in computer vision. Many work tried to use CNN to generate better representation. Many of them are proven successful practices [10, 13] by exploit more motion information with CNNs. Some other work tried to encode CNN features to better represent video contents, like VLAD encoding [20] and recurrent neural networks (RNNs) [8].

In this report, we use CNN encoding with richer appearance and motion feature, to explore how much the improvement we are able to achieve. We present our system for ActivityNet Challenge 2016 [3] classification task in this report. Our system consists of multiple CNNs, which captures visual information from multiple aspects of video data. Then, VLAD encoding is applied to the extracted features from the last pooling layer of these CNNs. In the end, we train one-vs-rest linear SVM for each class as then final classifier.

[*]This work was done when Ke Ning was visiting Michigan State University.

2. System description

Our system consists of three parts: deep feature extraction, feature encoding and classification. We describe these three parts of our system in detail in this section.

2.1. Deep learned feature

Many recent work used two-stream CNNs [10, 17, 18, 19] to extract appearance and motion visual information from video frames and stacked optical flows, and achieved good performance for action recognition. In general, two-stream CNNs can capture the appearance and motion information in videos, which simulate the ventral stream (object appearance) and dorsal stream (object motion) in the recognition process of human brain. Two popular fusion method are early fusion (representation fusion) and late fusion (score fusion). These two fusion methods fuse information from different models at the last classification stage, which might miss the internal structural information cross different networks.

To better exploit the visual information in videos, we employ C3D network [13] and C3D flow network [14], which can describe the evolving of video frames and optical flows over short time. 3D convolution and 3D pooling operations in C3D networks have more invariance over time, captures spatio-temporal information in videos from new aspects other then regular RGB network and stacked optical flow network. In the following part of this report, we denote these two networks as RGB_{3D} and Flow_{3D} network, respectively. We use visual feature on videos of these four networks.

Since the visual information of different modalities are highly correlated over both space and time, we would like to describe local visual information with spatio-temporally aligned networks. We apply CNNs on videos with temporal stride t_0, and make the temporal center of inputs for

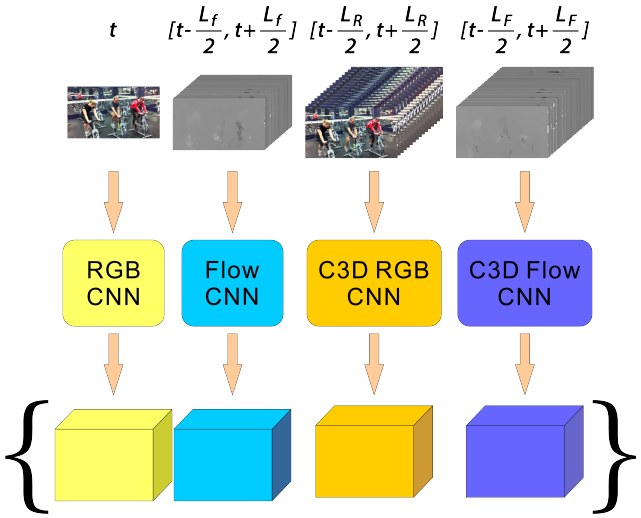


Figure 1. Multi-stream CNN fusion. L_f, L_R and L_F represents the temporal length of each network.

different networks to the same frame, so they can aligned temporally. And we concatenate the output feature map of last pooling layer to make the information spatially aligned, since the output of convolutional feature maps preserves spatial structure of input, as shown in Figure 1.

With both spatially and temporally aligned, these CNNs are able to describe local latent concepts from multiple aspects, like appearance, motion, etc. With spatial pyramid pooling [20], these networks generate 50 d-dimensional descriptors for each frame, where d stands for the sum of number of output channels of each network. We balance the weights of different networks by dividing standard deviation of each network. We also observe similar work of fusing two-stream CNNs on feature maps [4]. The difference is after fusing two networks, [4] used 3D convolution to generate representation for videos, while we use bag-of-words encoding on CNN feature maps.

2.2. Feature encoding and classification

To generate video representation, we first apply PCA-whitening on each descriptor to reduce the dimensionality and whiten the descriptors. We use VLFeat [15] implementation of Vector of Locally Aggregated Descriptor (VLAD) encoding [1, 20] to encode CNN feature maps, with power normalization, intra normalization and L_2 normalization as post-processing. In our experiments, we use VLAD-k with $k = 5$, which assigns each descriptor into the nearest 5 centers. At last, one-vs-rest linear SVMs are trained for each class, as the final classifier.

3. Experiments

3.1. Datasets

We test our system on two datasets: UCF101 and ActivityNet validation set.

UCF101 [12] is a commonly used dataset for action recognition in real-world videos, which contains $13,320$ videos from 101 action categories. There are 3 standard splits of these $13,320$ videos, and the average of accuracy over 3 splits is commonly reported. These videos are temporally trimmed, which means they are relatively short without background video clips.

ActivityNet [3] is a much larger dataset than UCF101 for action recognition in videos. It contains about $10,000$, $5,000$ and $5,000$ videos in the training, validation and testing set. These videos are temporally untrimmed, which means Some parts of these videos have no actions, and the temporal length can be much longer than videos in UCF101. Videos analysis in untrimmed videos are much more challenging than in trimmed videos. The performance on ActivityNet datasets is measured by interpolated mean average precision (mAP) and top-k accuracy with $k = 1$ or 3.

3.2. Experiment settings

We use the off-the-shelf OpenCV implementation TVL-1 algorithm to compute the optical flows. We use multiple CNNs to extract the visual feature. For RGB nets, we use ImageNet [2] pretrained Inception-BN network [5] and Sports-1M [6] pretrained C3D network [13]. The architecture of 2D flow network is VGG16 network [11]. Optical flow networks are trained on UCF101 dataset [12], initialized from corresponding RGB network as suggested in [18]. We also finetune these networks on the ActivityNet training set, but we don't have enough time for full finetuning. In all of our experiments, the temporal stride t_0 is set to 8 frames. The temporal length L_f, L_R, L_F of Flow CNN, RGB$_{3D}$ CNN and Flow$_{3D}$ CNN are 10, 16 and 16, respectively.

While encoding descriptors, we set the dimensionality of descriptors after PCA-whitening to $512 \times N_c$, where N_c is the number of networks being used. We set $C = 100$ in our SVM training. For the validation set, we train SVMs with videos in the training set. For testing set, we train SVMs with videos in both the training set and the validations set.

3.3. Experiment results

3.3.1 UCF101

We first test our system on UCF101 dataset. The results are shown in Table 1 and Table 2. Across single stream, 10-frame stacked flow network achieves the best performance. For any two streams, 2D RGB net and 2D Flow net achieves the best performance. Comparing to the state of the arts, our best result can outperform others on the average accuracy

by more than 1 percentage point.

Surprisingly, we observe that the performance of 4-stream is slightly worse than that of three streams without Flow_{3D}. But multi-stream CNNs fusion can still result in a reasonably good performance.

Features	Average accuracy
RGB net	82.95%
Flow net	89.14%
RGB_{3D} net	83.95%
Flow_{3D} net	87.57%
RGB+Flow	93.02%
RGB+Flow_{3D}	92.08%
RGB_{3D} + Flow	92.74%
RGB_{3D} + Flow_{3D}	91.94%
RGB + Flow + RGB_{3D}	**93.87%**
4-stream	93.51%

Table 1. The results on UCF101 dataset of different streams.

Methods	Average accuracy
iDT+FV [16]	85.9%
iDT+HSV [9]	87.9%
Two-stream CNNs [10]	88.0%
TDD+FV [17]	90.3%
Very deep two-stream CNNs [18]	91.4%
Transformations [19]	92.4%
Conv. two-stream CNNs [4]	92.5%
Ours	**93.87%**

Table 2. Comparison with state of the arts on UCF101.

3.3.2 ActivityNet

We test our system on the validation set of ActivityNet. Among all the single streams, the RGB network performs the best. This could because for real-world videos, most of the videos are in unconstrained environments. The background scene of videos can contain much information related to the action. Unlike the results in UCF101, the performances of two flow networks are relatively low. This could be due to the fact that these two networks are trained on a small dataset (UCF101). The final result of finetuned features is slightly worse than the one without finetuning. By applying fusion on finetuned and unfinetuned model, the result is slightly better.

As a comparison of different fusion methods, we applied early fusion and late fusion over these four networks (denoted as EF and LF in Table 3). We can see that both early fusion and late fusion have worse performance than jointly encoded representation. Different from UCF101, four streams outperforms three streams.

Features	mAP	Top-1 accuracy	Top-3 accuracy
RGB net	0.7628	77.35%	89.52%
Flow net	0.5948	60.80%	76.86%
RGB_{3D} net	0.7084	72.02%	85.23%
Flow_{3D} net	0.5780	59.45%	74.44%
RGB+Flow	0.7852	79.79%	91.18%
RGB+Flow_{3D}	0.7824	79.55%	90.85%
RGB+Flow+RGB_{3D}	0.7938	79.88%	91.71%
EF 4-stream	0.7910	79.88%	91.41%
LF 4-stream	0.7734	79.51%	90.67%
4-stream	0.7950	80.51%	91.94%
finetuned 4-stream	0.7925	80.23%	91.34%
Fusion	**0.8031**	**81.07%**	**92.14%**

Table 3. The results on the validation set of ActivityNet.

4. Conclusions

Fusing multiple CNNs on the feature map level can help with capturing more relevant cues from different neural networks, while these networks are capturing information from different aspects of visual data. Also, multi-stream CNNs can be used as a good feature extractor for further vision tasks in video analysis.

References

[1] R. Arandjelovic and A. Zisserman. All about vlad. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1578–1585. IEEE, 2013.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[3] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. *arXiv preprint arXiv:1604.06573*, 2016.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In

Advances in neural information processing systems, pages 1097–1105, 2012.

[8] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *arXiv preprint arXiv:1503.08909*, 2015.

[9] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *arXiv preprint arXiv:1405.4506*, 2014.

[10] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014.

[14] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. 2015.

[15] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[16] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558. IEEE, 2013.

[17] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. *arXiv preprint arXiv:1505.04868*, 2015.

[18] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[19] X. Wang, A. Farhadi, and A. Gupta. Actions˜ transformations. *arXiv preprint arXiv:1512.00795*, 2015.

[20] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1798–1807, 2015.

CUHK&ETHZ&SIAT Submission to ActivityNet Challenge 2016

Yuanjun Xiong[1], Limin Wang[2], Zhe Wang[3], Bowen Zhang[3], Hang Song[1], Wei Li[1], Dahua Lin[1], Yu Qiao[3], Luc Van Gool[2] and Xiaoou Tang[1]

[1]Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong
[2]Computer Vision Lab, ETH Zurich, Switzerland
[3]Shenzhen Institutes of Advanced Technology, CAS, China

Abstract

This paper presents the method that underlies our submission to the untrimmed video classification task of ActivityNet Challenge 2016. We follow the basic pipeline of very deep two-stream CNN [15] and further raise the performance via a number of other techniques. Specifically, we use the latest deep model architecture, e.g. ResNet and Inception V3 and introduce a new aggregation scheme (top-k and attention-weighted pooling). Additionally, we incorporate the audio as a complementary channel, extracting relevant information via a CNN applied to the spectrograms. With these techniques, we derive an ensemble of deep models, which, together, attains a very high classification accuracy (mAP 93.23%) on the testing set.

1. Introduction

In the past several years, the advance in deep learning techniques has given rise to a new wave of efforts towards vision-based action understanding. A number of deep learning based frameworks, including two-stream CNNs [7], 3D CNNs (C3D) [11], and Trajectory-pooled Deep convolutional Descriptors (TDD) [13], have been developed, which significantly pushed forward the state-of-the-art [12, 14]. Such improvement on performance, to a large extent, is owning to both the modeling capacity of deep architectures and more effective learning strategies.

However, it is worth noting that previous efforts focus mainly on the analysis of short video clips. These clips are typically extracted from longer videos such that they only contain the portions of frames that truly capture the actions of interest. Obviously, preparation of such data is a laborious procedure. Action recognition from *untrimmed videos*, a problem that is more pertinent to real-world demands, is drawing increasing attention from the community. While substantially reducing the efforts needed in manual annotation, this task on the other hand presents a new challenge to the recognition system – a significant (or even dominant) fraction of a given video is irrelevant to the action of interest.

Driven by the ActivityNet benchmark [1], we develop an integrated approach to recognizing actions from untrimmed videos[1]. Our approach follows the framework of very deep two stream CNNs presented in our earlier paper [15], which allows both appearance and motion patterns to be effectively fused and introduces various techniques to improve the training procedure, *e.g.* temporal pre-training, and scale jittering augmentation. On top of this framework, we develop several new techniques to further improve the recognition accuracy. While visual analysis plays a primary role in this task, we notice that the audio channels that come with these videos provide complementary information. To exploit such information, we develop a deep network called Audio CNN to derive complementary features from the spectrograms.

Combining both the visual and acoustic models, we attain a high recognition accuracy (mAP 93.23% on testing set). We want to emphasize that this performance is obtained only using the training data provided by the ActivityNet benchmark except using CNNs pre-trained on ILSVRC12 data for initialization – no additional data or annotations are used throughout both the training and testing procedures.

The rest of this paper is organized as follows. Section 2 presents our approach in detail, Section 3 reports our results under a variety of settings, finally Section 4 concludes this work.

[1]Codes and models will be available at https://github.com/yjxiong/anet2016-cuhk

Table 1. Performance of different network architectures on ActivityNet v1.3 validation set. Performance is measured by per-class mean average precision (mAP) and top-3 prediction accuracy. We use the variant "basic+a" in training these models.

Settings	Spatial Nets			Temporal Nets		
	BN-Inception	Inception V3	ResNet	BN-Inception	Inception V3	ResNet
mAP	79.7%	83.3%	83.3%	63.3%	64.3%	-
Top-3 Acc.	89.6%	91.5%	91.6%	77.0%	77.9%	-

2. Our Approach

Our approach to untrimmed video classification comprises two complementary components: visual and acoustic modeling. The visual analysis, which combines a variety of techniques, plays a primary role in this framework, while the acoustic model exploits complementary information from the audio channels to further improve the performance. Next, we present these components respectively in Section 2.1 and 2.2.

2.1. Visual Analysis System

Our visual analysis component works as follows: it samples multiple snippets from a given video, makes snippet-wise predictions using very deep two-stream CNNs, and finally aggregates the predictions via different strategies such as top-k and attention-weighted pooling.

Snippet-wise Predictor Deep convolutional neural networks (CNN) which learns from multiple modality of input data has been used extensively in visual recognition tasks [8, 16, 17, 2] and achieved superiority over models using a single modality. The snippet-wise predictor in our approach is a realization of the very deep two-stream CNN framework [15] which consists appearance and motion modeling parts. In this work, we adopt the recently proposed network architectures such as **ResNet** [3] and **Inception V3** [9] to improve the capacity of the frame-wise predictor.

During training of the snippet-wise predictor, the techniques introduced in [15], such as scale jittering and stronger dropout, is also applied to the these architectures. To further boost the performance, we experimented with the idea to sample several snippets from one input video to jointly train the CNNs by averaging the per-snippet prediction. This also enables us to apply more advanced aggregation techniques into the training process.

Video-level Classification To obtain video-level classification results, we use the following strategy: the snippet-wise predictor is first applied to an input video snippet with a 1FPS sampling rate, then an aggregation module will combine the snippet-wise class scores into the final prediction. We experimented with several advanced strategies for combing snippet-wise scores of the appearance nets. These include top-k pooling and attention weighted pooling. These strategies, when used in both training and testing, produced models that are complementary to each other and thus form effective components in the final ensemble.

Table 2. Performance comparison of the appearance modeling CNN variants on the validation set of ActivityNet v1.3. Here we analyze their performance using the Inception V3 [9] architecture. In the table, "basic" refers to the baseline approach in [15], "a" refers to models trained with multiple snippets from one video, "b" refers to models equipped with advanced aggregation strategies.

Variants	mAp	Top-3 Acc.
basic	82.9%	91.0%
basic+a	83.3%	91.5%
basic+ab	84.2%	92.1%
Ensemble	85.9%	92.9%

Table 3. Performance of different components in the visual analysis system on the validation set. Here, "Appearance CNN" refers to the appearance modeling part. "Motion CNN" refers to the motion modeling part. "Combined CNN" refers to the results by combining both appearance and motion modeling parts. "Visual All" refers to the results by further combining scores from other methods such as IDT [12] and TDD [13].

Variants	mAp	Top-3 Acc.
Appearance CNN	85.9%	92.9%
Motion CNN	68.3%	80.2%
Combined CNN	89.7%	95.0%
Visual All	90.4%	95.2%

2.2. Acoustic Analysis System

Audio signals in a video carry important cues for recognizing some action classes. To harness the information in this aspect, we combine the standard MFCC [5] representations with audio-based CNNs [10] to form the acoustic modeling system.

MFCC Mel Frequency Cepstral Coefficients (MFCC) [5] is a powerful feature descriptor used in automatic speech recognition system. In our approach, we extract MFCC features from companioned audios of the videos in the dataset, and train SVMs on descriptors aggregated with Fisher Vector [6]

Table 4. Performance of acoustic models on ActivityNet v1.3 validation set. Performance is measured by per-class mean average precision (mAP) and top-3 prediction accuracy. Here, "Gray" refers to the models trained with grayscale inputs. "MS" refers to the model trained with multiple time scales.

Methods	mAP	Top-3 Acc.
MFCC (FV+SVM)	14.2%	26.1%
Audio CNN	8.0%	17.1%
Audio CNN Gray	9.3%	19.3%
Audio CNN Gray+MS	10.3%	20.7%
Audio Ensemble	15.2%	29.1%

Table 5. Performance of fusion models on ActivityNet v1.3. Performance is measured by per-class mean average precision (mAP) and top-3 prediction accuracy. In "Visual + Audio" setting, we combine the visual and acoustic modeling system. On the testing set, we present the results of "Final Ensemble" where all components trained on training plus validation data are combined.

Validation Set	mAp	Top-3 Acc.
Visual	90.4%	95.2%
Audio	15.2%	29.1%
Visual + Audio	90.9%	95.6%
Testing Set	**mAP**	**Top-3 Acc.**
Visual CNN (Single)	91.2%	95.6%
Final Ensemble	93.2%	96.4%

Audio CNN The basic idea of Audio CNN works is to apply CNNs on spectrograms, or time-frequency-response maps, of audio signals. In this work, we propose to directly use the *grayscale* time-frequency map image to train the audio CNN. Then the audio CNN can be initialized by the same technique used on the temporal networks in [15]. It is also known that learning from multiple time scales help in acoustic models [18]. In this sense, we propose to stack multiple spectrograms with varying window size as the input to the audio CNN.

3. Experiments

We train our models on the official training set of ActivityNet v1.3 dataset [1]. There are $10,024$ videos for training, enclosing 15410 activity instances from 200 activity classes. The validation set contains 4926 videos and 7654 activity instances. We study the performance of our approach on this validation set. The final testing set comprises 5044 videos and is not annotated with any activity instance. We report the performance of our proposed models on this set according to the feedback of the test server of the challenge. Models for this setting are trained with the union of training and validation set.

In experiments, we compare the performance of very deep two stream CNN [15] using several network architectures, including BN-Inception [4], Inception V3 [9], and ResNet [3]. The performance of different network structures for spatial and temporal stream are summarized in Table 1. To analyze the effect of different training strategies, we compare the performance of appearance modeling CNNs with these strategies. The results are presented in Table 2. The contributions of appearance and motion CNNs are also summarized in Table 3. Then we report the performance of the two components in the acoustic analysis systems in Table 4.

Finally, we evaluate the fusion of visual analysis system and audio analysis system on both the validation and testing set. The results are illustrated in Table 5. The best mAP achieved by the final ensemble is 93.2%. We also took one chance on the testing server to evaluate a combination of one appearance CNN and one motion CNN. Its results are presented as "Visual CNN (Single)" in Table 5. It is exciting to see using this "single model" setting we can still achieve a reasonable mAP of 91.2%, which may better fit for industrial applications.

4. Conclusions

This paper has proposed an action recognition method for classifying temporally untrimmed videos. It is based on the idea of combining visual analysis and acoustic analysis. The results show that by carefully designing the visual and acoustic analysis systems and combining them, we can achieve exciting results in video classification tasks and boost the performance of state-of-the-art methods. Another fact to be noticed is that this high accuracy is achieved by evaluating only 1 frame per second, equivalent to only seeing around 4% of all frames of input videos. We believe this property is also very important for practically applying the system in industrial scenarios.

References

[1] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 1, 3

[2] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015. 2

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[4] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 3

[5] D. OShaughnessy. Invited paper: Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, 41(10):2965–2979, 2008. 2

[6] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013. 2

[7] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2, 3

[10] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool. Deep convolutional neural networks and data augmentation for acoustic event detection. *arXiv preprint arXiv:1604.07160*, 2016. 2

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1

[12] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 1, 2

[13] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 1, 2

[14] L. Wang, Y. Qiao, and X. Tang. MoFAP: A multi-level representation for action recognition. *IJCV*, pages 1–18, 2015. 1

[15] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 1, 2, 3

[16] Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. In *CVPR*, pages 1600–1609, 2015. 2

[17] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, 2016. 2

[18] Z. Zhu, J. H. Engel, and A. Hannun. Learning multi-scale features directly from waveforms. *arXiv preprint arXiv:1603.09509v2*, 2016. 3

MIL Participation on ActivityNet Challenge 2016

Masatoshi Hidaka, Katsunori Ohnishi, Atsushi Kanehira, Shohei Yamamoto, Keisuke Fukuta, Akio Hayakawa, Yuichiro Kikura, Tennin Yan, Takayoshi Takayanagi, Yoshitaka Ushiku, Tatsuya Harada
Machine Intelligence Laboratory @ The University of Tokyo

Method

Our method is based on Two-stream [1] video classification algorighm. We finetuned ResNet-101, ResNet-152 [2] pre-trained with ImageNet [3] using ActivityNet dataset for classifying each action frames into 200 classes. However, CNNs trained with optical flow image gave much worse accuracy than RGB image for this dataset. Besides original ResNet-101 separately trained with RGB images and optical flow images, we trained end-to-end fusion network (ResNet-101-fusion). In the network, outputs of conv5 layer of RGB network and optical flow network are stacked and followed by 1x1 convolution layer to reduce the number of channels to match the original shape. Other part of the network is same as original ResNet-101.
Frames not annotated with action (non-action frames) are not used for training.
For classification task, activations from last convolutional layer for a CNN are coded with VLAD [4] method and sum-pooled along whole video. One-vs-the-rest linear SVM is used for classify the video. Prediction scores from SVMs trained with different CNNs are summed with weight to gain final prediction score. CNNs we used are ResNet-152 finetuned with RGB image, ResNet-152 not finetuned, ResNet-101 finetuned with RGB image, ResNet-101-fusion finetuned with RGB and optical flow image. Additionally, we added prediction scores trained with three features provided by the organizers. Finally, for improving mean average precision, softmax function is applied to the scores from each video.
For detection task, sliding window based approach is applied. When training, action window classifier is trained for each action. Windows whose intersection over union (IoU) is higher than 0.5 are used as positive samples and other windows are negative. When detecting, first classifiers for classification task are applied to whole video and top 1 action is used as candidate. Window classifier for the candidate class is applied to the video and the windows with certain threshold are output as detection result.
For detection task, sliding window for averaging classification score for each frame is applied, and the windows higher than a threshold are output.

References

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

ActivityNet Challenge 2016 submission

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. CVPR 2016.
[3] Jia Deng,Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. CVPR 2009.
[4] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. CVPR 2010.

Spatiotemporal Features for Action Classification

Mujtaba hasan
Indian Institute of Technology
New Delhi, India
Mujtaba.hasan@live.com

Abstract

We describe our approach to the task activity classification in the activitynet challenge hosted at CVPR 2016. We use different spatio-temporal cues from frames and sequence of frames using 2-D and 3-D convolutional neural networks(CNN). The action prediction is done by combining the predictions of one-versus-rest linear SVMs learnt for each cue.

1. Introduction

Action recognition in videos is of key importance for many applications, including but not limited to video surveillance, video summarization, video retrieval etc. It is a challenging problem because of various difficulties like background clutter, view-point change and various action styles. The activity-net challenge 2016 represents one of the most comprehensible data for this task. It consists of more than 5000 untrimmed long test videos with actions ranging over 200 classes.

The performance of any action recognition system depends on the representation of the video. In this paper, we present our approach for this challenging task. We use frame based features and small clip based features extracted by CNNs.

2. Our approach

Our approach consists of feature extractions from uniformly separated frames of the video, non-overlappling 16-frame long clips and then linearly combined to form the video representations. Then we train 200 one-versus-rest SVMs to perform the classification task.

3. Feature extraction

We extract features from frames as well as short continuous clips of 16 frames from the video.

3.1. Frame features

Some actions are strongly attached to particular scenes which makes individual frames an important source of features. We make use of VGG_19[1] for individual frame feature extractions. VGG_19 is one of the most superior deep convolutional network and consists of 19 layers, 16 convolutional and 3 fully connected layers. It is trained on a large dataset, imagenet with millions of images and can extract visual concepts from a wide range of scenes and objects. We finetune the original VGG_19 network on activity-net dataset and extract outputs of fully connected layers fc7 and fc8. Then we apply mean pooling on features from sampled frames to create a frame based video representation.

3.1. Short clip features

Other than individual frames, we also use 3D Convolutional Neural Networks (3D CNN) to construct video representation from both spatial as well as temporal domains. We use the network architecture of C3D [5], which has alternating layers of 3D convolutional and 3D pooling layers with input data of continuous frames comprising a short clip of the video. We extract the fc7 and fc8 layer features from this network.

We finetune the C3D network for our dataset by removing the last softmax layer with the layer comprising of 200 outputs. The input to C3D network is 16-frame clip, we generate non-overlapping 16-frame clips of the video and input them into the C3D network and afterwards, apply the mean pooling to obtain the video representation.

4. Classification

We use the linear SVM with the linear kernel to combine all the features described above and do the classification. We set the the SVM parameter, C equal to 10 for all the features and then train 200 one-versus-rest classifiers with all the features L2-normalized.

Table 1: Single source results on validation set

Source	Layer	mAP
Frames	fc7	0.45
	fc8	0.42
Short clips	fc7	0.6
	fc8	0.62

Table 2

Source	mAP
Frames fc7+fc8	0.44
Clips fc7+fc8	0.6
All four combined	0.55

5. Experiment

We perform the experiment on the validation set with different set of features from each domain, frames and short clips. We combine and take individually, the fc7 and fc8 features from each source and compute mAP score over the validation set. It is tabulated in the table 1 above. We also combine the layers from each source individually and overall which is tabulated in table 2. We used caffe to develop and experiment our system on a single K40 GPU

6. Conclusion

In this challenge, we mainly focused on different short-term and long-term spatio-temporal features which are good for describing actions. Optical flow features and improved dense trajectories based video representations can be used as well for future directions.

7. References

[1] K. Simonyan and A.Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv, 2014

[2] L. Wang, Y. Qiao and X. Tang. Action recognition with trajectory-pooled deep convolutional descriptors. In CVPR June 2015

[3] Z.Xu, Y.Yang and A.G. Hauptmann. A discriminative CNN video representation for event detection. In CVPR June 2015

[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri. Learning spatiotemporal features with 3d convolutional networks, In ICCV June 2015

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R.Girschick, S. Guadarrama and T. Darrel, Caffe: Convolutional architecture for fast feature embedding, arXiv 2014

[6] A. Karpath, G. Toderici, S.Shetty, T.Leung, R.Sukthankar, and L. Fei-Fei, Large scale video classification with convolutional neural networks, CVPR 2014

Activity Net

Po-Yao Huang

June 2016

1 Features

The following are the features been used:

1. Residual Net CNN feature. Each one is a 2048-dimension semantic feature. Skip 2 in every 3 frames and then perform average pooling to form the representation.
2. C3D feature. I use the provided 500-dimension and perform average pooling.
3. MFCC feature. Window size is 20 ms with hanning window and shift by 10ms. Build bag-of-word with 3000 k-means and generate the histogram accordingly.

2 Classifier

I perform grid search for SVM parameter tuning and pick the best model. The cost is between $10^{-1} \ldots 10^{3}$. The gamma value is between $10^{-2} \ldots 10^{2}$. The kernel is with linear, Chi-Square and RBF.

UTS at ActivityNet 2016

Ruxin Wang Dacheng Tao
Centre for Quantum Computation and Intelligent Systems
Faculty of Engineering and Information Tchnology
University of Technology Sydney, Ultimo, NSW, Australia
Ruxin.Wang@student.uts.edu.au Dacheng.Tao@uts.edu.au

Abstract

In this notebook paper we present an overview of our solution to the ActivityNet Large Scale Activity Recognition Challenge 2016. We report system designs for both the untrimmed video classification task and the activity detection task. Specifically, we investigate and exploit multiple video representations: vectors of locally aggregated descriptors, improved dense trajectories, optical flow, 3D ConvNet, and acoustic features. Two promising deep models are utilized, namely ResNet-152 and Inception-V3. An advanced feature integration algorithm termed multi-view intact space learning is used to produce the final video representation. Furthermore, we exploit an efficient two-stage strategy for the detection task. Our system obtains state-of-the-art performance for large-scale activity classification and detection.

1. Introduction

Activity classification [4, 9, 27] and detection [15, 26, 17, 27] in long, real-world video sequences are challenging but critical components of various vision applications including video surveillance, video summarization, and video retrieval. Algorithms must not only identify the activity in a video but also deduce when the activity occurs. There has been considerable progress in this area over the last few years, with data-driven feature extraction [24] largely outperforming conventional handcrafted features [21] for video classification or detection activities.

However, many datasets used for activity classification and detection contain trimmed video clips of only a few seconds - a much easier task than the untrimmed case. The ActivityNet [1] challenge provides an opportunity for the research community to design algorithms for the untrimmed video scenario. The dataset used in this challenge contains 200 classes, with each class including 100 untrimmed videos, each specifying on average 1.54 activity instances. These data present a major challenge for modern algorithms.

In this notebook paper, we investigate possible solutions for classification and detection tasks on these video data. Specifically, we study how performance is affected by ensembles of multiple video representations including vectors of locally aggregated descriptors (VLAD) [6], improved dense trajectories (IDT) [21], optical flow [16], 3D ConvNet [20], and acoustic features [11]. Considering that each representation has partial video information but multiple representations contain redundant video information, we investigate an advanced feature integration algorithm termed multi-view intact space learning [23], which produces a useful representation for each video. We also design an efficient two-stage detection strategy. Our system shows promising performance for both untrimmed video classification and activity detection.

2. Video Representations

To maximally extract useful information from videos for classification and detection, we employ five video representations as detailed below.

2.1. Vectors of locally aggregated descriptors

Each video frame provides static yet informative cues that are strongly associated with the video content. Image classification studies have shown that these cues are well captured by hierarchical extraction through deep convolutional neural networks (CNNs) [8]. To generate the video representation, average pooling on the frame-level CNN features provides fewer discriminative descriptors than pooling with VLAD [6]. Therefore, we use VLAD to generate the first of our video representations.

Specifically, we employ three CNN models to collect frame-level features: ResNet-152 [5] pre-trained on ImageNet [3], ResNet-152 [5] pre-trained on Places2 [28], and Inception-V3 [19] pre-trained on ImageNet [3]. Pre-training equips these models with high representation capacity. In ResNet-152, the res5c_relu layer of size $7 * 7 *$

2048 is used to extract features, while in Inception-V3, the output of the last inceptions sized $8 * 8 * 2048$ is used as the deep feature of a frame. For each model, the dimension of the extracted features is reduced to 1024 by principle component analysis (PCA). The VLAD encoding procedure is then applied to the dimension-reduced features. In this way, three VLADs are generated for a video. The number of the centers used in VLAD is set to 512.

2.2. Improved dense trajectories

Handcrafted features can provide semantic characteristics that are not revealed by the learned features. Therefore, these features can be used as complementary information to enhance the video representations. Specifically, we choose IDT [22], a state-of-the-art handcrafted feature. In IDT, histogram of oriented gradients (HOG), histogram of flow (HOF), and motion boundary histograms (MBH) are computed for each video as features. As in [22], Fisher Vector encoding [14] is further utilized to quantize the features and create a high-dimensional representation. This provides the second part of our video representations.

Of note, IDT suffers from the inconsistencies arising from human and camera motion, which may significantly affect homography estimation. This problem is often overcome by masking the human regions in videos using a human detector. Here we employ a state-of-the-art detector, Faster R-CNN [13], on top of the ResNet-152 model [5] pre-trained on the COCO dataset [10].

2.3. Optical flow

The relevance of consecutive frames is also informative for identifying video contents. Optical flow is widely used to reveal this information. To extract motion features from optical flow images, we again employ a deep CNN model to generate the fourth part of our video representations. Specifically, we choose the two-stream architecture [16] consisting of the spatial stream ConvNet and temporal stream ConvNet, which are designed for video classification. After principled learning of the whole model on UCF-101 [18], the well-trained temporal stream is used to extract representations from the optical flow. Referring to [16], we set the optical flow stacking depth to L=10 for optimal performance.

2.4. 3D ConvNet features

We use the 3D ConvNet architecture [20] to construct video features from the spatial and temporal dimensions in a unified manner. 3D ConvNet takes the whole video as input and outputs a powerful feature through a set of 3D convolutional and 3D pooling layers. In particular, we employ the superior architecture in [20], namely C3D, which simultaneously models the general appearances and motion information of activities. The output of the fully connected fc7 layers is used as the feature. The C3D model is already pre-trained on the Sports-1M dataset [7] so no fine-tuning is applied. Since C3D is designed to accept a 16-frame video clip, each video is segmented into clips of the correct size with an 8-frame overlap. The clips are sequentially fed into C3D to compute the features, whose dimensions are then reduced from 4096 to 500. The stacked features are used as the fourth part of our video representations.

2.5. Acoustic features

Acoustic features act as a weak enhancer for our video representations since they cannot provide enough discriminative information for activity classification and detection alone. However, when used in collaboration with the above representations, acoustic features are useful, particularly when a class of activities has specific audio information, such as sound of blowing hair or leaves. In addition, semantic information translated from the monologue or dialogue of the video can provide more accurate clues for action analysis. However, we have not utilized this information for the action classification task. Here we extract MFCC [11] from the audio signals and then quantize them based on BoWs with 4000 words. The resultant BoW vector is our final video representation.

3. Latent Intact Representation

The above representations describe different video instance views. Each view may only capture partial information, but together they capture redundant information about the instance. Hence, it is both valuable and necessary to integrate this multi-view information. For this purpose, we use the multi-view intact space learning (MISL) algorithm to extract the latent intact representation from the multi-view information [23]. Assume that each of the above representations is represented by $z_i^v \in \mathbb{R}^{D_v}$, where i is the sample index and v is the representation index. MISL assumes that the representations $\{z_i^v\}$ depend on a latent intact representation via a view generation function. A linear example is that

$$z_i^v = \mathrm{W}_v x_i + \epsilon_i^v, \tag{1}$$

where $x_i \in \mathbb{R}^d$ is a sample point in the latent intact space, X, $W_v \in \mathbb{R}^{(D_v \times d)}$ is the v-th view generation matrix, and ϵ_i^v is the view-dependent noise. MISL measures the reconstruction error over the latent intact space using the Cauchy loss:

$$\begin{aligned} &\frac{1}{mn}\sum_{v=1}^{m}\sum_{i=1}^{n}\log(1+\frac{\|z_i^v - w_v x_i\|^2}{c^2}) \\ &+C_1\sum_{v=1}^{m}\|W_v\|_F^2 + C_2\sum_{i=1}^{n}\|x_i\|_2^2, \end{aligned} \tag{2}$$

where c is a constant scale parameter, C_1 and C_2 are non-negative constants, and $m = 5$. The last two terms reg-

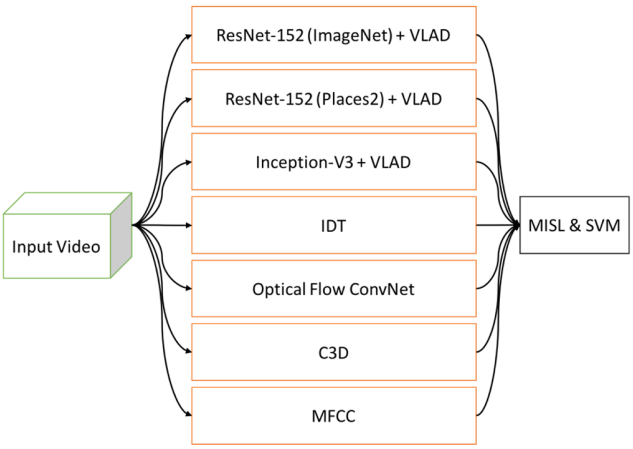


Figure 1. Framework of the proposed system for action classification.

ularize the respective quantities to avoid overfitting. Considering that there are a large number of training samples in this challenge, we use linear MISL. Following the learning process in [23], we optimize the parameters $\{W_v\}$ and the latent intact representations $\{x_i\}$ using the iteratively reweighted residuals algorithm. Once $\{W_v\}$ are trained properly, the inference process is as follows. Given a new data sample with the representations $\{z^1, \ldots, z^m\}$, the corresponding latent intact representation is obtained by solving

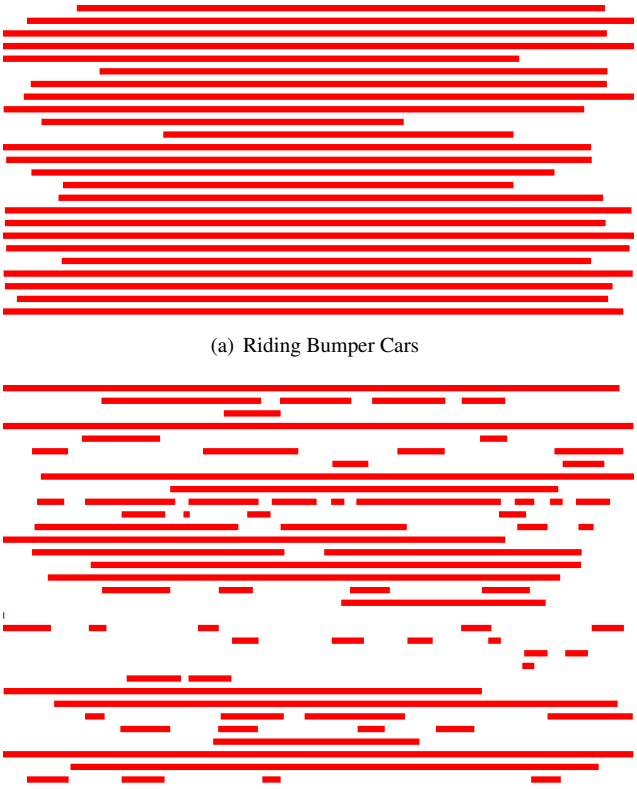
$$\min_x \frac{1}{m} \sum_{v=1}^{m} \log(1 + \frac{\|z^v - w_v^* x\|^2}{c^2}) + C_2 \|x\|_2^2 . \quad (3)$$

4. Untrimmed Video Classification

Figure 1 illustrates the framework of our proposed system for untrimmed video classification. No video preprocessing is performed, but instead we directly extract the above-mentioned five representations for each video. The representations are fused by finding the latent intact representation [23]. For multi-class classification, linear support vector machines (SVMs) [2] are used with the one-vs.-rest setting, meaning that each classifier separates the samples of one class against the samples of all other classes. Therefore, 200 classifiers are trained in total. The SVM parameter is set to C=10.

5. Activity Detection

Activity detection aims to identify the temporal location of an activity. As can be seen from Figure 2, different actions have different durations, which sometimes makes the annotation indefinite. For example, do we need to segment the action when view changes for a very short time or person changes by montage? Actually, it is easier to locate the action of Riding Bumper Cars than that of Playing Ten Pins. It is well known that the vast majority of short videos from Youtube record a particular action from beginning to end [25], which means the action tends to take place in the middle of the video, as indicated in Figure 3. In order to improve the mAP of action detection under the criterion of 0.5 IoU, we employ an effective pipeline for action detection on Youtube videos.



(a) Riding Bumper Cars

(b) Playing Ten Pins

Figure 2. Action annotations of two particular classes on the validation set. Each red segment indicates the action duration, which is normalized by the whole video. Detection of the first action (Riding Bumper Cars) is much easier than the second action (Playing Ten Pins).

We follow the detection pipeline shown in Figure 4. At first, we try to localize actions with high precision. Videos with single and long actions are the primary focus at this stage. Then, to improve the recall, we segment the videos with multiple short actions to generate more predictions. To perform classification in the detection pipeline, we must retrain the SVM classifiers on activity intervals rather than whole videos. For this, we crop activity instances from a video according to the ground truth intervals. The beginning and end of an instance are randomly located such that the instance interval and the ground truth interval have at least 0.7 IoU overlap. We use VLAD, IDT, and C3D as features in this task.

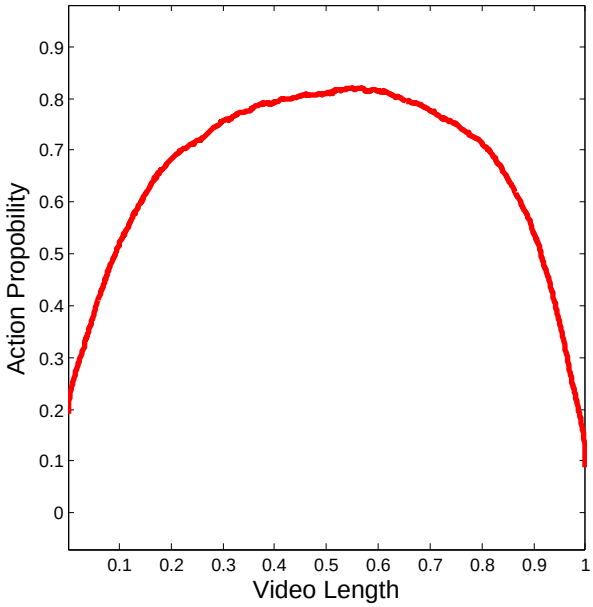


Figure 3. Action distribution on the validation set. Most of the actions take place in the middle of videos.

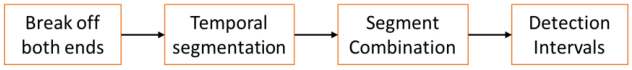


Figure 4. Detection pipeline.

5.1. Break off both ends

The irrelevant contents at the beginning and end of an activity video may decrease activity detection performance. Therefore, we first remove both ends of the video. Specifically, given a video, we uniformly divide it into 10 intervals. The representation is generated for each interval and then fed into the SVM classifiers of the ground truth video class. If the fusion of the SVM outputs is less than 0, the corresponding interval is regarded as irrelevant; otherwise, it is relevant. This is performed from the beginning until a relevant interval is found, with a similar operation applied to the end. In this way, the irrelevant beginning and end can be removed. At this stage, we pretend that there is only one action in the video and remove the irrelevant contents at the beginning and end of a video. However, we also classify each video into two classes: single action video and multi-action video with the scores on the 10 intervals. At the following stage, we focus on the multi-action videos to improve the recall.

5.2. Temporal segmentation

A cropped video is obtained after the first step. We next apply a sliding window (30 frame stride) over the cropped video covering 100 frames. When the window is located at a position, the covered frames are fed into the feature extraction process and then the SVM classifiers of the ground truth video class, thereby producing a decision score. As the window slides, a set of scores is generated to form a 1D video representation that can be used to reveal when the activities occur. We then apply the kernel temporal segmentation method [12] on the 1D representation to generate a set of temporal segments.

5.3. Segment combination

To obtain the final activity interval, we must judge whether adjacent segments should be merged into one interval. For this purpose, adjacent segments are concatenated into intervals during training. For example, 10 consecutive segments can be concatenated into $10 * 11/2 = 55$ intervals. If a generated interval and a ground truth interval have at least 0.7 IoU overlap, this interval is regarded as a positive sample; otherwise, it is a negative sample. We then train an SVM classifier on these samples.

We define the beginning segment of an activity as an anchor segment. During testing, given a set of temporal segments, the following operations are started from each of the anchor segments. To decide whether the i-th and $(i+1)$-th segments are merged, we generate the representation from the temporarily merged interval and use the SVM classifier to test. If a positive result is obtained, the merging is successful and we further test the merging with the (i+2)-th segment. If a negative result is obtained, the merging is unsuccessful, and we then proceed to test the following segments sequentially until a new anchor segment is found. The resulting intervals reveal the beginning and end of each activity in the video. We directly discard predictions less than 5% of the whole video in the detection task.

6. Experiments

When we test on the validation set, the SVM classifiers and MISL are trained on the training set. When the test set scores are predicted, both the training and validation sets are used to train MISL and SVM.

6.1. Classification results

We first investigate the performance of different combinations of the video representations on the validation set, as shown in Table 1. We also give the hardest and easiest class names based on the Top-1 error in Table 2. As shown in Figure 5, there are a small number of classes that tend to be confused with other classes, such as Long Jump and Triple Jump, Polishing Shoes and Cleaning Shoes, Mowing the Lawn and Cutting the Grass.

As shown in Table 3, we evaluate the proposed method five times on the test set. The first submission is based on VLADs+IDT. Then, the second and third submissions are based on VLADs+IDT+OF and VLADs+IDT+OF+C3D, respectively. Finally, the fourth and fifth submissions are based on VLADs+IDT+OF+C3D+MFCC.

Representations	mAP	Top-1
VLADs	0.879	0.832
VLADs+IDT	0.897	0.855
VLADs+IDT+OF	0.909	0.863
VLADs+IDT+OF+C3D	0.923	0.874
VLADs+IDT+OF+C3D+MFCC	0.932	0.881

Table 1. Action classification results on validation set

Hardest Class	Easiest class
Removing curlers	Tango
Long jump	Riding bumper cars
Triple jump	Preparing pasta
Rock-paper-scissors	Playing rubik cube
Volleyball	Playing ice hockey
Drinking beer	Scuba diving
Gargling mouthwash	Using the pommel horse
Washing face	Ping-pong
Kayaking	Using the rowing machine
Peeling potatoes	Rock climbing
Having an ice cream	BMX
Smoking a cigarette	Plastering
Shot put	Ice fishing
Painting furniture	Sailing
Installing carpet	Hitting a pinata
Drinking coffee	Welding
Applying sunscreen	Playing blackjack
Cutting the grass	Playing pool
Mowing the lawn	Windsurfing
Cleaning shoes	Arm wrestling
Putting on shoes	Elliptical trainer
Doing nails	Using the monkey bar
Hand washing clothes	Snow tubing

Table 2. The hardest and easiest classes based on the Top-1 classification error.

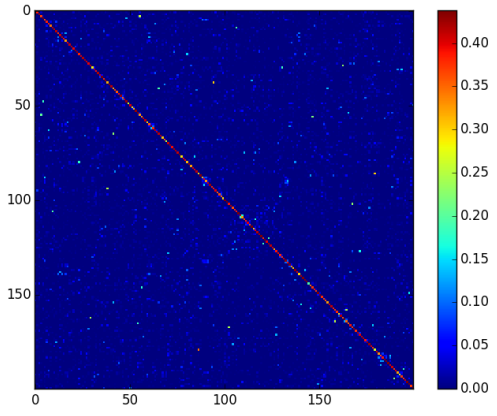


Figure 5. Confusion Matrix on validation set. Classes are sorted in alphabetical order.

Representations	mAP	Top-1	Top-3
submission1	0.89354	0.85726	0.95844
submission2	0.90992	0.86418	0.96524
submission3	0.91823	0.87291	0.96722
submission4	0.92286	0.87633	0.97044
submission5	0.92413	0.87792	0.97084

Table 3. Action classification results on test set

$> 50\%$	$> 40\%$	$> 30\%$	$> 20\%$	$> 10\%$
78.20%	83.20%	87.70%	92.20%	96.39%

Table 4. Action duration statistics from single action videos (75%) on the validation set.

	single trim	multi segment
mAP	39.76%	43.65%
recall	52.50%	62.24%

Table 5. Detection results of the proposed method on the validation set. The recall is evaluated by the criterion of 0.5 IoU.

6.2. Detection results

As shown in Table 4, 75% videos have only one action, and we give the statistics of the single-action videos on the validation set. 78.2% actions last for more than 50% of the video length in the single-action videos. We investigate the effectiveness of the proposed method on the validation set. We set the evaluation criterion as at least 0.5 IoU. As can be seen from Table 5, we obtain the mAP of 39.76% by predicting the beginning and the end of an action without temporal segmentation. The recall in this case is only 52.50%, but high precision can be preserved. To further improve the recall, we generate the action intervals by temporal segmentation and combination, and in this way, the mAP is improved to 43.65%. Finally, we evaluate the proposed method on the test set, obtaining the mAP of 42.478%.

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[2] C.-C. Chang and L. C. LIBSVM. a library for support vector machines, 2001. *Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm*, 2012.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016.

[6] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into

compact codes. *Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos. Vlad3: Encoding dynamics of deep features for action recognition. 2016.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[11] B. Logan et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[12] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[14] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[15] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. 2016.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[17] B. Singh and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. 2016.

[18] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. 2015.

[21] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, pages 1–20, 2015.

[22] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[23] C. Xu, D. Tao, and C. Xu. Multi-view intact space learning. *Pattern Analysis and Machine Intelligence*, 37(12):2531–2544, 2015.

[24] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, pages 1798–1807, 2015.

[25] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, pages 4633–4641, 2015.

[26] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. 2016.

[27] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. 2016.

[28] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places2: A large-scale database for scene understanding. Technical report, Technical report, 2015.

MSR Asia MSM at ActivityNet Challenge 2016

Zhaofan Qiu, Dong Li, Chuang Gan, Ting Yao*, Tao Mei, and Yong Rui
Microsoft Research, Beijing, China
{tiyao, tmei, yongrui}@microsoft.com

Abstract

This notebook paper presents overview and comparative analysis of our system designed for untrimmed video classification task in ActivityNet Challenge 2016. We investigate and exploit multiple spatio-temporal clues, i.e., frames, motion (optical flow), and short video clips, using 2D or 3D convolutional neural networks (CNNs). The mechanism of different quantization methods are studied as well. Furthermore, improved dense trajectory with fisher vector encoding on long video clips and MFCC audio features are utilized. All activities are classified by late fusing the predictions of one-versus-rest linear SVMs learnt on each clue. Finally, OCR is employed to refine the prediction scores.

1. Introduction

Recognizing activities in videos is a challenging task as video is an information-intensive media with complex variations. In particular, an activity may be represented by different clues including frames, motion (optical flow), short video clips, long video clips, audio and OCR. In this work, we aim at investigating these multiple clues to activity classification in videos.

The remaining sections are organized as follows. Section 2 describes our activity recognition system. Section 3 presents all the features, while Section 4 details feature quantization strategies. In Section 5, we provide empirical evaluations, followed by the conclusions in Section 6.

2. Recognition Framework

Our activity recognition framework is shown in Figure 1. In general, the untrimmed video classification process is composed of three stages, i.e., multi-stream feature extraction, feature quantization and prediction generation. For deep feature extraction, we follow the multi-stream approaches in [4, 6], which represent the input video by a hierarchical structure including individual frames, consecutive frames and short clips. In addition to deep features, two most complementary hand-crafted features, i.e., iDT and audio MFCC, are exploited to further enrich the video representations. After extraction of raw features, different quantization and pooling methods are utilized on different features to produce representations of each video. A linear SVM is trained on each kind of video representations and the predictions from multiple SVMs are combined by linearly fusion. When training SVM, we fix $C = 100$ for all the experiments. Finally, OCR is employed to refine the list of recognized videos for each activity.

3. Multi-Stream Features

In our framework, we extract the features from multiple clues including frames, motion, short clips, long clips, audio and OCR.

3.1. Frame

To extract frame-level representations from video, we first uniformly sample 50 frames from each video, and then use pre-trained/finetuned 2D CNNs as frame-level feature extractors. We choose three popular 2D CNNs in image classification: VGG [7], GoogLeNet [5, 9] and ResNet [1]. The performances between features extracted from different layers of different architectures will be discussed later.

3.2. Motion

To model the change of consecutive frames, we apply another CNNs to optical flow "image," which can extract motion features between consecutive frames. When extracting motion features, we follow the setting of [12], which fed 20 optical flow images, consisting of two-direction optical flow from 10 consecutive frames, into VGG_16 network in each iteration. We use VGG_16 model and sample rate is 50 per video, which means 50×20 optical flow "images" are considered for each video.

3.3. Short Clip

In addition to frames and motion between consecutive frames, we further exploit popular 3D CNN architecture, C3D [10], to construct video clip features from both spatial and temporal dimensions. The C3D model is pre-trained on

*Principal Designer.

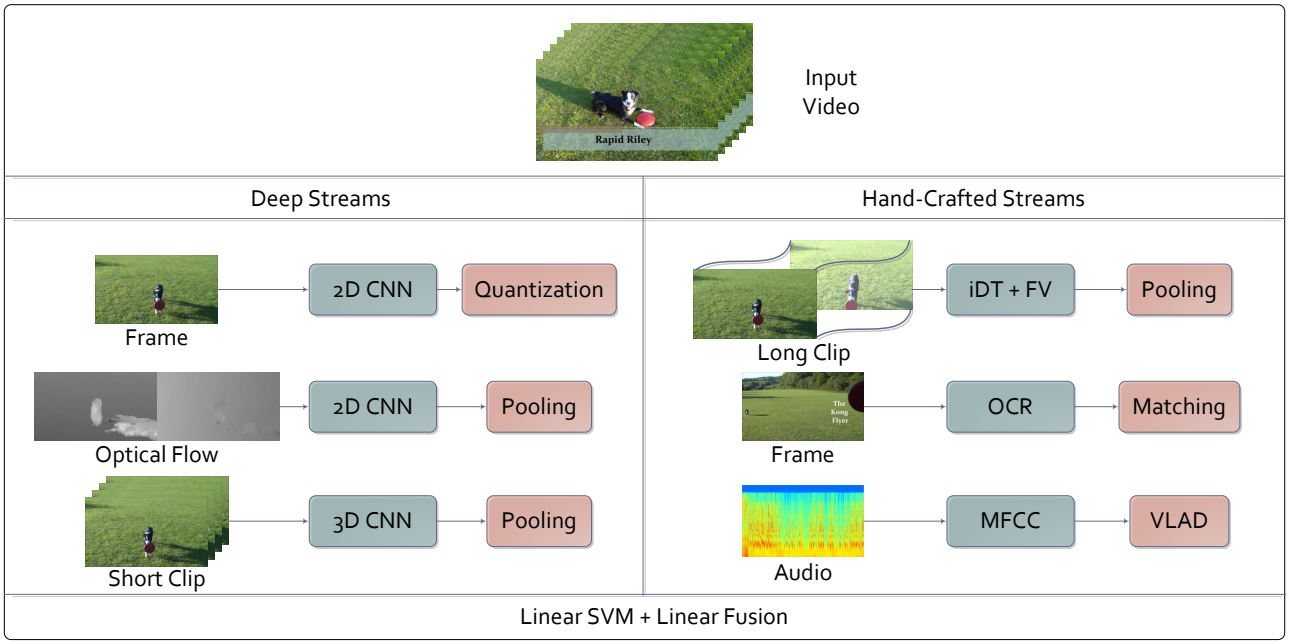


Figure 1. Framework of our proposed system.

Sports-1M dataset [3]. We fix the length of short clip to 16 frames, and sample rate is also 50 per video.

3.4. Long Clip

For long clips, we choose the state-of-the-art handcrafted features - improved dense trajectory (iDT) [11] on each sampled clip. Specifically, trajectory feature, histogram of oriented gradients (HOG), histogram of flow (HOF), and motion boundary histogram (MBH) are computed for each trajectory obtained by tracking points in video clips. Furthermore, fisher vector encoding is used to quantize the features and create high dimensional representations for each clip. Considering that the extraction of iDT is very time consuming, we split each video into a set of five-second clips evenly without any overlap.

3.5. Audio

For audio features, MFCC are extracted and exploited. As the duration of different videos are different, the counts of MFCC are also different.

3.6. OCR

Tesseract OCR [8] is used to extract text from video frames. We apply the detector on each frame from the whole video, followed by string matching with activity name. Before matching, we simplify the activity name by removing meaningless word, e.g. "doing the," and remove some misleading categories, e.g. "polo." Finally, we simply take the videos as positive samples if the activity name appears in the text of their frames.

4. Feature Quantization

In this section, we describe three quantization methods to generate video-level representations from frame-level or clip-level features.

4.1. Average Pooling

As shown in the Figure 1, we use average pooling upon the extracted features from consecutive frames, short clips and long clips. For a set of frame-level or clip-level features $F = \{f_1, f_2, ..., f_N\}$, the video-level representations are produced by simply averaging all the features in the set:

$$R_{pooling} = \frac{1}{N} \sum_{i:f_i \in F} f_i \ , \tag{1}$$

where $R_{pooling}$ denotes the final representations.

4.2. VLAD

Recently, Vectors of Locally Aggregated Descriptors (VLAD) [2] shows good ability on feature quantization. With K-means centers $C = \{c_1, c_2, ..., c_K\}$, video-level representations from VLAD can be described as:

$$\begin{aligned} u_k &= \sum_{i:NN(f_i)=c_k} (f_i - c_k) \\ R_{vlad} &= normalize(u) \end{aligned} \ , \tag{2}$$

where $NN(f_i)$ denotes f_i's nearest neighbor in C. We choose the variant of VLAD called VLAD-k, which replaces the nearest neighbor with k-nearest neighbors, and fix $k = 5$. For feature normalization, we choose power, l_2 and intra-normalization by default.

Table 1. Top1-accuracy of different 2D CNN architectures on ActivityNet validation set. The video feature are extracted on 50 sampled frames followed by average pooling.

Network	Fintuned	Layer	Top1
VGG_19		fc6	66.59%
GoogLeNet		pool5	68.76%
ResNet_152		pool5	**71.43**%
VGG_16	$\surd$	fc6	67.03%
GoogLeNet	$\surd$	pool5	68.57%
ResNet_50	$\surd$	pool5	68.43%
ResNet_152	$\surd$	pool5	**74.82**%

4.3. Deep Quantization

VLAD has two obvious weaknesses: (1) high computation and storage cost; (2) label information is ignored. Therefore, we present a novel network-based quantization method called Deep Quantization (DQ).

First, we train a generative neural network with parameters θ on the top of feature extraction network. Following the fisher kernel method, the video-level representations are defined as

$$\begin{aligned} L_{Generative}(\theta) &= \sum_{f \in TrainingSet} -log\, p(f,\theta) \\ \widehat{\theta} &= \arg\max_{\theta} L_{Generative}(\theta) \\ R_{DQ} &= normalize(\sum_{i:f_i \in F} \frac{\partial(-log\, p(f_i,\widehat{\theta}))}{\partial \widehat{\theta}}) \end{aligned} \quad , \quad (3)$$

where $p(f,\theta)$ is the generative network output. After optimizing parameters θ, the gradient calculation and accumulation can be processed in an end-to-end manner during backpropagation, and no extra storage is required. To further improve the ability of representations, we propose a semi-supervised optimizing function as:

$$\begin{aligned} L(\theta) &= L_{Generative}(\theta) + \lambda L_{Classification}(\theta) \\ \widehat{\theta} &= \arg\max_{\theta} L(\theta) \\ R_{DQ} &= normalize(\sum_{i:f_i \in F} \frac{\partial(-log\, p(f_i,\widehat{\theta}))}{\partial \widehat{\theta}}) \end{aligned} \quad . \quad (4)$$

The detailed description of our deep quantization network and more experimental analysis will be published on arXiv.org soon.

5. Experiment

5.1. 2D CNNs Comparison

Here we compare three popular 2D CNN architectures: VGG, GoogLeNet and ResNet. The comparison results on validation set are shown in Table 1.

The settings of 2D CNN are generally divided into two parts, i.e., "*pre-trained model + average pooling*" and "*finetuned model + average pooling*." All the four finetuned networks are initialized by pre-trained models, and finetuned on ActivityNet training set. We can observe that ResNet_152 achieves the highest accuracy among the three architectures and it will be further improved by finetuning.

Table 2. Top1-accuracy of different quantization methods on ActivityNet validation set. All the local feature are extracted from ResNet_152 architecture.

Method	Fintuned	Layer	Top1
Average Pooling	$\surd$	pool5	74.82%
Average Pooling		pool5	71.43%
VLAD		rec5c	76.70%
Deep Quantization		rec5c	**78.55**%

5.2. Quantization Comparison

Table 2 shows the results of different quantization methods on ResNet_152. We exploit VLAD and our Deep Quantization on the outputs of Res5c layer which is the last convolutional layer. It is worth noting that we only apply these two quantization methods on default ResNet_152 model. For VLAD, we first reduce the feature dimension to 1024 by PCA, and then apply k-means with $k = 256$, which means the dimension of representations for each video is 1024×256. For Deep Quantization, we set the number of hidden state to 128, making the feature dimension of 2048×128 in total.

It can be observed that VLAD obtains large performance improvement over Average Pooling method (76.70% vs 71.43%), which is even higher than finetuned model. Our proposed Deep Quantization model achieves better accuracy than VLAD (78.55% vs 76.70%), and it is the best setting of our 2D CNN.

5.3. Performance Comparison

Table 3 shows the performances of all the components in our submission and their fusion weights. The fusion weights are tuned using gradient descend on validation set by minimizing the classification loss. The OCR results are considered as post-processing and employed after linear fusion.

Overall, our Deep Quantization on ResNet_152 achieves the highest accuracy (78.55%) of single component, and it also obtains the highest fusion weight (24.9%). Although MFCC only gets 17.92% top1-accuracy, its fusion weight (19.1%) is the second highest due to the high complementarity between aural and visual features.

For the final submission, we train the SVMs using training and validation sets. All the components are fused using the weights tuned on validation set. Our final performance on test set is also shown on Table 3. Our top1-accuracy on test set is about 2% higher than validation set. This result basically indicates that more data used in training process may lead to higher recognition accuracy.

Table 3. Comparisons of different components in our framework on ActivityNet validation set. We also include the performance on ActivityNet test set from leader-board. Please note that there are two different settings of iDT, while the "Sample-10" means we randomly sample 10 long clips and average the predicting probabilities.

Stream	Feature	Fintuned	Layer	Quantization	Top1	Top3	MAP	Fusion Weights
Frame	VGG_19		fc6	Ave	66.59%	82.70%	70.22%	0.7%
	GoogLeNet		pool5	Ave	68.76%	84.73%	73.37%	1.2%
	ResNet_152		pool5	Ave	71.43%	86.45%	76.56%	0.6%
	VGG_16	$\surd$	fc6	Ave	67.03%	83.68%	70.12%	0.4%
	GoogLeNet	$\surd$	pool5	Ave	68.57%	85.26%	72.19%	1.4%
	ResNet_50	$\surd$	pool5	Ave	68.72%	86.13%	72.96%	8.4%
	ResNet_152	$\surd$	pool5	Ave	74.82%	87.59%	79.43%	6.3%
	ResNet_152		res5c	VLAD	76.70%	89.07%	81.52%	3.8%
	ResNet_152		res5c	DQ	78.55%	91.16%	84.09%	24.9%
Motion	VGG_16	$\surd$	fc6	Ave	49.05%	65.96%	49.06%	8.3%
Short Clip	C3D		fc6	Ave	65.80%	81.16%	67.68%	8.8%
Long Clip	iDT+FV			Ave	64.70%	77.98%	68.69%	14.3%
	iDT+FV			Sample-10	65.90%	80.15%	69.18%	1.8%
Audio	MFCC			VLAD	17.94%	26.10%	15.47%	19.1%
Fusion all					83.23%	94.24%	89.17%	
+OCR					84.26%	94.65%	90.03%	
On test set					86.68%	95.53%	91.93%	

6. Conclusion

In ActivityNet Challenge 2016, we mainly focused on multiple visual features and different strategies of feature quantization. The audio features can help classify some activities and OCR can be further employed to improve the accuracy. Our future works include the exploration of ASR and more in-depth studies of how fusion weights of different clues could be determined to boost the classification performance.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[2] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[4] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *ICMR*, 2016.

[5] P. Mettes, D. C. Koelma, and C. G. Snoek. The imagenet shuffle: Reorganized pre-training for video event detection. In *ICMR*, 2016.

[6] Z. Qiu, Q. Li, T. Yao, T. Mei, and Y. Rui. Msr asia msm at thumos challenge 2015. In *THUMOS'15 Action Recognition Challenge*, 2015.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[8] R. Smith. An overview of the tesseract ocr engine. In *ICDAR*, 2007.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *ICCV*, 2015.

[11] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[12] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

Recurrent 3D Convolutional Neural Networks for Untrimmed Video Classification

Xiaodong Yang* Pavlo Molchanov* Jan Kautz
NVIDIA
{xiaodongy,pmolchanov,jkautz}@nvidia.com

Abstract

In this paper, we present a method based on the recurrent three-dimensional convolutional neural works for action recognition in untrimmed long videos. This is vital for a number of real applications where videos are usually unconstrained and include extensive background scenes and activities. In order to address these challenges, we employ the connectionist temporal classification to train the networks to localize the most discriminative actions at the nucleus phase of unsegmented videos. Our proposed method is evaluated on the benchmark dataset of ActivityNet Challenge 2016.

1. Introduction

Content based video classification is fundamental to intelligent video analytics including automatic categorizing, searching, indexing, segmentation, and retrieval of videos. Conventional research primarily devotes to recognize actions in segmented short videos [4]. However, most user generated videos (e.g., videos from surveillance and internet portals) are with untrimmed long sequences which could contain unrelated activities and background scenes. So this work focuses on predicting the labels of activities present in long untrimmed videos.

2. Method

As illustrated in Fig. 1, we present a network that employs the recurrent three-dimensional convolutional neural networks (3D-CNN) with connectionist temporal classification (CTC). 3D-CNN is used to extract spatio-temporal features in a short-term window, and a recurrent layer is applied to model the long-term temporal evolution. CTC enables action classification to be based on the nucleus phase of video without requiring explicit pre-segmentation.

We initialize 3D-CNN with the C3D network [3] pretrained on the large-scale Sports1M action recognition dataset. This networks is consist of 8 convolutional layers with $3 \times 3 \times 3$ filters and 2 fully connected layers trained on 16-frame clips. CTC is a cost function designed for sequence prediction in unsegmented or weakly segmented input streams [1]. It has been successfully applied for online detection and classification of dynamic hand gestures [2]. CTC is applied in this work to identify and label the nucleus of an action, while assigning the *no action* class to the remaining clips. It is able to solve the alignment of class labels to particular clips in the video.

References

[1] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

[2] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks. In *CVPR*, 2016.

[3] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[4] X. Yang and Y. Tian. Action recognition using super sparse coding vector with spatio-temporal awareness. In *ECCV*, 2014.

UC Merced Submission to the ActivityNet Challenge 2016

Yi Zhu Shawn Newsam
University of California, Merced, USA
{yzhu25,snewsam}@ucmerced.edu

Zaikun Xu
University of Lugano, Switzerland
xuz@usi.ch

Abstract

This notebook paper describes our system for the untrimmed classification task in the ActivityNet challenge 2016. We investigate multiple state-of-the-art approaches for action recognition in long, untrimmed videos. We exploit hand-crafted motion boundary histogram features as well feature activations from deep networks such as VGG16, GoogLeNet, and C3D. These features are separately fed to linear, one-versus-rest support vector machine classifiers to produce confidence scores for each action class. These predictions are then fused along with the softmax scores of the recent ultra-deep ResNet-101 using weighted averaging.

1. Introduction

Human action recognition in video is a fundamental problem in computer vision due to its increasing importance for a range of applications such as video recommendation and search, video highlighting, video surveillance, human-robot interaction, human skill evaluation, etc.

The ActivityNet challenge [4] is a large scale benchmark designed to stimulate research on human activity understanding in user generated videos. This challenge consists of two tasks on 200 activity categories: (a) untrimmed classification and (b) detection. We focus on the former which involves predicting the activities present in a long video. Accounting for YouTube blocks and deleted videos, we downloaded 9942 training, 4874 validation, and 5001 test videos.

2. Recognition Framework

In this section, we present our multi-stream action recognition framework based on: (i) Fisher vector encoded MBH features, (ii) C3D fc7 features, (iii) GoogLeNet pool5 features, (iv) VGG16 pool5 features, and (v) ResNet-101 softmax scores. The first two modules are clip-based while the last three are frame-based. An overview of the framework can be found in Fig. 1.

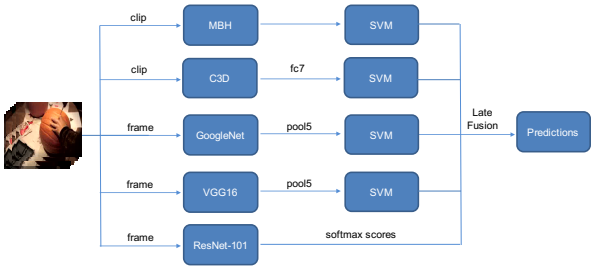


Figure 1. **Multi-stream framework**. We combine five modules using late fusion to obtain the final prediction scores. The MBH module is hand-crafted, while the rest are based on deep networks. For ResNet-101, we directly use the softmax scores since this performs better than using the extracted features.

2.1. MBH Features

Improved dense trajectories (IDT) [15] are state-of-the-art hand-crafted features for modeling temporal information in videos, and the motion boundary histogram (MBH) features are the best performing component of the IDT features. We use the provided[1] Fisher vector encoded MBH features [13, 9], whose dimension is 65536 for each video, to train a linear, one-versus-rest support vector machine (SVM) classifier. We fix the SVM hyper-parameter C to 100 [2].

2.2. C3D

In [14], the authors show that 2D ConvNets "forget" the temporal information in the input signal after each convolution. They therefore propose 3D ConvNets, which analyze sets of contiguous video frames organized as clips, and show its effectiveness at learning spatio-temporal features in video volume data analysis problems.

We therefore adopt fc7 features[2] extracted from a pre-trained C3D model as an additional signal. The network is not fine-tuned on the ActivityNet challenge dataset. The inputs to the C3D model are 16 frame clips with 50% overlap and the outputs are 4096 dimension feature activations.

[1]The MBH features are provided by the organizers.
[2]The C3D extracted fc7 features are provided by the organizers.

These features are reduced to 500 dimensions using PCA. Average pooling is used to combine the clip-level features for a single video. Finally, a linear, one-versus-rest SVM is trained with C set to 1.

2.3. GoogLeNet

We also use features[3] extracted from the pool5 layer of a Google inception net (GoogLeNet) [8]. This network is an enhanced version of [12] which utilizes a reorganized hierarchy of the complete ImageNet dataset [1]. The features are frame-based and have dimension 1024. They are mean-pooled across all frames in a video followed by L1-normalization to obtain a video-level representation. Again, a linear, one-versus-rest SVM is trained with C set to 1.

2.4. VGG16

VGG16 [11] is a popular deep architecture that demonstrated good performance on action recognition in [16] using a two-stream [10] pipeline. We only employ the spatial stream. We use the pre-trained VGG16 model for initialization and fine-tune it on the challenge dataset. During fine-tuning, we perform 60K iterations with learning rate 10^{-4}, 30K iterations with 10^{-5}, and 30K iterations with 10^{-6}. Momentum and weight decay are set to 0.9 and 5×10^{-4}.

We adopt the latent concept descriptor (LCD) encoding method in [17] to encode the pool5 layer of our fine-tuned VGG16 model, followed by VLAD encoding [6]. We reduce the dimensions of the pool5 features from 512 to 256 using PCA. The number of centers in VLAD encoding is set to 256 and we use VLAD-k with k set to 5. The encoded features are then power- and intra-normalized. The resulting per-frame features have dimension 65536 which are mean-pooled to obtain a video-level representation. A linear, one-versus-rest SVM is trained with C set to 1.

2.5. ResNet-101

Residual learning [3] has recently become an effective method to construct ultra-deep networks for object recognition and detection. We extend it here to action recognition. We adopt the pre-trained 101-layer model for initialization and fine-tune it on the ActivityNet video data. The learning rate is 10^{-4} for the first two epochs, 10^{-5} for the following two epochs, and 10^{-6} for the last epoch. Momentum and weight decay are set to 0.9 and 10^{-4}.

We also investigated using features extracted from last convolutional module, whose dimension is 2048, to train an SVM, similar to our other modules. This, however, performs 3.3% worse on the validation set than using the softmax scores.

[3]The GoogLeNet extracted pool5 features are provided by the organizers.

Model	Top-1 Accuracy
(i) MBH	57.32%
(ii) C3D fc7	60.04%
(iii) GoogLeNet pool5	67.13%
(iv) VGG16* pool5	63.19%
(v) ResNet-101*	71.81%
(i) + (ii)	62.78%
(i) + (iii)	69.40%
(i) + (iv)	68.79%
(ii) + (iii)	68.11%
(ii) + (iv)	64.35%
(iii) + (iv)	68.56%
(ii) + (iii) + (iv)	69.09%
(i) + (v)	73.05%
(ii) + (iii) + (iv) + (v)	73.56%
(i) + (iii) + (iv) + (v)	74.68%
(i) + (ii) + (iii) + (iv) + (v)	75.14%

Table 1. Action recognition results on the validation set of the ActivityNet challenge 2016. All performances are reported using top-1 accuracy. Top: Single module performances. Bottom: Fused module performances. * indicates the network is fine-tuned on the challenge dataset.

3. Experiment Results

Given a test video, we uniformly sample 25 frames to extract the frame-level feature activations and perform mean-pooling to obtain the final video representation.

Late fusion iteratively combines pairs of prediction scores. First, the outputs of two modules are combined in a weighted fashion where the scores of the more accurate module are weighted twice that of the less accurate one. Additional scores are then combined with this in a similar fashion. After late fusion, we adopt a Multi-class Iterative Re-ranking (MIR) method [7] to re-rank the predictions of classifiers based on the difficulty scale of the videos. Table 1 shows our experimental results on the validation set of the ActivityNet challenge 2016.

We can see from the top part of Table 1 that the residual network achieves the best performance among all modules. It is 14.5% better than the state-of-the-art hand-crafted MBH features and outperforms the other deep networks.

The bottom part of Table 1 shows the performances of various module combinations. We observe that combinations that only include deep networks are generally not as effective as combinations that include the MBH features. Although the MBH features perform the worst alone, they are orthogonal to the deep learning based approaches. This may be attributed to MBH being effective at capturing low-level motion features while the deep networks model high-

Submission	mAP	Top-1 Accuracy	Top-3 Accuracy
Run 1	68.00%	66.16%	83.36%
Run 2	75.98%	72.48%	87.54%
Run 3	79.41%	76.17%	90.19%
Run 4	81.64%	77.74%	90.93%
Run 5	83.1%	78.44%	91.07%

Table 2. Action recognition results on the test set of the ActivityNet challenge 2016.

level information related to static appearance. The MBH features and the deep networks are thus quite complementary. When fusing all modules, our system achieves a validation accuracy of 75.14%.

We also investigate incorporating action proposals generated by [5] during prediction. Instead of uniformly sampling 25 frames across the video, we sample 25 frames from the action proposals. The intuition is that these action proposals have a higher probability of containing action frames, so that the average pooling of these frames should lead to higher recognition accuracy. However, this turns out to perform worse than uniform sampling.

4. Submission Details

We use both the training and validation data as the training set for our submissions. Note, though, that the implementation details and parameter settings remain the same as when we use only the training data to train. We do not use the test data for training or parameter tuning.

We submit five runs to the evaluation server, and the performance for each run is shown in Table 2. Our runs are as follows:

- Run 1: VGG16
- Run 2: VGG16 + MBH
- Run 3: VGG16 + MBH + ResNet-101
- Run 4: VGG16 + MBH + ResNet-101 + GoogLeNet
- Run 5: VGG16 + MBH + ResNet-101 + GoogLeNet + C3D

5. Conclusion

We show that the ultra-deep architecture of ResNet is indeed helpful in learning discriminative features for complex tasks, such as human activity understanding. In addition, although hand-crafted MBH features achieve the lowest accuracy alone, they are complementary to approaches based on deep networks. Finally, the combination of all modules using late fusion gives the best performance.

Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation through the donation of the Titan X GPU used in this work. This work was funded in part by a National Science Foundation CAREER grant, #IIS-1150115.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 2008. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 2

[4] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 2015. 1

[5] F. C. Heilbron, J. C. Niebles, and B. Ghanem. Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos. In *CVPR*, 2016. 3

[6] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating Local Image Descriptors into Compact Codes. *TPAMI*, 2012. 2

[7] Z. Lan, S.-I. Yu, and A. G. Hauptmann. Improving Human Activity Recognition Through Ranking and Re-ranking. *arXiv preprint arXiv:1512.03740*, 2015. 2

[8] P. Mettes, D. Koelma, and C. G. M. Snoek. The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection. In *ICMR*, New York, USA, 2016. 2

[9] D. Oneata, J. Verbeek, and C. Schmid. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In *ICCV*, 2013. 1

[10] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. *NIPS*, 2014. 2

[11] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 2

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper With Convolutions. In *CVPR*, 2015. 2

[13] J. Snchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *IJCV*, 2013. 1

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 2015. 1

[15] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 1

[16] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards Good Practices for Very Deep Two-Stream ConvNets. *arXiv preprint arXiv:1507.02159*, 2015. 2

[17] Z. Xu, Y. Yang, and A. G. Hauptmann. A Discriminative CNN Video Representation for Event Detection. In *CVPR*, 2015. 2