ADSC Submission at THUMOS Challenge 2015

Jun Yuan[2*], Yong Pei[1*], Bingbing Ni[1], Pierre Moulin[3] and Ashraf Kassim[2]

[1]Advanced Digital Sciences Center, Singapore 138632
[2]Dept. of Electrical & Computer Engineering, National University of Singapore
[3]3University of Illinois Urbana-Champaign, IL 61820-5711 USA

yuanjunsg@gmail.com, {Pei.yong, bingbing.ni}@adsc.com.sg, moulin@ifp.uiuc.edu, ashraf@nus.edu.sg

Abstract. This notebook paper describes our approaches for the action recognition and temporal localization tasks of the THUMOS Challenge 2015. For the action recognition task, we use the subsequence-score distribution (SSD) framework. We use the Improved Fisher Vectors (IFVs) encoding of the Improved Dense Trajectories (IDTs) to capture motion, as well as a VGG-16 deep net model to extract 4096 dimension feature vector to capture the context information. A linear SVM is trained for classification of 101 categories' action video clips. For the temporal localization task, we use the IFV encoding at 9 different temporal scales, and apply the above SVM to obtain a pyramid score descriptor. The score features are used for generating action labels at frame level, and by proper post processing we are able to detect the 20 class actions in given videos.

Keywords: Action recognition, temporal localization, dense trajectories, deep net features, subsequence score distribution, temporal pyramid score descriptor

1 Motion and Scene Features

For motion features, we use the Improved Dense Trajectories (IDTs) from [2]. It contains several local descriptors (HOG, HOF and MBH) computed along the IDTs. This feature descriptor has achieved great performance in action recognition field in recent years. We first apply PCA on these local descriptors and reduce the dimensionality by a factor of two, and use Improved Fisher Vector (IFV) encoding to get the descriptors for video segments. We use 100,000 randomly selected trajectories from the UCF-101 dataset [6] to generate the projection matrix and dictionary. Each video segment is represented by a 2DK dimensional IFV, where D is the dimension of descriptors after projection, and K is the number of dictionary clusters (K = 256). The dictionary is further shared across the training, background, validation and testing set of the THUMOS'15 Challenge [7]. We use VLFeat implementation [8] for IFV encoding.

For scene features, we use a 4096 dimensional feature vector from each video frame using the convolutional neural network. We fine-tuned the VGG-16 model [3] on the fully connected layers, and use the outputs from the last rectified linear layer as features. The MatConvNet implementation [9] is used for scene feature extraction.

2 The Action Recognition Task

2.1 Subsequence Generation

We apply the shot boundary detection to each input video to produce subsequence video clips. The shot boundary proposal via HOG from [4] and colour histogram-based shot boundary detection algorithm in [5] are used.

---

*Denotes equal contribution

The shot boundary detection is applied on validation and test videos. Each input video is divided into small intervals with shot boundaries, and the adjacent intervals are concatenated into subsequences. For example, 10 consecutive video segments can be concatenated into 10*11/2 = 55 possible subsequences. Afterwards, we use IFV encoding on motion features. Each subsequence has a 10K dimensional IFV and a 4K dimensional scene vector.

2.2 Motion Features

We train a 1-vs-rest linear SVM with C =100 on UCF-101 training samples and THUMOS'15 Background Set. The background dataset serves as hard negative samples. The classifier is then applied on all subsequences of validation and test videos to get class scores, which are further sorted and L2-normalized. We use the normalized sorted scores as the final motion descriptor. We use the LIBLINEAR implementation for classifier training across all tasks [10].

2.3 Scene Features

After we get the sorted scores (motion feature) on validation data, we pick the subsequences with the highest score as the relevant subsequences of 101 actions. The combination of UCF-101 dataset and the relevant subsequences are used to train the base classifier of scene features.

We apply the above classifier on all subsequences of validation and test videos, sort the scores in descending order, and L2 normalize the sorted scores. We use final normalized sorted scores as scene descriptors.

2.4 Experiments

We combine the final motion descriptors and scene descriptors for the classification task. A linear SVM with C = 100 on validation data is trained and applied to test data to obtain the final scores.

3 The Temporal Localization Task

We adopt the following action detection pipeline in Fig 3.1 to produce the video segments that contain the 20 action classes. We only use motion features in the temporal action localization task, since the contextual information is likely to cause confusion to the detectors.

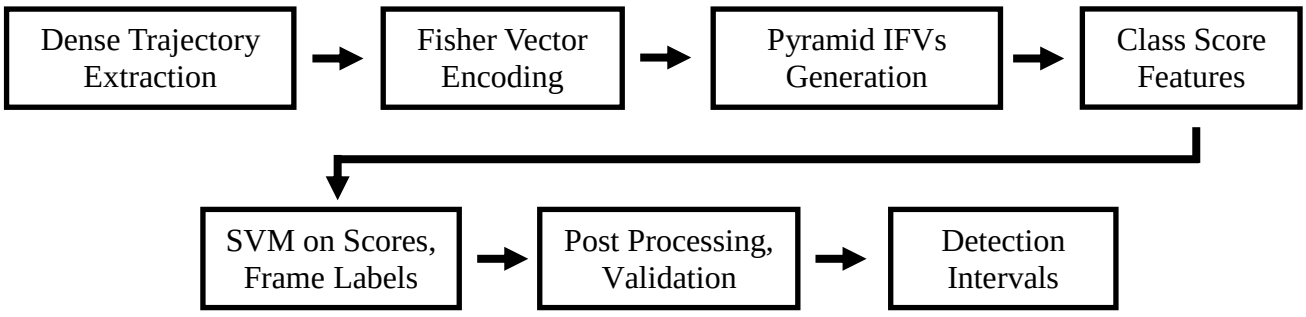

Fig 3.1 Detection Pipeline

3.1. Dense Trajectory Extraction

This process is the same as the Action Recognition Task in Part I.

3.2. Fisher Vector Encoding

The trajectory features are encoded using the FVs for every 5 consecutive frames. The sliding windows are not overlapped. We use the same projection matrix and dictionary as in part I, and the FVs are not normalized to retain its additivity. This is particularly useful for fast feature generation in step 3.

3.3. Pyramid IFVs Generation

The FVs from step 2 are re-combined and normalized to produce the Improved Fisher Vectors (IFVs) at different temporal resolutions. Specifically, we use the window boundaries as anchor frames and add up its neighboring FVs from resolution 10 to 90. For example, if the frame 1001 is an anchor frame, the FVs from 996 - 1005, 991 - 1010, ... , 955 - 1045 are summed up and renormalized to produce IFVs. The next anchor frame 1006, 1011, ..., are processed with the same.

3.4. Class Score Features

The IFVs are fed into the 101-class SVMs in Part I to obtain a 101-dimensional class scores. Thus, each anchor frame has 9*101 scores as new features. The score feature vectors are further L2-normalized to length 1.

3.5. Score Classifier, Frame Labels

We train a new 1-vs-rest SVM classifier with the score features on the validation set. We split the videos that contain the actions from the 20 classes into three splits, two for training and the rest one for testing. An addition class, namely the “background” class is used to distinguish action from backgrounds. This classifier is applied to all the test videos. Each anchor frame outputs a label from the 21 classes, and the background is discarded.

3.6. Post Processing, Validation

If an anchor frame contains only few trajectories, it is reset to background. Afterwards, we use median filters on the output labels to suppress the spikes. The video segments that are too short are also discarded.

Further, for some runs we combine the temporal localization results with classification labels from Part I to produce the final detection intervals. The detected video segments that do not correspond to top-1 or top-3 (in our different runs) classes are treated as false alarms. Thus, the final detection intervals are produced.

References

1. M. Hoai, A. Zisserman. *Improving Human Action Recognition Using Score Distribution and Ranking*. Asian Conference on Computer Vision, 2014
2. Wang, H., Schmid, C. *Action Recognition with Improved Trajectories*. International Conference on Computer Vision (Dec 2013)
3. Ken Chatfield, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman. *Return of the Devil in the Details: Delving Deep into Convolutional Nets*. BMVC 2014
4. M. Hoai, A. Zisserman. *Thread-Safe: Towards Recognizing Human Actions Across Shot Boundaries*. Asian Conference on Computer Vision, 2014
5. Rainer L. *Comparision of Automatic Shot Boundary Detection Algorithms*. Storage and Retrieval for Image and Video Databases VII, (17 December 1998)
6. Soomro, K. and Roshan Zamir, A. and Shah, M. *UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild*. CRCV-TR-12-01, 2012
7. Gorban, A. and Idrees, H. and Jiang, Y.-G. and Roshan Zamir, A. and Laptev, I. and Shah, M. and Sukthankar, R. *THUMOS Challenge: Action Recognition with a Large Number of Classes*. http://www.thumos.info/, 2015
8. A. Vedaldi and B. Fulkerson. *An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/, 2008
9. A. Vedaldi and K. Lenc. *MatConvNet -- Convolutional Neural Networks for MATLAB*. CoRR, abs/1412.4564, 2014
10. 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 9(2008), 1871-1874.