Weakly Supervised Temporal Action Detection with Shot-Based Temporal Pooling Network

Haisheng Su[1], Xu Zhao[1(✉)], Tianwei Lin[1], and Haiping Fei[2]

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai, China
[2] Industrial Internet Innovation Center (Shanghai) Co., Ltd., Shanghai, China
{suhaisheng,zhaoxu,wzmsltw}@sjtu.edu.cn, feihaiping@3in.org

Abstract. Weakly supervised temporal action detection in untrimmed videos is an important yet challenging task, where only video-level class labels are available for temporally locating actions in the videos during training. In this paper, we propose a novel architecture for this task. Specifically, we put forward an effective shot-based sampling method aiming at generating a more simplified but representative feature sequence for action detection, instead of using uniform sampling which causes extremely irrelevant frames retained. Furthermore, in order to distinguish action instances existing in the videos, we design a multi-stage Temporal Pooling Network (TPN) for the purposes of predicting video categories and localizing class-specific action instances respectively. Experiments conducted on THUMOS14 dataset confirm that our method outperforms other state-of-the-art weakly supervised approaches.

Keywords: Temporal action detection · Weak supervision
Shot-based sampling · Temporal pooling network · Class-specific

1 Introduction

Recently, impressive progress has been achieved on video analysis, which motivates two important tasks: action recognition and temporal action detection. Action recognition [1–4] is a crucial problem for video understanding which aims to classify manually trimmed videos. However, temporal action detection is more challenging since it not only deals with the classification of action categories in long untrimmed videos, but also localizes the boundaries of action instances. It can be applied in many areas such as smart surveillance and security system.

There are many deep learning based works focusing on the task of temporal action detection [5–9]. Most of them are performed with full supervision which relies on the temporal annotations of action instances greatly. However, it is time-consuming to annotate the temporal boundaries of each action instance for

This research has been supported by NSFC Program (61673269, 61273285).

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11304, pp. 426–436, 2018.
https://doi.org/10.1007/978-3-030-04212-7_37

a large-scale dataset. Furthermore, due to the subjective judgement, the annotating results can vary from person to person. Therefore, weakly supervised action detection draws attention of many researchers. In order to effectively locate the action instances in the videos using only video-level class labels during training, there are two crucial points: (1) remove considerable amount of irrelevant frames existing in the untrimmed videos; (2) generate high-quality detections especially in the videos containing multiple action instances of various classes.

A long and untrimmed video usually comes up with extremely irrelevant information, where action instances only occupy small parts, thus a sampling measure would contribute to removing the irrelevant frames and accelerating the speed of action detection. Uniform sampling method is widely used [10,11], however, it fails to utilize action structure information. Besides, traditional snippet-based classifiers rely on discriminative parts of actions greatly. Based on these two issues, we propose a shot-based sampling method to sample the input visual feature sequence generated by two-stream network [12], which can generate a more simplified and representative feature sequence for action detection.

How to generate proposals from a video sequence is another difficult problem for action detection under weak supervision. Bottom-up mechanism adopted in [13] first selects a set of clips as proposals, which are further classified, then the classification results are merged to match the video-level class labels. However, it fails to distinguish multiple action instances from each other existing in the video. We propose temporal pooling network to detect the class-specific discriminative frames of a given video in a top-down way, by means of computing one dimensional weighted Temporal Class Activation Maps (T-CAM). Besides, a novel attention module is designed to modulate the video representations, which can highlight the salient frames while suppressing the irrelevant counter-parts. To the best of our knowledge, we are the first to extend the class activation mapping [14] used for discriminative localization to the temporal domain, and the attention module designed for temporal attention weights learning is unique to our work. In sum, our main contributions include: (i) video representative parts selection using an effective shot-based sampling method; (ii) a top-down temporal pooling network for weakly supervised temporal action detection; (iii) extensive experiments reveal the good performance of our method.

2 Our Approach

In this section, we introduce the technical details of our approach. The framework is illustrated in Fig. 1.

2.1 Two-Stream Network for Feature Extraction

We adopt pretrained multiple two-stream network [13] to extract video feature representations, since this kind of architecture has shown great performance in action recognition task. The architecture of each two-stream network contains two branches: spatial network handles a single RGB frame and temporal network

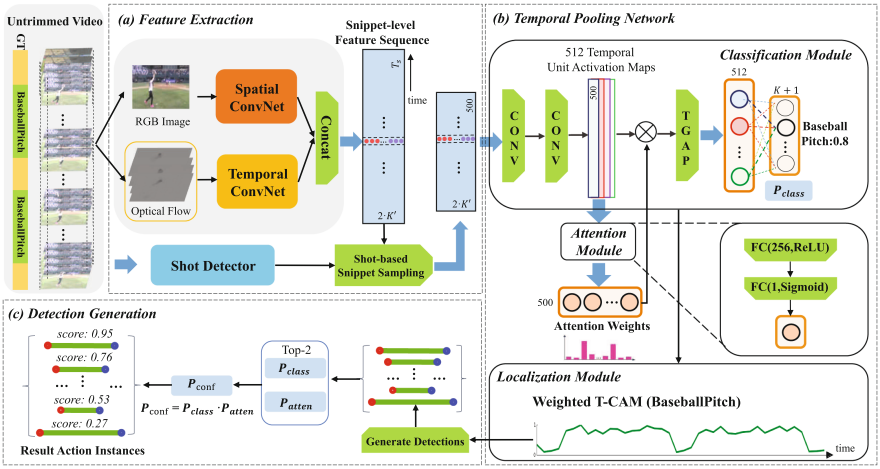

Fig. 1. Framework of our approach. (a) Two-stream network is used to extract visual features in snippet-level, then we adopt shot-based sampling strategy to sample the input feature sequence. (b) The architecture of temporal pooling network: *classification module* handles the sampled feature sequence for action classification; *attention module* learns attention weights for each sampled snippet; *localization module* generates weighted temporal class activation maps (T-CAM) to locate actions in temporal domain. (c) Detection generation: during prediction, we choose top-2 prediction results and group consecutive snippets with high activations to form the final detections

takes optical flow stacking of 5 frames as input. We compute optical flow using GPU implementation of [15] from the OpenCV toolbox.

Denote a given video $V = \{x_n\}_{n=1}^{T_v}$ consists of T_v frames, to extract the video features, we divide the video into $T_s = T_v/\sigma$ consecutive video snippets, where σ is the frame number of a snippet. A snippet is represented as $s = \{x_n\}_{n=x_f}^{x_f+\sigma}$, where x_f is the starting frame, $x_f + \sigma$ is the ending frame. Each snippet is independent without overlapping with each other and is processed by pretrained two-stream network respectively. Then we concatenate output scores in fc-action layer of two-stream network including both RGB and flow modalities of all snippets to form the feature sequence $F = \{f_{s_i} = \{f_{S,s_i}, f_{T,s_i}\}\}_{i=1}^{T_s}$, where f_{S,s_i} and f_{T,s_i} are output score vector of spatial and temporal network respectively with length K', where K' includes K action categories and one background category. This feature sequence is then fed to the shot-based temporal pooling network.

2.2 Shot-Based Sampling Method

We claim that the speed of temporal action detection task can be accelerated using a subset of discriminative snippets in a video. In order to remove the irrelevant frames and reduce the computational cost, we sample the input feature sequence instead of using all snippets for video classification. However, different sampling methods are bound to result in various classification performance.

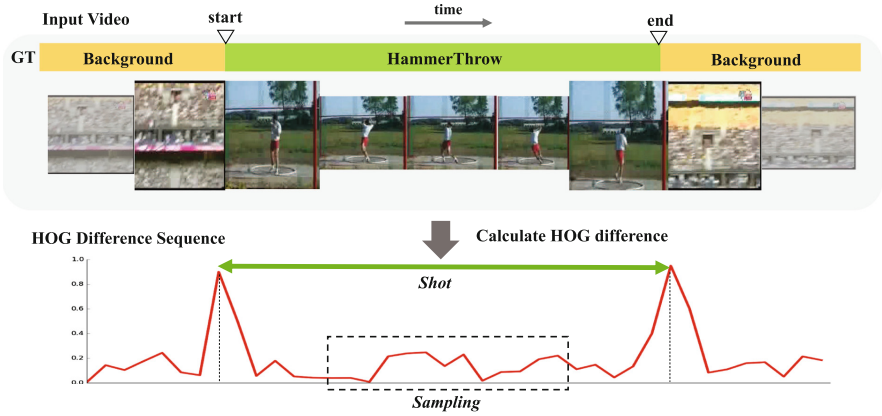

Fig. 2. A qualitative example of shot generated on THUMOS14

Generally, an effective sampling method should possess the ability to generate a simplified sequence which can contain the discriminative parts of action instances within the long untrimmed video. We propose the **shot-based snippet sampling method** that firstly detects the shot changes of the video based on the difference of histogram of oriented gradient (HOG) features [16] between two adjacent frames. An action shot will be detected if the absolute value of this difference is higher than a threshold. Each shot is denoted by (s_i^b, s_i^e), where s_i^b, s_i^e represent the beginning and ending of each action shot respectively. A qualitative example on THUMOS14 dataset is shown in Fig. 2. It should be noted that the shots are generated unsupervisedly.

Since the definition of the action boundaries is usually vague, we sample $\frac{N_t}{N_{shot}(V_j)}$ snippets around the middle of each shot as key frames to compose a new snippets sequence $S' = \left\{s_i'\right\}_{i=1}^{N_t}$, where $N_{shot}(V_j)$ denotes the number of shots generated in the video V_j and N_t is the number of sampled snippets in total used for training phase and we set it as 500 empirically. Finally, the sampled feature sequence can be represented as $F' = \left\{f_{s_i'}\right\}_{i=1}^{N_t}$.

2.3 Temporal Pooling Network

Inspired by the idea using CAM [14] for discriminative parts localization of images in CNN, we extend it to the temporal domain and propose a novel multi-stage architecture for weakly supervised temporal action detection. We use the sampled feature sequence as the input of Temporal Pooling Network (TPN). The architecture of TPN mainly contains three sub-modules listed as follows.

Classification Module. This module aims to classify the categories of each input untrimmed video based on the sampled feature sequence. It begins with

two temporal convolutional layers which directly handle the sampled feature sequence. These two layers have the same configurations: kernel size 3, stride 1, and 512 convolutional filters with ReLU activation function. Refer to the global average pooling proposed in [17], we extend it to the temporal domain and use temporal global average pooling (T-GAP) to encode video-level representation, then the 512 temporal pooled features are fed to the classification layer, which outputs the classification result with Sigmoid activation. Finally, the prediction result can be represented as a $K^{'}$-dimensional score vector $p_{class} = \{p^{c}_{class}\}^{K^{'}}_{c=1}$.

Attention Module. In this weakly supervised learning system, neither the ground-truth action categories of each snippet nor the ground-truth of the confidence map is provided. We therefore propose an indirect method to learn an importance weight for each sampled snippet feature representation with only weak-labels, thus to further highlight the snippet relevant to the actions as a soft selection. Concretely, we adopt a multilayer perceptron model with one hidden layer to learn the attention weights. Each snippet feature representation in the last convolutional layer is fed to the attention module respectively which consists of two fully connected layers with ReLU and Sigmoid activation separately. Then the temporal feature maps combined with the generated attention weights $\lambda = \{\lambda_i\}^{N_t}_{i=1}$ are followed by T-GAP to aggregate the video-level representation $\overline{R} = \sum^{N_t}_{i=1} \lambda_i r_i$, where r_i denotes the i^{th} snippet feature map in the last convolutional layer. Note that the temporal attention weights are trained in a class-agnostic way.

Localization Module. The goal of this module is to temporally identify the discriminative parts of the video which contribute most to the classification results. Refer to CAM in [14], we derive one-dimensional temporal class activation mapping (T-CAM) combined with temporal attention weights. We denote $w^c(z)$ as the z^{th} element of the weight matrix $W \in \mathbb{R}^{Z \times K^{'}}$ in the classification layer corresponding to class c. The input to the final sigmoid layer for class c is

$$s^c = \sum_{z=1}^{Z} w^c(z)\overline{R}(z) = \sum_{z=1}^{Z} w^c(z) \sum_{i=1}^{N_t} \lambda_i r_i(z) = \sum_{i=1}^{N_t} \lambda_i \sum_{z=1}^{Z} w^c(z) r_i(z). \quad (1)$$

Finally, we define the weighted T-CAM for class c of length N_t as $M^c_i = \lambda_i \sum^{Z}_{z=1} w^c(z) r_i(z)$. The weighted T-CAM describes the probability of each snippet being in a specific action class of the video.

2.4 Training Procedure

Using the simplified feature sequence generated by the shot-based sampling method, we train a classification model for the K action categories as well as background. The training data is constructed as $\Omega_{class} = \left\{(F^{'}(V_j), y_j)\right\}^{M}_{j=1}$,

where y_j denotes the ground-truth class label of video V_j. To train the classification module, we combine the multi-label cross-entropy loss and l_2 regularization loss to form the loss function:

$$L_{class} = \sum_{j=1}^{M} y_{label}^{j} log y_{pred}^{j} + \lambda \cdot L_2(\Theta_{class}), \tag{2}$$

where y_{pred}^{j} and y_{label}^{j} are predicted results and ground-truth video-level class labels of video V_j respectively, M is the number of training videos. λ balances the cross-entropy loss and l_2 regularization loss, and Θ_{class} is the classification module. We set $\lambda = 10^{-4}$ empirically. As for parameters in SGD, we train the model 300 epochs and the learning rate decays from 10^{-3} to 10^{-4} after 50 epochs. The batch size is set to 1.

2.5 Detection Generation and Prediction

During prediction, we use uniform sampling method to sample the input feature sequence and group consecutive snippets with activations higher than threshold θ among the weighted T-CAM into candidate detections with varied durations. Through empirical validation, we set θ as 5% of the maximum score. For a better observation of the localization ability of our model, we use top-2 video-level classification results of [13]. By combining the predicted action class score p_{class} and proposal attention weight p_{atten}, we can get a confidence p_{conf} for each proposal:

$$p_{conf} = p_{class} \cdot p_{atten}, \tag{3}$$

$$p_{atten} = \frac{\Sigma_{i=t_{start}}^{t_{end}} ReLU(M_i^c)}{t_{end} - t_{start} + 1}, \tag{4}$$

where p_{atten} is weighted mean T-CAM followed by a ReLU of all the snippets within the proposal durations denoted by $[t_{start}, t_{end}]$.

3 Experiments

3.1 Dataset and Setup

We evaluate our STPN on THUMOS14 dataset [18] for temporal action detection task. The THUMOS14 dataset contains 1010 and 1574 untrimmed videos for validation and testing respectively, while only 200 and 213 videos are temporal annotated among 20 action categories. Each video can contain multiple action instances of various classes. We train our model over 200 untrimmed videos on validation set and use 213 temporally annotated videos on the testing set to evaluate action detection performance. It should be noted that we don't use any temporal action instance annotations during training phase.

For temporal action detection task, we use mean Average Precision (mAP) as evaluation metric on this dataset. A predicted result instance is regarded as

correct only when it gets the correct category label and its intersection over union (IoU) with the ground-truth instance is higher than θ.

Parameters used in each module have been given before. We extract features from pretrained two-stream network using Caffe [19]. And we implement STPN using TensorFlow [20].

3.2 Evaluation on Visual Feature Encoder

Visual encoders are used to extract snippet-level features. To study the contributions of different visual encoders, we evaluate them individually and coherently with the strictest IoU threshold 0.5 as shown in Table 1. We can observe that two-stream network [12] shows better performance than C3D network [3].

Table 1. Comparison of different visual encoders used in STPN on THUMOS14

Video feature encoders	mAP ($\theta = 0.5$)
C3D Network	7.9
Spatial-Stream Network	7.3
Temporal-Stream Network	11.3
Two-Stream Network	**14.0**

3.3 Evaluation on Attention Module

Attention module serves as a soft selection method to guide the model to explicitly focus on important parts of the input videos. To study the contribution of attention module, we evaluate STPN with and without attention module separately under the strictest IoU threshold 0.5 as shown in Table 2. We can observe that STPN with attention module gives a significant boost in performance.

Table 2. Comparison of different architectures used in STPN on THUMOS14

Networks	mAP ($\theta = 0.5$)
STPN (w/o attention module)	10.9
STPN	**14.0**

3.4 Evaluation on Shot-Based Sampling Method

We claim that the speed of action detection task can be accelerated using a subset of discriminative snippets in a video. To check the effects of our proposed sampling method, we evaluate temporal pooling network without snippet sampling, with uniform sampling and with shot-based sampling method during training phase using one Nvidia 1080 graphic card. We use the strictest threshold 0.5 and compute the mean time cost of per video during evaluation. As shown in Table 3, TPN with snippet sampling performs better than without sampling and our proposed shot-based sampling method leads to the best performance in both mAP and time consumption.

Table 3. Comparison of different sampling methods used in STPN on THUMOS14

Methods	mAP ($\theta = 0.5$)	Time cost
w/o snippet sampling	2.2	2.35 s
Uniform sampling	11.1	0.21 s
Shot-based sampling	**14.0**	**0.19 s**

3.5 Comparison with the State-of-the-Art Methods

Our approach is also compared with some state-of-the-art methods [6–8,11,13,21,22] of full supervision or weak supervision. In [11], Singh et al. introduce a hide-and-seek strategy to force the network to seek other relevant parts when the most discriminative parts are hidden. However, this method hides the temporal regions randomly and blindly without guidance, thus is inefficient. Wang et al. [13] adopt a bottom-up mechanism for weakly supervised action detection in untrimmed videos, where clip proposals are first generated for classification. However, the use of softmax function across proposals blocks it from distinguishing multiple action instances existing in the videos. Compared with these weakly supervised methods, our STPN can not only highlight the important parts of the input video feature sequence automatically and efficiently, but also detect the class-specific action instances accurately. Comparison results are shown in Table 4. We can observe that our approach outperforms other weakly supervised state-of-the-art methods and even achieves comparable performance to that of fully supervised methods, which demonstrates the effectiveness of our temporal pooling network on learning from long and untrimmed videos. Qualitative examples on THUMOS14 dataset are shown in Fig. 3.

Table 4. Comparison of our method with other state-of-the-art methods on THUMOS14 for action detection. * indicates using full supervision for training

mAP@$IoU(\theta)$	0.5	0.4	0.3	0.2	0.1
Oneata et al. [21]*	14.4	20.8	27.0	33.6	36.6
Shou et al. [6]*	19.0	28.7	36.3	43.5	47.7
Lin et al. [7]*	24.6	35.0	43.0	47.8	50.1
Zhao et al. [8]*	29.8	41.0	51.9	59.4	66.0
Gao et al. [22]*	**31.0**	41.3	50.1	56.7	60.1
Singh et al. [11]	6.8	12.7	19.5	27.8	36.4
Wang et al. [13]	13.7	21.1	28.2	**37.7**	44.4
Ours	**14.0**	**21.4**	**29.1**	37.3	**44.8**

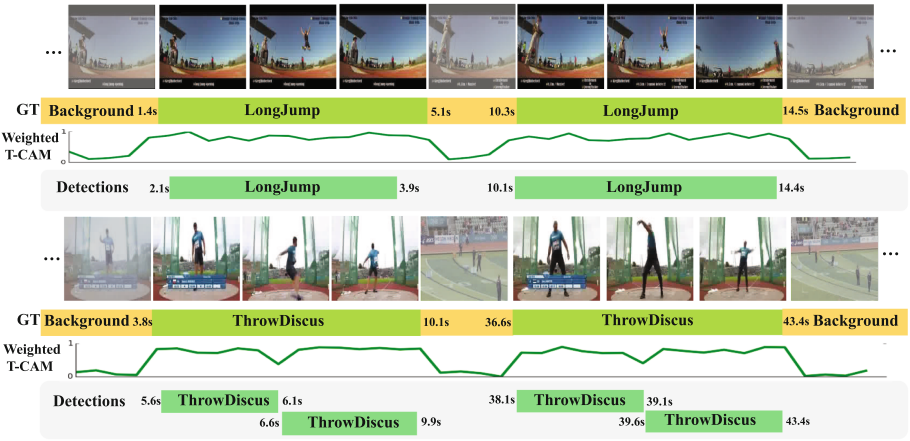

Fig. 3. Qualitative examples of detections generated by STPN on THUMOS14

4 Conclusion

In this paper, we propose a top-down architecture, called STPN, for weakly supervised temporal action detection. In our approach, we first adopt the shot-based sampling method and two-stream network to generate a more representative and simplified feature sequence. Then the classification module can recognize the action categories accurately and the attention module can highlight the relevant frames while suppressing the irrelevant counter-parts simultaneously. Next, the localization module can detect the class-specific discriminative snippets of untrimmed videos by means of generating the weighted T-CAM. Final, we group consecutive snippets with high activations into proposals of variable lengths. Our approach achieves the state-of-the-art performance on the THUMOS14 dataset. In the future, we will try more advanced detection methods and post-processing strategies to generate higher quality proposals with lower redundancy.

References

1. Wang, L., Qiao, Y., Tang, X.: MoFAP: a multi-level representation for action recognition. 2016 Int. J. Comput. Vis. **119**, 254–271 (2016). https://doi.org/10.1007/s11263-015-0859-0
2. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732. IEEE Press, New York (2014)
3. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision, pp. 4489–4497. IEEE Press, New York (2015)
4. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
5. Lin, T., Zhao, X., Fan, Z.: Temporal action localization with two-stream segment-based RNN. In: 2017 IEEE Conference on Image Processing, pp. 1–4. IEEE Press, New York (2017)
6. Shou, Z., Wang, D., Chang, S.: Action temporal localization in untrimmed videos via multi-stage CNNs. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1058. IEEE Press, New York (2016)
7. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: 25th ACM International Conference on Multimedia, pp. 988–996. ACM, California (2017)
8. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: 2017 IEEE International Conference on Computer Vision, pp. 6–7. IEEE Press, New York (2017)
9. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. arXiv preprint arXiv:1806.02964 (2018)
10. Gan, C., Wang, N., Yang, Y., Yeung, D., G.Hauptmann, A.: DevNet: a deep event network for multimedia event detection and evidence recounting. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2568–2577. IEEE Press, New York (2015)
11. Singh, K.K., Lee, Y.J.: Hide-and-Seek: forcing a network to be meticulous for weakly-supervised object and action localization. In: 2017 IEEE International Conference on Computer Vision, pp. 1961–1970. IEEE Press, New York (2017)
12. Simoyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576. Curran Associates Inc., New York (2014)
13. Wang, L., Xiong, Y., Lin, D., van Gool, L.: UntrimmedNets for weakly supervised action recognition and detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2–6. IEEE Press, New York (2017)
14. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929. IEEE Press, New York (2016)
15. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24673-2_3

16. Yamasaki, T.: Histogram of oriented gradients. In: Journal of the Institute of Image Information and Television Engineers, pp. 1368–1371 (2010)
17. Lin, M., Chen, Q., Yan, S.: Network in network. In: 2014 IEEE International Conference on Learning Representations, pp. 1–4. IEEE Press, New York (2014)
18. Jiang, Y.G., et al.: THUMOS challenge: action recognition with a large number of classes. In: ECCV Workshop, vol. 5. Springer, Heidelberg (2014)
19. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: 22nd ACM International Conference on Multimedia, pp. 675–678 (2014)
20. Abadi, M., Agarwal, A., Barham, P., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
21. Oneata, D., Verbeek, J., Schmid, C.: The LEAR submission at thumos2014. In: Thumos14 Action Recognition Challenge, pp. 1–7. Springer, Heidelberg (2014)
22. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. arXiv preprint arXiv:1705.01180 (2017)