Research Article

Human Activity Recognition and Location Based on Temporal Analysis

Hongjin Ding ¹, Faming Gong ¹, Wenjuan Gong,¹ Xiangbing Yuan,² and Yuhui Ma ¹

¹China University of Petroleum (East China), Qingdao 266580, China

²China Petroleum & Chemical Corporation Sheng Li Oilfield Branch Ocean Oil Production Plant, Dong Ying 257237, China

Correspondence should be addressed to Faming Gong; gaming@163.com

Received 6 July 2018; Revised 30 October 2018; Accepted 5 November 2018; Published 16 December 2018

Academic Editor: Shang-Hong Lai

Copyright © 2018 Hongjin Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Current methods of human activity recognition face many challenges, such as the need for multiple sensors, poor implementation, unreliable real-time performance, and lack of temporal location. In this research, we developed a method for recognizing and locating human activities based on temporal action recognition. For this work, we used a multilayer convolutional neural network (CNN) to extract features. In addition, we used refined actionness grouping to generate precise region proposals. Then, we classified the candidate regions by employing an activity classifier based on a structured segmented network and a cascade design for end-to-end training. Compared with previous methods of action classification, the proposed method adds the time boundary and effectively improves the detection accuracy. To test this method empirically, we conducted experiments utilizing surveillance video of an offshore oil production plant. Three activities were recognized and located in the untrimmed long video: standing, walking, and falling. The accuracy of the results proved the effectiveness and real-time performance of the proposed method, demonstrating that this approach has great potential for practical application.

1. Introduction

Detecting human behavior has always been an important research topic in the field of computer vision [1–3]. Automating human activity recognition is crucial for maximizing our understanding and application of videos. In recent years, because of the explosive growth of video data and the urgent need for all aspects of intelligent video processing, behavior detection methods based on timing have received increased attention. While some attempts [4, 5] have been made to tackle untrimmed videos, the performance of existing methods remains far from satisfactory. There are many obstacles to the automated detection of behavior over time. When detecting a motionless target, the boundary of the object is usually very clear [3], so it is possible to set a more specific bounding box. Furthermore, activity recognition is possible using only the information of a static image without incorporating temporal information [6–8]. However, when attempting to discriminate the start and end of a behavior, it is difficult to provide an accurate boundary, i.e., the precise

number of frames that contain the start and end of the action. In timing behavior detection, it is impossible to use only static image information. Identification of timing behavior requires incorporation of temporal information by using a recurrent neural network (RNN) to read features extracted with a convolutional neural networks (CNNs) on each frame [9], or a temporal convolution method [10], or other techniques [11]. Moreover, the duration of temporal behavior segments can be quite long. For example, the video dataset produced by the monitoring system of the offshore oil drilling platform used in our experimental work (discussed later in this paper) ranged from a few seconds to several thousand seconds.

The main purpose of this research was to design a precise timing behavior detection method that can identify and locate stills, falls, and their time nodes in unpruned long videos. The keys to success for this task lie mainly in meeting the following two requirements. First, high-quality timing clips are necessary. The quality of timing fragments directly affects the temporal accuracy for recognizing the behavior. Many methods use the approach of generating

candidate regions and then categorizing the candidates [12]. The important factor is that the candidate quality must be high, so the number of candidates is reduced as much as possible while ensuring the average recall rate. Second, for all methods, it is very important to obtain accurate timing behavior boundaries and accurate classification, i.e., to obtain accurate category information for the temporal behavior fragments.

To make the classification more accurate, in this paper, we used a refined temporal actionness [13] grouping (RTAG) to improve the action classification by setting the threshold of the action score. The input was an unprocessed monitoring video of an offshore oil production platform entered in an operational classifier trained by a deep CNN network to output an operational score. The waveform resulting from the action score entered the refined temporal action packet network to form a candidate area and then was sent to the action classifier to obtain the detection result. In other words, the method involved generating candidates first and then classifying them.

The identification and positioning of human activity based on time series analysis have great potential significance for practical application. For example, offshore oil drilling and production platforms are far away from land and experience complex sea conditions. Personnel working on these platforms are faced with various risks, such as the possibility of injury by machinery, falling into the sea, or difficulty escaping unforeseen disasters. In addition, security problems and economic losses will occur should the rig be boarded by attackers. To understand the whole process of an accident or other incident, and to develop an early warning system that could prevent similar events in the future, it is important to locate the time period during which the incident occurred. However, this task is difficult to accomplish manually [14]. Automated positioning of temporal human activity can allow quick location of the segment of surveillance video to be searched and support analysis of whether the behavior of a worker was abnormal.

The main contributions of this paper are as follows. (1) We used an RTAG network multithreshold refinement action classification boundary to meet the requirements for different levels of precision, and we used a deep learning web framework to improve accuracy. (2) We demonstrated the effectiveness and real-time performance of our proposed approach by applying the method to complex scenes to identify and locate human activities from the surveillance video of an offshore oil platform.

The remainder of this paper is presented as follows. Section 2 provides a review of related research. In Section 3, we describe our method and the proposed human activities recognition framework. Section 4 provides our experimental results and analysis, while Section 5 presents our conclusions and suggestions for future work.

2. Related Work

To locate the time of an abnormal event quickly from a video, and to understand the actions of multiple offshore oil-excavation platform workers in a timely manner, a human

activity recognition system must involve three components: action recognition, object detection, and temporal action detection. In recent years, significant progress has been made in research on behavior identification. Early methods generally were based on hand-crafted visual features [15, 16]. With further research and development, behavior detection methods based on deep learning have had significant effects on detection performance, e.g., convolutional neural networks [9] and two-stream architecture [17]. The application of a 3D-CNN network [18] combines the temporal and motion features of video and can extract and classify video from multiple dimensions. However, for the most part, these methods have dealt with short videos or small pieces of video. For some methods that explore unpruned long videos, other combined methods have proven useful, such as a segmented network structure [19]. Since deep learning has been applied to behavior detection, unsupervised learning techniques [20] have learned the spatiotemporal action characteristics of a video from data and have implemented end-to-end [4] training in a cascade.

At present, object recognition methods can be classified into two broad categories: (1) model-based or context-based recognition methods and (2) two-dimensional or three-dimensional object recognition methods. In the early days of this field, the main methods for object detection [21–23] involved generating candidates through a bottom-up system approach, sometimes using sliding windows to generate candidate regions [5, 24]. Then, the candidates were classified. At present, candidate regions generated based on the deep learning method have demonstrated a better average recall rate when there are fewer candidates. The depth model also introduces powerful modeling capabilities for capturing the appearance of objects. Spatial structure modeling has strong visual characteristics and is still the key to detection. The potential value of spatiotemporal regions-of-interest (RoI) pools [25] to model the spatial three-dimensional structure of a target with minimal extra cost has been further demonstrated using R-FCN, which is a variant of Faster R-CNN [26].

The development of temporal detection has achieved remarkable results. Previous work on behavior detection mainly used the sliding window as a method for generating candidate regions and focused on designing hand-structured feature representations and classifications [5, 24]. Recent work has incorporated in-depth networks into the detection framework and has achieved better performance [4]. However, the performance indicators for research regarding sequential motion positioning are still very low. Classifying hundreds or thousands of candidates requires oversight. Because of the cost of monitoring, all these methods are limited to relatively small datasets and cannot be generalized simply to more data types.

Commonly, contemporary methods for identifying an initial timing action location are based on a sliding window scheme. First, feature extraction is performed on the video. Different length candidates are generated through the sliding window, and the candidates are then classified. In comparison, the most recent method uses the action area [27] to reduce the complexity of the search. First, the target area is extracted from the video, feature extraction is performed, and

FIGURE 1: An overview of the proposed framework.

then the target area is selected for classification. These methods must annotate each frame of the picture and then carry out video training. For a large-scale dataset, the tagging work cannot be done quickly, which presents an urgent problem to be solved for action recognition based on unsupervised timing motion positioning. Because of the similarity between timing motion location and target detection, many time-series motion location methods use a framework similar to some object detection methods.

3. Method: Human Activities Recognition Framework

For this research, we used the framework shown in Figure 1 to identify and locate quickly segments of human activity in the video of an actual offshore oil production platform. Our proposed approach works as follows. One input untrimmed video is divided into same-length segments. Next, the segments enter the binary classifier trained by temporal segment networks [19]. The binary classifier scores the similarity between the regional candidates and the standard actions, resulting in a one-dimensional actionness [13] score waveform. The fractional waveform is sent to the RTAG network. We set different RTAG network thresholds to achieve different positioning accuracy requirements. Proposals of varying accuracy are generated by the RTAG network, and all proposals are input into the activity classifier and completeness filters for detection of the action class and boundary. Finally, a detection result is obtained.

Figure 1(a) is the process of generating fragments of the same length. Figure 1(b) shows that a set of temporal action

proposals (in orange color) are generated by refined temporal actionness grouping (RTAG) from the action detection framework suggested by evaluating the operability of the video snippets. These proposals will be evaluated against cascading classifications to verify their relevance and completeness.

Since the main factor affecting the accuracy of temporal action detection is the quality of the candidate, the technique presented in this paper focuses on generating high quality proposals. Taking into account the purpose and requirements of the search can expedite the location of a certain behavior fragment, which requires our network framework to have a higher detection efficiency.

3.1. Binary Classifier Based on the Temporal Segment Network.

To evaluate the actionness, we learn a binary classifier based on the temporal segment network proposed in [19]. The role of the binary classifier is to determine the action score of the input video segment and then form a one-dimensional action score waveform. We define the one-dimensional waveform output by the binary classifier as the actionness. This is a measure that does not consider the action category but only measures whether there is motion in the video snippets. In order to enable the network to use the entire video information, it will be divided into k segments, and a short snippet will be randomly selected from each segment. The selected snippets are obtained by the two-stream convolutional neural network. Obviously, a lot of short snippets can be sampled, and sequences of short snippets are respectively input into the network. Each snippet can obtain a score of the video classification, and the scores

FIGURE 2: Temporal segment network.

of these snippets can be combined to obtain the final category score. The video-level framework is shown in Figure 2.

The temporal segment network models are composed of spatial stream ConvNets and temporal stream ConvNets. First, we divide the given video (V) into K segments $\{S_1, S_2, \dots, S_K\}$ of equal durations. Second, the temporal segment network models a sequence of snippets as follows:

$$\begin{aligned} TSN(T_1, T_2, \dots, T_K) \\ = H(G(F(T_1; W), F(T_2; W), \dots, F(T_K; W))). \end{aligned} \quad (1)$$

Here, $(T_1, T_2, \dots, T_K)$ is a series of snippets. Randomly sampled T_K is snippet from its corresponding segment S_K . $F(T_K; W)$ is the function representing ConvNet with parameters W , which operates on the short snippet T_K and produces class scores for all the classes. The segmental consensus function G combines the outputs from multiple short snippets to obtain a consistency of class hypotheses between them. Based on this consensus, the prediction function H predicts the probability of each action class for the whole video. In this network, framework chooses the widely used Softmax function for H . The loss function regarding the segmental consensus

$$G = G(F(T_1; W), F(T_2; W), \dots, F(T_K; W)) \quad (2)$$

is formed as

$$L(y, G) = -\sum_{i=1}^C y_i \left(G_i - \log \sum_{j=1}^C \exp G_j \right), \quad (3)$$

where the number of action classes is C and the ground-truth label is y_i concerning class i . The class score is $G_i = g(F_i(T_1), \dots, F_i(T_k))$, using an aggregation function g . In

the backpropagation process, the gradient of the model parameter W relative to the loss value L can be expressed as

$$\frac{\delta L(y, G)}{\delta W} = \frac{\delta L}{\delta G} \sum_{k=1}^K \frac{\delta G}{\delta F(T_k)} \frac{\delta F(T_k)}{\delta W}, \quad (4)$$

where K is the number of segments used by the temporal segment network.

In order to train the binary classifier, we treat all annotated action instances as positive regions and randomly sample negative regions from the video portion of the no-action annotated, in a ratio of 1:1.

3.2. Proposals Generation Based on Refined Temporal Action Grouping Method (RTAG). In this paper, the generation of candidate suggestions is the first step toward accurate detection of temporal action, and it is also the key to achieving accurate boundary locations. To obtain high quality candidates, we introduce the concept of actionness [13]. Actionness is a probability score that evaluates the behavior of unclassified segments in any activity that belongs to this behavior. Therefore, an activity instance is likely to appear in a video portion that contains a segment with relatively high activity, a candidate with a high score. If the binary classifier output is below the set threshold (τ), the output value is given as 0; if the classifier output is greater than or equal to the threshold, the output value is 1.

When we trained the classifier, we took all annotated action instances as positive region samples, and all random sample instances without any motion in the video as negative region samples. Using a series of fragments extracted from the video, we used the classifier as trained above to evaluate the behavior score of each fragment. The range of scores is 0 to 1, so the score can be understood as the probability that the fragment contains an action. In order to generate temporal region candidates [28], we grouped consecutive segments and obtained the high action scores. Since our goal was to address

FIGURE 3: RTAG Network Principle of Discrimination.

the specific requirements of offshore oil platform scenarios, two additional requirements were robustness to noise and the ability to handle long-term changes. With these goals in mind, we designed a robust grouping scheme that allows occasional outliers, e.g., allowing a small fraction of low-actionness snippets within an action segment.

The binary classifier can be trained from video and used to calculate the actionness of snippets. Our goal is to find continuous regions with high probability of action.

After the video frame passes through the binary classifier, a one-dimensional waveform is generated. This waveform shows the score of the region candidate's similarity to the standard action [29]. Our goal is to determine whether the video frame qualifies as showing a standard action based on this similarity waveform. Then, an effective judgment will improve the accuracy of motion recognition.

As shown in Figure 3, this waveform is an action probability waveform of a one-dimensional signal sequence. The level of the waveform represents the probability that the candidate region is a standard action: the higher the waveform, the greater the probability that this prediction will be sampled as a standard action. As the level of the waveform decreases, so does the likelihood that the prediction will be sampled as a standard action. Therefore, a number of samples can be taken and different thresholds can be selected to determine the standard actions. Figure 3 shows two concurrent grouping processes with foreground/background thresholds τ of 0.6 and 0.8. The foreground of each snippet is marked with a "1" and the background is marked with a "0." The blue block boxes are emitted temporal action proposals.

In this paper, different thresholds are used to refine the prediction of standard actions to improve the timing accuracy. The scheme first obtains a variety of action frames through a threshold, where the fragment is a continuous fragment subsequence whose action score exceeds a certain threshold. Then, to generate a region proposal, a segment is selected as a starting point. The region times with high action scores are reexpanded by merging the subsequent segments.

3.3. Detecting Temporal Action Process. After producing a series of candidate areas, the candidates are divided into specific action categories. As mentioned earlier, this step is accomplished using a cascading pipe.

There are two steps: activity classification and completeness filtering. The first step is to delete those belonging to the background and classify the remaining ones. The retained subset may still contain incomplete or overcomplete

instances. The second step will use a class-specific completeness filter to filter these suggestions.

3.3.1. Activity Classification. After the content belonging to the background is removed, the remaining candidate regions are classified using a TSN-based activity classifier [19]. As the process represented by the TSN framework in Figure 2, when training the activity classifier, we set the regional proposal that overlaps with the ground-truth instance and has an IOU above 0.7 as a positive sample. However, for negative samples selection, we will treat the proposal as a negative sample when 5% of its time span overlaps with any annotated instance. Because of the small portion of the overlay action, it is also possible to have a low IOU value. The probability from the activity classifier is denoted as P_a .

However, if we treat them as negative samples, the activity classifiers can be seriously confusing because they may still contain some very unreasonable criteria for participating in the abovementioned improvements [33], and we can exclude these samples from the training set so that activity classification can focus on distinguishing behaviors and backgrounds of interest.

The trained classifier is applied to the video at a fixed frame rate, producing a classification score for each sample segment. Fragment classification scores supported for each region are aggregated into region-level scores to classify candidates into an activity class or context. Specifically, activity classifier A classifies input suggestions as the $K+1$ class, i.e., an active class with tags $1, \dots, k$ and an additional "background" class with tag 0. The classifier limits its scope to the process stage and makes predictions based on the corresponding characteristics. It is implemented as a linear classifier based on advanced features. Given the suggestions, the activity classifier will generate a normalized response vector through the Softmax layer [34].

When people analyze the movements of a human body, they pay more attention to the local movement details, but in video surveillance, often the details of the movement characteristics are not obvious. The action classification task is completed through the RTAG network, so this method has an improved recognition capability.

3.3.2. Completeness Filtering. The second step detects the completeness of the action, that is, the completeness of the action candidate area after the activity classifier. We use the completeness filtering from [13]. To assess the completeness, a simple feature representation is extracted and used to train

FIGURE 4: The proposal classification module.

class-specific SVMs. The feature comprises three parts: (1) two levels of temporal pyramids: The first level pools the snippet scores within the proposed region and the second level split the segment into two parts and pool the snippet scores inside each part; (2) the average classification scores of two short periods: the ones before and after the proposed region. The method is illustrated in Figure 4.

The activity classifiers first remove background proposals and classify the proposals to its activity class. Then, the class-aware completeness filters evaluate the remaining proposals using features from the temporal pyramid and surrounding fragments. The output of the SVMs for one class is denoted as S_c . Then, final detection confidence for each proposal is $S_{Det} = P_a \times S_c$.

4. Experimental Results and Analysis

In this section, we present an evaluation of the effectiveness of our proposed method, and an assessment of the feasibility of practical application. First, we introduce the implementation details of the evaluation dataset and explain our method, and then we discuss the effect of each component in the framework.

4.1. Experimental Preparation. In our approach, we divided the datasets into training sets, validation sets, and test sets in a 2:1:1 ratio. The raw data came from the streaming server at the deep-sea oil production platform. The monitoring equipment on each offshore platform remains stationary, and we used the working platform as a monitoring scene. The real-time monitoring video was transmitted through microwaves and stored in the streaming media server. We use the minibatch stochastic gradient descent algorithm to learn the network parameters, with the batch size set to 256 and the momentum set to 0.9. We set a smaller learning rate in our experiments. For spatial networks, the learning rate is initialized as 0.01 and decreases to its 1/10 every 2,000 iterations. For temporal networks, we initialize the learning rate as 0.005. The maximum iteration is set as 20,000. For

TABLE 1: Data Distribution of Different Action Types.

Action	label	Data/Group
Standing	0	350
Walking	1	350
Falling	2	50
Others	3	100

the extraction of optical flow and warped optical flow, we choose the TVL1 optical flow algorithm [35] implemented in OpenCV with CUDA. To speed up training, we employ a data-parallel strategy with multiple GPUs, implemented with our modified version of Caffe [36] and OpenMPI. The whole training time on UCF101 is around 3 hours for spatial TSNs and 24 hours for temporal TSNs with 4 TITANX GPUs.

In view of the unique nature of the ocean platform scene, this research presets three actions from the perspective of security work: standing, walking, and falling. A total of 1,000 sets of action sequences were collected as the human body motion model library standard. The specific data distribution for each type of action is shown in Table 1.

4.2. Experimental Design and Analysis. We trained the structured segmented networks in an end-to-end manner with the original video frames and the generated candidates as input. We used stochastic gradient descent (SGD) to learn the CNN parameters. The network training is conducted with the publicly available TSN toolbox [19] and Caffe [37]. The initial learning rate of the RGB network was set to 0.1, and the initial learning rate of the optical flow network was set to 0.5. To generate candidates, we used the action classifier trained on the training datasets. The relationship between the classifier recognition rate and the number of image training sets is shown in Figure 9.

Figure 9 displays the trend of the recognition rate of the classifier relative to the number of training sets. We used 10 sets of data for training, from 1,000 to 10,000 samples each, in increments of 1,000. The experimental

FIGURE 5: Target detection combined with classifiers in a marine platform scenario.

FIGURE 6: Scene detection of one person standing.

results showed that, with the change of the number of training sets, the recognition rate of the classifier changed as well. The classifier's recognition rate reached the best level when trained on 8,000 samples. This is an overfitting phenomenon. The weight learning iterations are sufficient (Overtraining), fitting the noise in the training data and the nonrepresentative features in the training examples. Therefore, we chose 8,000 samples as the standard for training the classifier in this experiment. Further combined with the target detection model, the experimental results shown in Figure 5 were obtained.

Figure 5(a) is the original image of the ocean platform taken by the camera. The scene is complex, and the human-like sundries interfere with the detection of people. Figure 5(b) shows that similar-looking interference is detected incorrectly as humans. Figure 5(c) is the final result attained by the combination of the trained target classifier and the target detection model. We can see that Figure 5(c) performs the target detection well. This result demonstrates that, in the complex scene of an offshore platform, the proposed method can detect people accurately.

In the video test, a video was randomly selected, set, and input into the completed training model. We chose a random sample from the standing test set to evaluate. After detection, the test results shown in Figure 6 were obtained.

The system automatically located the target segment and judged the status of the human behavior in the segment. In Figure 6, the oil workers stood on the offshore platform and performed actions with their upper limbs. No movement occurred in their lower limbs, so this behavior was judged as standing. The system demonstrated high accuracy for determining standing and accurate positioning. In various other scenarios, the detection of standing behavior had a good result as well.

For walking behavior in different scenarios, we also conducted a corresponding number of tests. The results are shown in Figure 7.

To assess performance for the walking test set, we selected at random a video that showed a worker walking on the oil production platform. The boxes were marked to show pedestrians.

FIGURE 7: Scene detection of two persons walking.

FIGURE 8: One person recognition and location.

TABLE 2: Action Detection Results for the Oil Production Plant Offshore Platform Video Data Set.

Oil Production Plant Offshore Platform Video Data Set, mAP@ α							
Method	0.1	0.2	0.3	0.4	0.5	0.75	Average
Wang et. al. [30]	17.2	15.0	14.3	12.2	10.5	7.6	12.8
Oneata et. al [31]	30.5	29.6	25.5	23.2	20.5	18.2	24.58
Singh et. al. [32]	33.4	30.5	28.3	26.4	25.3	22.6	27.75
RTAG+CNN	58.3	51.2	48.6	43.5	35.5	29.2	44.38

And to show the relevant time location. Our method achieved a very good detection rate for the detection of walking behavior in different scenarios.

The strong findings for standing and walking demonstrate that our method can meet the needs of practical applications with good versatility. Figure 8 displays the results. After the unprocessed long video entered the proposed network framework, the network quickly located the target segment in the video and judged the active state of the segment. The speed at which the system reads in video is about 15 frames per second, which basically meets the processing speed requirements of the ocean production plant for video. Using the video dataset described, we compared the performance of our proposed method to the approaches used by other researchers. Table 2 displays the results.

Measured by mean average precision (mAP) for different IoU thresholds α , the average mAP of IoU thresholds ranged from 0.1 to 0.75.

According to the abovementioned experimental results, we offer the following conclusions. (1) Based on class-independent behavior, our candidate method for RTAG tagging is good at generating time candidates, and it can be generalized well to unseen activities. The sparse scheme generated by RTAG facilitates the detection performance. (2) The two-stage cascade design of the classification module is crucial for high-accuracy time-of-flight motion detection. It is also a general-purpose design that is well suited for activities with different time structures. (3) This method can predict motion instances directly in unpruned video. This approach also can combine the feature extraction process

FIGURE 9: Relationship between target detection, recognition rate, and training set data in a marine platform scenario.

with the CNN network to form an end-to-end capacity to train the entire framework directly from the original video.

5. Conclusion

Our proposed technique for time-series motion detection achieved good results for human motion recognition and timing positioning in complex scenes. We placed the general framework for time-series motion detection tasks in a specific practical application scenario, refined the candidate generation method, and simplified the subsequent classification network. For this classification model, we used the approach of generating the candidates first and then classifying them. We introduced temporal grouping of the action candidates, and a cascade design for the candidate classifier. Empirical testing using the real-world scenario demonstrated that the requirements for activity recognition were well satisfied.

In addition, we demonstrated that our method is accurate and versatile. It can locate time boundaries accurately and can deal well with activity categories that have different time structures. For a small number of false positives, the model can be improved by further adjusting the threshold. Effective fusion based on a variety of traditional human body motion recognition algorithms makes it suitable for use in offshore oil production platform environments that are far from land, thus ensuring the safety of workers and the smooth development of platform work.

Although the proposed method achieved better results than other methods for target detection, the accuracy of local fine motion recognition still needs to be improved. Because of the limited size of the dataset, other potential issues such as recognition of complex actions could not be considered in this paper. Solving the identification and accurate positioning of these complex actions will become the next main step for research.

Data Availability

The [DATA TYPE] data used to support the findings of this study are included within the supplementary information files.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the Ministry of Science and Technology of China. Under the special plan of innovation work of the Ministry of Science and Technology, it obtained research and application support for innovative methods of oil and gas development in the big data environment [no.2015IM010300]. At the same time, it was awarded the Natural Science Foundation of Shandong Province support [no. ZR2015FM022].

Supplementary Materials

We provide the key part of the implementation code in the paper. **1.Prerequisites.** The training and testing in SSN is reimplemented in PyTorch for the ease of use. We need the following software to run SSN. Python3, [PyTorch][pytorch], [DenseFlow][df] (for frame extraction and optical flow). Other minor Python modules can be installed by running `pip install -r requirements.txt`. GPUs are required to for optical flow extraction and running SSN. Usually, 4 to 8 GPUs in a node would ensure a smooth training experience. **2.Prepare the proposal lists.** `python gen_proposal_list.py DATASET FRAMES_PATH`. **3.Train RTAG models.** **4. Training binary actionness classifier.** `python binary_train.py thumos14 MODALITY -b 16 -lr_steps 20 40 -epochs 45`. `python binary_train.py activitynet1.2 MODALITY -b 16 -lr_steps 3 6 -epochs 7`. **5. Generating RTAG proposals.** `python gen_bottom_up_proposals.py ACTIONNESS_RESULT_PICKLE -dataset thumos14 -subset validation-write_proposals data/thumos14_tag_val_proposal_list.txt -frame_path FRAME_PATH`. `python gen_bottom_up_proposals.py ACTIONNESS_RESULT_PICKLE -dataset thumos14 -subset testing-write_proposals data/thumos14_tag_test_proposal_list.txt -frame_path FRAME_PATH`. **6. Evaluating on benchmark datasets.** **7. Using reference models for evaluation.** `python ssn_test.py DATASET MODALITY none RESULT_PICKLE -use-reference`. Additionally, we provide the models trained with Kinetics pretraining; to use them, run `python ssn_test.py DATASET MODALITY none RESULT_PICKLE -use_kinetics-reference`. **8. Training SSN.** **9. Training with ImageNet pretrained models.** `python ssn_train.py thumos14 MODALITY -b 16 -lr_steps 20 40 -epochs 45`. **10.Related Projects.** (i) [UntrimmedNets][untrimmednets]: Our latest framework for learning action recognition models from untrimmed

videos (CVPR'17). (ii) [Kinetics Pretrained Models][action_kinetics]: TSN action recognition models trained on the Kinetics dataset. (iii) [TSN][tsn]: state-of-the-art action recognition framework for trimmed videos (ECCV'16). (iv) [CES-STAR@ActivityNet][anet]: winning solution for ActivityNet challenge 2016, based on TSN. (v) [Enhanced-MV][emv]: real-time action recognition using motion vectors in video encodings. (*Supplementary Materials*)

References

- [1] N. Zheng, G. Loizou, X. Jiang, X. Lan, and X. Li, "Preface: Computer vision and pattern recognition," *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1265–1266, 2007.
- [2] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.
- [3] T. Song and L. Pan, "On the universality and non-universality of spiking neural P systems with rules on synapses," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 960–966, 2015.
- [4] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2678–2687, 2016.
- [5] T. Song, X. Zeng, P. Zheng, M. Jiang, and A. Rodriguez-Paton, "A Parallel Workflow Pattern Modelling Using Spiking Neural P Systems with Colored Spikes," *IEEE Transactions on NanoBioscience*, 2018.
- [6] M. S. Ryoo and L. Matthies, "First-person activity recognition: feature, temporal structure, and prediction," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 307–328, 2016.
- [7] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From Activity Recognition to Intention Recognition for Assisted Living Within Smart Homes," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 368–379, 2017.
- [8] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 498–509, 2016.
- [9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *Proceedings of the CVPR*, pp. 1725–1732, 2014.
- [10] J. Donahue, L. A. Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the CVPR*, pp. 2625–2634, 2015.
- [11] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Spatio-temporal closed-loop object detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1253–1263, 2017.
- [12] L. Anselma, L. Piovesan, and P. Terenziani, "Temporal detection and analysis of guideline interactions," *Artificial Intelligence in Medicine*, vol. 76, pp. 40–62, 2017.
- [13] T. Song, A. Rodriguez-Paton, P. Zheng, and X. Zeng, "Spiking Neural P Systems with Colored Spikes," *IEEE Transactions on Cognitive and Developmental Systems*, 2018.
- [14] T. Song, P. Zheng, M. L. Dennis Wong, and X. Wang, "Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control," *Information Sciences*, vol. 372, pp. 380–391, 2016.
- [15] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.
- [16] S. Pang, T. Ding, A. Rodriguez-Paton, T. Song, and P. Zheng, "A Parallel Bioinspired Framework for Numerical Calculations Using Enzymatic P System with an Enzymatic Environment," *IEEE Access*, vol. 6, pp. 65548–65556, 2018.
- [17] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the NIPS*, pp. 568–576, 2014.
- [18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3d Convolutional Networks," in *Proceedings of the ICCV*, pp. 4489–4497, 2015.
- [19] L. Wang, Y. Xiong, Z. Wang et al., "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," in *Proceedings of the ECCV*, pp. 20–36, 2016.
- [20] S. Hoochang, H. R. Roth, M. Gao et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [21] J. X. Tao Song and L. Pan, "Spiking neural P systems with request rules," *Neurocomputing*, vol. 193, pp. 193–200, 2016.
- [22] C. Vondrick, A. Khosla, H. Pirsiavash, T. Malisiewicz, and A. Torralba, "Visualizing object detection features," *International Journal of Computer Vision*, vol. 119, no. 2, pp. 145–158, 2016.
- [23] V. Object and D. Features, "Hoggles Visualizing Object Detection Features," *International Journal of Computer Vision*, vol. 1, no. 8, 2013.
- [24] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.
- [25] J. Li, L. Chen, S. Cai, C. Cai, J. Zhong, and Z. Chen, "Imaging with referenceless distortion correction and flexible regions of interest using single-shot biaxial spatiotemporally encoded MRI," *NeuroImage*, vol. 105, pp. 93–111, 2015.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the NIPS*, pp. 91–99, 2015.
- [27] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proceedings of the CVPR*, pp. 1049–1058, 2016.
- [28] A. C. Papanicolaou, P. G. Simos, and E. M. Castillo, "The Hippocampus and Memory of Verbal and Pictorial Material," *Learning Memory*, vol. 9, no. 3, pp. 99–104, 2017.
- [29] Y. Tian, Y. Kong, Q. Ruan, G. An, and Y. Fu, "Hierarchical and spatio-temporal sparse representation for human action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1748–1762, 2018.
- [30] L. Wang, Y. Qiao, and X. Tang, "Action Recognition and Detection by Combining Motion and Appearance Features," in *THUMOS Action Recognition Challenge*, 2014.
- [31] D. Oneata, J. Verbeek, and C. Schmid, "The LEAR Submission at Thumos 2014," *Computer Vision and Pattern Recognition [cs.CV]*, 2014.
- [32] G. Singh and F. Cuzzolin, "Untrimmed video classification for activity detection: submission to activitynet challenge," *Computer Vision and Pattern Recognition (cs.CV)*, 2016.
- [33] K. Schindler and L. Van Gool, "Action Snippets: how many frames does human action recognition require?" *Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [34] M. M. A. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, and R. R. Yager, "Deep learning approach for active classification of

- electrocardiogram signals,” *Information Sciences*, vol. 345, pp. 340–354, 2016.
- [35] C. Zach, T. Pock, and H. Bischof, “A Duality Based Approach for Realtime Tv-L₁ Optical Flow,” in *Proceedings of the 29th DAGM Symposium on Pattern Recognition*, vol. 5, pp. 214–223, 2007.
- [36] Y. Jia, E. Shelhamer, J. Donahue et al., “Caffe: convolutional architecture for fast feature embedding,” *Neural and Evolutionary Computing (cs.NE)*, 2014.
- [37] A. S. Garea, D. B. Heras, and F. Argüello, “Caffe CNN-based classification of hyperspectral images on GPU,” *The Journal of Supercomputing*, pp. 1–13, 2018.