**Real-Time Head Action Recognition Based on HOF and ELM**

Tie HONG†, Student Member, Yuan Wei LI††, and Zhi Ying WANG†, Nonmembers

**SUMMARY**  Head action recognition, as a specific problem in action recognition, has been studied in this paper. Different from most existing researches, our head action recognition problem is specifically defined for the requirement of some practical applications. Based on our definition, we build a corresponding head action dataset which contains many challenging cases. For action recognition, we proposed a real-time head action recognition framework based on HOF and ELM. The framework consists of face detection based ROI determination, HOF feature extraction in ROI, and ELM based action prediction. Experiments show that our method achieves good accuracy and is efficient enough for practical applications. 

**key words:** action recognition, head action recognition, histograms of flow orientations (HOF), extreme learning machine (ELM)

1. Introduction

Recently, action recognition has been widely studied and achieved significant research progresses. The goal of action recognition is to enable machine automatically recognize human action in realistic action video. In this area, many action datasets, such as the popular UCF50 [1], UCF101 [2], HMDB51 [3] and Hollywood-2 [4] are proposed and greatly promote researches. In this paper, our head action recognition problem is slightly different from most existing researches. Most existing datasets and researches aim at classify a wide range of human actions, such as Run, LongJump, Nunchucks, SkyDiving, DriveCar, Kiss, AnswerPhone. However, our head action recognition problem (see Sect. 3.1 and Table 1) focuses only on actions of human head. Compared to existing researches, the number of action classes is lower, but our problem definition also contains many challenges, such as “left-rotate”↔“left-skew”, confusing cases in “no-action” class (like too slow/fast confusing actions as described in Sect. 4.1). The most widely used areas for head action recognition are games and entertainment. For example, controlling characters in the game brings a real experience to the player. Adjusting sounds and screens enhance the audience’s live experience in interactive entertainment. In addition, frequently nodded (nose-down/nose-up) may indicate the driver is tired. In some intelligent monitoring scenarios, the head action can also reflect the abnormal behavior of the monitored object. In some psychological aspects, the head action can also reflect the psychological state of the person. Frequently swinging head around (left-rotate/right-rotate) may reflect inner tension and timidity. Under this definition, we propose a real-time head action recognition framework based on histograms of flow orientations (HOF) and extreme learning machine (ELM). In summary, we make following three-fold contributions:

(1) We define a new head action recognition problem and build a corresponding head action dataset;

(2) We propose a real-time head action recognition framework which consists of face detection based ROI determination, HOF feature extraction in ROI, and ELM based action prediction;

(3) ELM is used to classify HOF descriptor and trained effectively with data augmentation, data balancing and boosting “no-action” samples.

2. Related Work

We just give a brief introduction to existing researches in this section. Most early action recognition approaches adopt Bag-of-Features (BOF) and its variants [5]. To represent actions, BOF approaches use collections of local space-time hand crafted descriptors aggregated over a period of time. In addition, some other popular local descriptors include histograms of flow orientations (HOF) [6], histograms of 3D gradients (HOG3D) [7], motion boundary histograms (MBH) [8], shapes of point trajectories [9], local trinary patterns [10] and so on. The drawback is that these work need to manually craft descriptors which highly determine the recognition performance. As CNN has achieve great success in many computer vision targets, like object detection, object recognition, image segmentation and others, many researchers exploit CNN for action recognition task. The two-stream CNNs framework [11] proposed by K. Simonyan and A. Zisserman is a significant work. They exploit two CNNs to model RGB and optical flow respectively and achieve good performance on several large action datasets. Based on two-stream framework, many researchers [12] propose their improvements. To capture the temporal structure in video, LSTM architecture also is considered for action recognition and further improve the recognition performance [13]. Compared to traditional descriptor, CNN based methods...
stand out in the recognition accuracy, but require an enormous amount of computing power. Considering our demand for fast speed on general hardware (like mobile phone), we describe head action via the traditional HOF feature.

Some existing works, such as face landmark detection, human pose estimation, may be modified for our head action recognition problem. Considering real-time requirements without GPU, detecting face landmarks and estimating head pose angular based on some fast face landmark detection algorithms [14] is really a good solution. Combining face landmark based framework and our optical flow based framework is part of our future studies. In contrast, face landmark based framework directly recognizes head action based on head pose angular, while our optical flow based framework uses motion information described by HOF. We think that a difficulty of face landmark based framework is robust face detection and head pose angle estimation in a wide angle region (specially for some challenging camera angles and complex action types as shown in our supplementary materials). Currently, our algorithm only needs to detect frontal face in a small angle range via a lightweight and fast face detection model to trigger few ELM classifications and have lower requirements for face detector.

3. Our Approach

3.1 Head Action Definition

Assume facing the camera, we define head action based on the coordinate system given in Fig. 1. When keeping head upright, as shown in the figure, the front of the body is parallel to the XY plane and Z-axis point straight ahead. Each action is produced by rotating head around one of the three axes and then back to the upright pose. Under this coordinate system, we define seven kinds of head actions as shown in Table 1 which contains a special “no-action” type. For example, action 1 (“nose-down”) is to rotate > +30° around X axis. As a special case, action 7 (“no-action”) means that the current head action doesn’t fall into any of 1–6 actions, such as, keeping still, rotating enough angle but not back and so on. Especially, our head action dataset deliberately introduces many confusing samples into action 7 (see Sect. 4.1).

3.2 Overview

The overview of our approach is given in Fig. 2. Firstly, cascade face detector of OpenCV is used to detect face in current image for two purposes: 1) thinking as “no-action” and return if no face is found, and else 2) the biggest face is used to determine ROI for HOF extraction. Considering recognition speed, we adopt HOF instead of CNN to describe head action information in image sequence. Optical flow is the base of HOF which largely determines the recognition accuracy and speed. We compute optical flow field for head action recognition via a highly optimized variant of PGM [15], which is efficient and robust enough for both large and small motion. Different from most work that normalize input image size and compute HOF from the whole image, we just consider the optical flow in the region of interest (ROI) determined by face detection. The ROI is a rect with fixed size (216 × 216 in our implementation) and the same centre as the face bounding box. Just considering motion in the ROI is a reasonably effective way for our limited action recognition problem to improve recognition accuracy and robustness. After optical flow field and face bounding box are available, we extract HOF descriptor similar to [16]. Finally, we feed the HOF feature vector to ELM and predict the current action type.

3.3 HOF Feature Extraction

For the sake of clearness, we describe the calculation of HOF feature in detail. Figure 3 gives a visual demonstration

| Action ID | Description | Rotation axis | Angle |
|-----------|-------------|---------------|-------|
| 1         | nose-down   | X             | > +30° |
| 2         | nose-up     | X             | < −30° |
| 3         | left-rotate | Y             | > +30° |
| 4         | right-rotate| Y             | < −30° |
| 5         | right-skew  | Z             | > +30° |
| 6         | left-skew   | Z             | < −30° |
| 7         | no-action   | -             | -     |
of HOF descriptor extraction. We follow [16] but introducing ROI and obtain our descriptor by computing histograms of PGM flow vectors accumulated in a video block. Each block is divided into cells and the normalized histograms from blocks generated by striding in XY plane are concatenated into a descriptor vector.

As shown in Fig.3, $B_t$ images (interframe space is 100ms) are stacked as the coordinate system where the topmost layer is the latest image. Face detector is applied to the latest image and then ROI is determined based on the biggest detection bounding box. Note that, we ignore current image if no face is found. Then, we compute HOF descriptor for the image cube in the ROI. The image cube is divided into cells whose size are $C_x \times C_y \times C_t \ (24 \times 24 \times 5$ in our implementation), and each block consists of $B_x \times B_y \times B_t$ cells. As shown in the figure, our block size is $2 \times 2 \times 3$ cells and so $[B_x, B_y, B_t]^T = [48, 48, 15]^T$. We discretize flow vector into eight orientation bins and a specific no-motion bin. The no-motion bin is activated when flow magnitude is too small or flow of this pixel is marked as “occluded” (occlusion mask is obtained by forward-backward PGM flow consistency check).

3.4 Action Classification

We choose basic ELM as our action classifier with a view to its high class accuracy and efficient learning. Efficient learning makes our algorithm very easy to adjust and evaluate parameter settings. As we only apply ELM to one ROI window each frame instead of the sliding window strategy in most object detection algorithm, ELM prediction (once forward propagation of a single hidden layer network) is efficient enough. In our implementation, the number of hidden nodes is 4000 and activation function is sigmoidal function.

4. Experiments

4.1 Dataset

Considering there are no existing datasets for the head action recognition problem defined in this paper, we collect ourselves head action dataset with the front-facing camera of a mobile phone. Our action dataset consists of 21 video sequences which are divided into 17 action-included sequences and 4 no-action sequences. Each sequence includes multiple different categories of head actions. Before the data was augmented, there are a total of 826 action samples, and about 30 minutes of all no-action video sequences. 80% are divided into training sets, and others as test sets.

Figure 4 give a visual example of action “right-rotate”, “right-skew” and “no-action”. We walk around and record head action at the same time by a hand-held mobile phone whose front-facing camera aims to the face. When recording video for action 1–6, we keep action period about 2 seconds which is consistent with preference $B_t = 15$ in Sect.3.3. Note that, walking around introduces diversification in camera pose and background which make our action samples more diverse. For action-included sequences, we annotate the end frames of actions manually. All subsequences from no-action sequences do not belong to any of action 1–6, however, we deliberately make them difficult to distinguish from action 1–6. For example, we perform action 1–6 very slow/fast (the third row of Fig. 4) or with long time interruption.

4.2 Training Details

All actions in our dataset are divided into training dataset and testing dataset randomly. Then, we augment dataset by 1) mirroring, 2) resizing, 3) translating face bounding box, 4) adjusting aspect ratio and 5) adjusting the end frame of actions. With these augmentation steps, we can obtain a large number of samples. However, training ELM with so many samples needs huge memory which goes beyond 8GB limit of our computer. In order to reduce training samples and reserve diversity at the same time, for each action type, K-means clustering algorithm is used to divide the augmentative samples into 10 classes and a certain proportion of samples is random sampled from each class respectively.

For negative samples, we adopt the boosting strategy. We first random sample some negative samples and train an ELM model. For boosting, the previous ELM model is used to boost hard negative samples and a new ELM model is trained with the more comprehensive negative dataset. We only boost once in our experiment.

4.3 Performance Evaluation

In this section, we perform quantitative evaluation. Table 2 reports the average class accuracy and speed of our method. All experiments are carried on Intel i5-4200U (1.6GHz) with 8GB memory. The environment for feature extraction is c++/OpenMP and the environment for ELM prediction is Matlab. Our method achieves high class accuracy (95.63%) and is fast enough (≈ 20 fps) for real-time applications. For details of classification error, Fig. 5 reports the confuse ma-
Table 2  The class accuracy and speed of our approach.

| Accuracy | Time cost (ms) |
|----------|----------------|
| Feature extraction | 95.63% | 50 |
| ELM prediction | 1.72% | 2 |

Fig. 5  The confuse matrix of our approach.

We can find that the major confusion occurs between “no-action” and six other meaningful actions, while few errors occur among six meaningful actions. “No-action” is so hard to distinguish as our specially designed no-action sequences (Sect. 4.1) introduce many very confusing actions. Different from many previous works/datasets, we consider “no-action” in our dataset as the requirement of practical applications. Therefore, we not only know what action is, but also when action happens. Interestingly, the most confusing pair among six meaningful actions is regarding “right-rotate” as “right-skew” and regarding “left-rotate” as “left-skew”, which matches the intuitive understanding.

5. Conclusion

In this paper, we define a new head action recognition problem and collect special dataset for our problem. Our problem definition and dataset are compatible with some practical applications, such as game interaction. Then, we propose a real-time head action recognition method for the defined problem. With our carefully designed recognition framework and training, our method achieves high accuracy and fast speed.

References

[1] K.K. Reddy and M. Shah, “Recognizing 50 human action categories of web videos,” Machine Vision & Applications, vol.24, no.5, pp.971–981, 2013.
[2] K. Soomro, A.R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” CoRR, vol.abs/1212.0402, 2012.
[3] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” ICCV, pp.2556–2563, Nov. 2011.
[4] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” CVPR, pp.2929–2936, June 2009.
[5] X. Peng, L. Wang, X. Wang, and Y. Qiao, “Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice,” CoRR, vol.abs/1405.4506, 2014.
[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” CVPR, pp.1–8, June 2008.
[7] A. Klaeser, M. Marszalek, and C. Schmid, “A spatio-temporal descriptor based on 3D-gradients,” BMVC, pp.99:1–99:10, Sept. 2008.
[8] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” ECCV, vol.3952, pp.428–441, Springer Berlin Heidelberg, 2006.
[9] R. Messing, C. Pal, and H. Kautz, “Activity recognition using the velocity histories of tracked keypoints,” ICCV, pp.104–111, 2010.
[10] L. Yeft and L. Wolf, “Local trinary patterns for human action recognition,” ICCV, pp.492–497, 2009.
[11] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” Advances in Neural Information Processing Systems 27, pp.568–576, 2014.
[12] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, “Towards good practices for very deep two-stream convnets,” Computer Science, 2015.
[13] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, “Modeling spatial-temporal clues in a hybrid deep learning framework for video classification,” Proc. 23rd ACM International Conference on Multimedia, pp.461–470, 2015.
[14] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” CVPR, pp.1867–1874, 2014.
[15] Y. Li, “Pyramidal gradient matching for optical flow estimation,” CoRR, vol.abs/1704.03217, 2017.
[16] V. Kantorov and I. Laptev, “Efficient feature extraction, encoding, and classification for action recognition,” CVPR, pp.2593–2600, 2014.