PAPER

Automatic Retrieval of Action Video Shots from the Web Using Density-Based Cluster Analysis and Outlier Detection

Nga Hang DO[†a)], *Nonmember and* **Keiji YANAI**[†], *Member*

SUMMARY In this paper, we introduce a fully automatic approach to construct action datasets from noisy Web video search results. The idea is based on combining cluster structure analysis and density-based outlier detection. For a specific action concept, first, we download its Web top search videos and segment them into video shots. We then organize these shots into subsets using density-based hierarchy clustering. For each set, we rank its shots by their outlier degrees which are determined as their isolatedness with respect to their surroundings. Finally, we collect high ranked shots as training data for the action concept. We demonstrate that with action models trained by our data, we can obtain promising precision rates in the task of action classification while offering the advantage of fully automatic, scalable learning. Experiment results on UCF11, a challenging action dataset, show the effectiveness of our method.

key words: *automatic construction, action datasets, Web videos, density-based clustering, outlier detection*

1. Introduction

High-quality datasets play important roles in computer vision and pattern recognition tasks. With sufficient and high-quality training data, most pattern recognition methods have achieved promising results. As data sources, most recently released datasets exploit Web data which are extremely numerous and easy to obtain. However, since Web data are generated and uploaded by general users, data corresponding to the concept of interest account for only a small proportion among retrieved results. Therefore, constructing high quality datasets with Web data requires extensive human effort of manual annotation. In case of constructing action datasets, in general, we need annotators to localize relevant video parts (shots) of the pre-defined actions in video sources by watching the whole of them carefully. Since the task is too exhausting, even largest action datasets cover not more than 101 concepts with only several thousands of video shots. This situation has given rise to the need for constructing action datasets with less human effort.

Previous works which aim to automatically obtain action shots of specific action concepts from noisy data [1]–[3] generally require textual information provided together with videos such as movie script [1] or metadata (tags) [2], [3]. Laptev et al. [1], [4], [5] proposed methods to automatically associate movie scripts with actions and obtain video shots in movie representing particular classes of human actions. Their methods actually can help reduce human effort in constructing a realistic action database. However, targeted videos are only the movies with available scripts and trainable actions are limited to only actions appearing in movies. On the other hand, our proposed system can be applied to extract data for various types of actions which are distributed over much more immense video source.

Our previous work [2], [6], [7] proposed to collect video shots corresponding to any kind of action concept using Web videos. We conducted experiments for more than 100 action concepts in the previous work and obtained promising results. The previous work is treated as baseline in this paper. In our previous work, before video downloading, videos are ranked based on usage frequencies of tags. Here tags refer to the most concise words which describe the videos and are provided by video uploaders. Videos which have tags with high co-occurrence frequencies are considered as relevant videos and selected to download.

Until several years ago, we had been able to collect tags using Web API (in case video uploaders provided this kind of words). Recently, however, Youtube API usage policy has changed so that we have been no more able to gather tags as before. Therefore, we can not apply tag-based video selection as proposed in our previous approach. In this work, we propose a new approach which does not require tag information of Web videos.

In action recognition, in most cases, a primary action is considered as a target in both training videos and test videos. Even with only one action, the task is still challenging due to variability of human actions. The actions can look different when they are seen from different views or performed by different people. They can even be manipulated in many disparate ways. Thus, to obtain good recognition performance, training data should capture actions in many different conditions. In other words, a high quality action database should reflect as much as possible the diversity of the concept. However, previous approaches for automatic construction of action database do not cope with concept diversity. Especially, our baseline [6] applies VisualRank [8] which is originally an image ranking method with a visual feature based similarity matrix to rank shots. Shots sharing the most visual characteristics with others are ranked to the top and selected as relevant shots. Therefore, this method tends to obtain only visually similar shots. In this paper, we propose to group related shots into clusters before shot ranking by a hierarchy clustering method [9]. Different clusters while sharing some appearance characteristics

Manuscript received March 14, 2016.
Manuscript revised June 26, 2016.
Manuscript publicized July 21, 2016.

[†] The authors are with The University of Electro-Communications, Chofu-shi, 182–8585 Japan.
a) E-mail: dohang@mm.cs.uec.ac.jp

DOI: 10.1587/transinf.2016EDP7108

still hold unique aspects of the concept. Consequently, our obtained shots are much more diverse than shots obtained by the baseline [6]. According to our experiment results, the more diverse the training data are, the better recognition performance we can achieve.

After obtaining clusters, we rank instances in each cluster by outlier factor [10]. Outliers are instances deviating from the major distribution of the data. In other words, outliers belong to sparse regions while relevant instances lie in dense regions. The most densely linked instances from each cluster are ranked to the top and then used as training data for the concept. As action concepts, we experiment on those used in YouTube (also called as UCF11) dataset [11]. As visual features, we extract temporal features using a ConvNet trained on UCF-101 dataset [12] following a recent state-of-the-art approach for action recognition [13].

Furthermore, we train action models with our automatically collected data and test them on the test data of these datasets. We performed action classification by a popular supervised framework with the intention of comparing classification performance using a manually constructed dataset and a dataset collected automatically by our proposed approach. Experiment results show that even though our data are not qualified as "clean" data as standard training data (manually collected data), classification rates are promising and show potential for development of approaches for automatic construction of action databases. Our work is inspired by Chen et al.'s work [14] which uses density analysis of Web images for automatic image dataset construction.

Even though our work focus on actions, the application of our approach is not limited to actions. Actually, this work tries to improve our previous work regarding tag problem and diversity problem. Since our previous work was experimented with action videos, this work also focused on actions for easier comparison. For the reason why our previous work focused on actions, at that time, most of the works on automatic construction of training datasets with Web data had paid attention to only images, so there had been very few action datasets for action recognition. The purpose of our work is to reduce human effort in construction of action datasets which has been considered as much harder work compared to construction of image datasets.

Our contributions can be summarized as follows: (1) We validate if our automatically constructed datasets can be used as training datasets for standard action video classification task. To the best of our knowledge, there have not been any work with the same purpose before. (2) We propose a novel approach for fully automatic construction of action datasets which requires only visual information of videos yet obtains better quality datasets compared to the previous approach.

Remainder of this paper is constructed as follows. We first introduce some more related work for dataset construction and action recognition in Sect. 2. In Sect. 3 we describe our proposed approach. We then report the results of our experiments in Sect. 4 and finally, conclude this work in Sect. 5.

2. Related Work

We discuss here several related work on two topics: dataset construction and action recognition.

Dataset Construction: Many recent work have tackled the problem of building qualified training datasets automatically from data retrieved by Web search engines but most of them have been applied only on images [14]–[17]. Collins et al. [15] presented a framework for incrementally learning object categories from Web image search results. Given a set of seed images a non-parametric latent topic model is applied to categorize collected Web image. Schoroff et al. [16] proposed to first filter out the abstract images (e.g., drawings, cartoons) and then use text and metadata surrounding the images to re-rank the images searched in Google. Chen et al. [14] proposed NEIL (Never Ending Image Learner) which is a program using a semi-supervised learning algorithm that jointly discovers common sense relationships and labels instances of the given visual categories. NEIL learns multiple sub-model automatically for each concept. As an approach which also alleviates the multi-modal problem of concepts, [17] divides seed images into multiple groups and trains classifiers on each group separately. Images obtained from different groups usually capture some different looks of the concept.

As for automatic construction of action datasets using unconstrained videos, there are very few approaches as we introduced in the previous section. Moreover, these approaches require textual information associated with videos [1], [2]. Adrian et al. [3] proposed a method to learn automatically concept detectors from YouTube videos for any kind of concepts including objects, actions and events. Their method also requires textual description of the target concept provided by YouTube users. Furthermore, each concept must be manually assigned a canonical YouTube category and low-quality videos are eliminated to improve the quality of downloaded material. In this work, we propose a fully automatic approach for action dataset building which exploits only visual features of raw videos retrieved from video sharing sites. Our approach neither needs additional information nor manual annotation.

Action Recognition: Most action recognition methods followed the standard framework of pattern recognition. First, a sufficiently large corpus of training data is collected, in which the concept labels are generally obtained through expensive human annotation. Next, concept classifiers are learned from the training data. Finally, the classifiers are used to detect the presence of the actions in the test data. We also adopt this standard framework in action recognition task, except that instead of using provided training data, we use our automatically collected data to train concept classifiers.

As popular video presentation, successful hand-crafted features such as HOG, HOF or MBH extracted along dense trajectories [18] have been adopted and developed in many work recently [19], [20]. These features are generally encoded by Bag-of-Visual-Words or Fisher Vectors. In very recent years, followed by their success in image recognition field, deep learning Convolutional Neural Networks (CNNs) have received great attention and obtained promising results in action recognition [13], [21], [22]. Following this trend, we also train a temporal CNN using a method proposed in [13] and use this model to extract features from video shots.

3. Approach

In this work, we present an approach which autonomously extracts from noisy Web videos relevant video shots for given action concepts. Our approach consists of three steps: shot collection, shot clustering and shot selection. See Fig. 1 for the illustration of our proposed framework. In shot collection, we download videos of the concepts and segment them to shots. These shots are then organized into subsets by hierarchical clustering [9]. Finally, relevant shots are ranked by outlier factors [10] and selected from each of all clusters using a simple selection strategy. In the followings, we explain in detail each step.

3.1 Shot Collection

We first prepare keywords for given action concepts. The concepts can be defined in any form: either "verb" (such as "dive") or "verb+non-verb" (such as "throw+hammer", "cut+in+kitchen") or "non-verb" (such as "pole vault"). In case verb included in the keyword, we search for its videos in both forms: "verb" and "verb-ing" (such as "diving", "throwing+hammer"). We filter out videos belonging to "entertainment", "music", "movies", "film" and "games" categories during searching since these categories generally contain extremely long videos. Top search results are downloaded and segmented into video shots using color histogram. RGB histograms of every frame are computed and then segmentation points are put between frames whose histogram intersection is larger than a predefined constant. Each shot represents one single scene. For each concept, we download around 100-200 videos and obtain around 700-2000 shots.

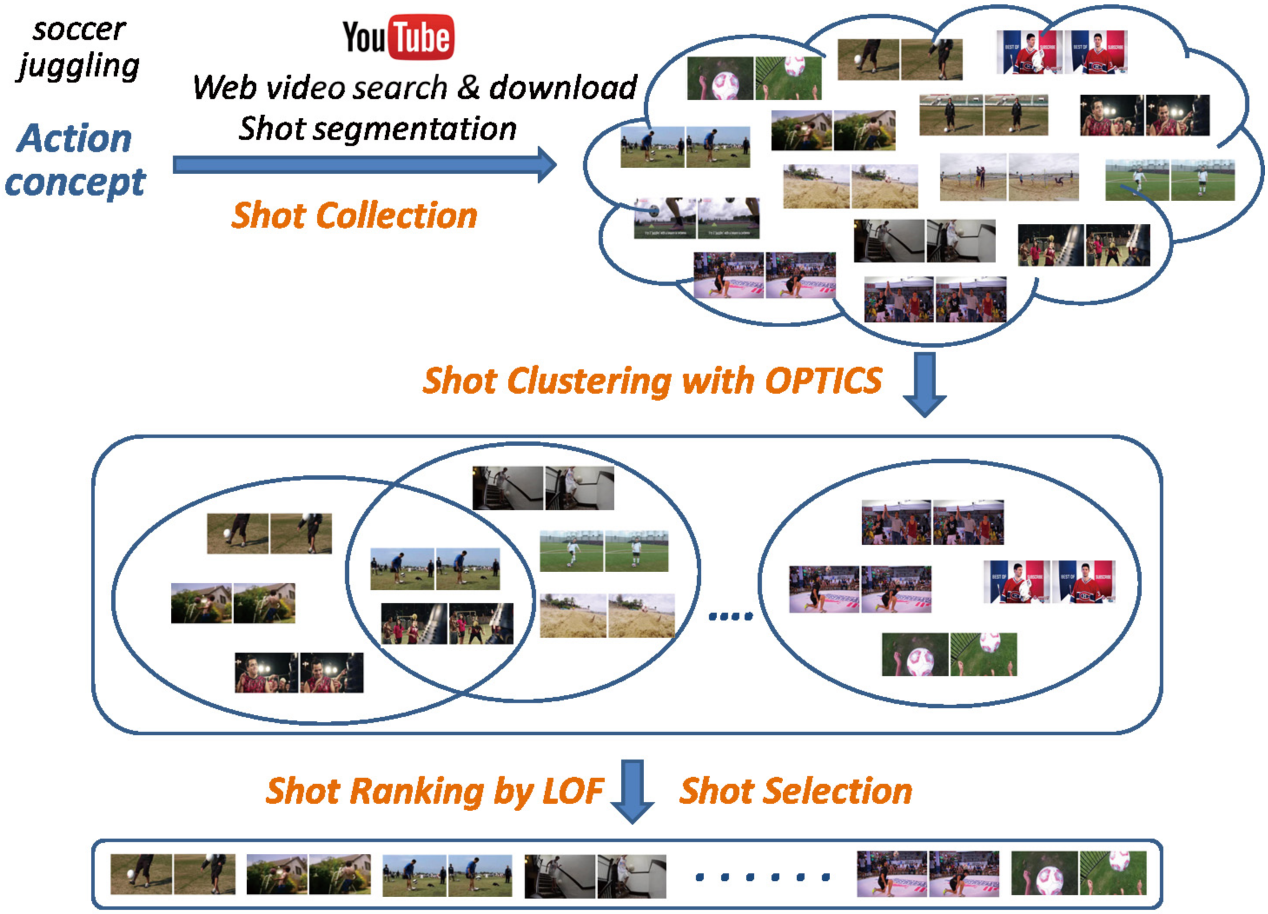

Fig. 1 Framework of our approach for automatic construction of action shot datasets which consists of three steps: shot collection, shot clustering and shot ranking.

3.2 Shot Clustering

With shots obtained after above step, we group related shots into clusters before shot ranking and selection. This step helps deal with concept diversity. With web data retrieved for a given concept, there will also be common characteristics shared among subsets of data. Therefore, rather than hard clustering data into a specific number of subsets as some approaches which also aim to deal with intra-class variations in concepts [17], [23], we use hierarchy clustering which allows different clusters to share the same instances. We adopt OPTICS ("Ordering Points To Identify the Clustering Structure") [9] to find clusters. Rather than the popular Mean Shift, OPTICS is prefer due to its computational efficiency. The hierarchical structure of the clusters can be obtained based on the density of the data distributed around their points. Follows are our brief introduction of this clustering algorithm. For the detail, please refer to [9].

The basic idea of a density-based clustering algorithm is that for each object of a cluster the neighborhood of a given radius has to contain at least a given minimum number of objects (MinPts). Clusters are formally defined as maximal sets of density-connected objects. We introduce here some important definitions while briefly reviewing OPTICS algorithm.

Let p be an object from a dataset D, k be a positive integer and d be a distance metric, then:

Definition 1: $k-\text{dist}(p)$, the k-distance of p, is defined as the distance $d(p,o)$ between p and object $o \in D$ statisfying: 1. at least k objects $q \in D$ having $d(p,q) \leq d(p,o)$, and 2. at most (k-1) objects $q \in D$ having $d(p,q) < d(p,o)$

Definition 2: $N_{k-\text{dist}(p)}(p) = \{q | q \in D, d(p,q) \leq k - \text{dist}(p)\}$ denotes the k-distance neighborhood of p.

Definition 3: $\text{reach} - \text{dist}_k(p,o) = \max(\{k - \text{dist}(o), d(p,o)\})$ represents reachability distance of an object p with respect to object o.

The OPTICS-algorithm computes a "walk" through the data, and calculates for each object the smallest reachability-distance with respect to an object considered before it in the walk. A low reachability-distance indicates an object with a cluster, and a high reachability-distance indicates a noise object or a jump from one cluster to another cluster. Each cluster should hold different characteristics of the concept. The differences are caused by variations of conditions which videos taken under (viewpoints, scenes, illumination and so on) or diversity in meaning of the concept itself.

3.3 Shot Selection

For each obtained cluster, we assign outlier factor for each shot based on outlying property relative to its surrounding space. Here surrounding space of a shot implies a group of other shots which are visually similar to it as illustrated in Fig. 2. Differently from shot clustering step, in this step surrounding space of a shot is limited within in its own cluster. We use calculation method of LOF (Local Outlier Factor) proposed in [10]. There are numerous methods of outlier detection which have been proposed so far in the literature [24]. Among those, LOF is one of the most efficient and easy-to-implement. Especially, it makes use of computation during clustering ($k-\text{dist}, N_{k-\text{dist}}$). Therefore, we chose it to simplify the calculation process. Actually, the combination of OPTICS and LOF is quite natural and has been employed in some previous work [25]. LOF of a point p is formally defined as follows.

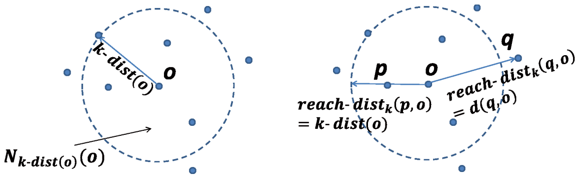

Fig. 2 k-distance and reachability distance (k = 4)

$$LOF_{\text{MinPts}}(p) = \frac{\sum_{o \in N_{\text{MinPts-dist}(p)}(p)} \frac{\text{MinPts}-\text{dist}(p)}{\text{MinPts}-\text{dist}(o)}}{|N_{\text{MinPts}-\text{dist}(p)}(p)|} \tag{1}$$

LOF of an object is calculated as the average ratio of its MinPts − dist and that of its neighbors within MinPts − dist. A large MinPts − dist corresponds to a sparse region since the distance to the nearest MinPts neighbors is large. In the contrast, a small MinPts − dist means that the density is high. In each cluster, shots are ranked according to LOF. Shots with low LOF degrees are considered as relevant shots and brought to the top of the cluster. MinPts is the most important parameter for finding clusters and calculating LOF. Larger MinPts means more clusters. Optimized value of MinPts varies on the concept. In our experiments, we try several values and report the one with the best performance on average.

We propose a simple shot selection strategy which can guarantee that shots are selected from all clusters. Let N_s be number of shots we want to collect for a concept and N_c be number of clusters we obtained. Since some shots are shared among some clusters, simply selecting top N_s/N_c shots from each cluster obtains less than N_s shots. Our selection strategy tries to keep selecting shots from clusters which are still available until number of selected shots reaches N_s or no available clusters left. An "available cluster" must have more shots than twice of its maximal number of shots to be selected. This definition of available cluster is inspired by experimental results in the baseline [6] which show that only shots ranked among top-half should be considered as relevant shots. Selection order for clusters is determined by the mean LOF of their shots. Our selection strategy is summarized in Algorithm 1.

In Algorithm 1, N_t and N_m represent the total number of selected shots and the maximal number of shots can be selected from each cluster, respectively. $\mathbb{C} = \{C(c) | c = 1 : N_c\}$ is the group of obtained clusters. Each cluster $C(c)$ has following fields: $C(c).is$ means index of start-to-select shot, $C(c).ns$ means the number of shots to select from $C(c)$, $C(c).ts$ is the total number of shots in $C(c)$ and

Algorithm 1 Shot selection

```
$N_t \leftarrow 0$
$N_m \leftarrow N_s/N_c$
for $c = 1$ to $N_c$ do
    $C(c).is \leftarrow 1$
    $C(c).av \leftarrow 1$
end for

while $N_t < N_s \& \exists c : C(c).av = 1$ do
    for $c = 1$ to $N_c$ do
        if $C(c).av = 1$ then
            if $C(c).ts > 2 * N_m$ then
                $C(c).ns \leftarrow N_m$
            else
                $C(c).ns \leftarrow C(c).ts/2$
                $C(c).av \leftarrow 0$
            end if
        end if
    end for
    for $c = 1$ to $N_c$ do
        for $i = C(c).is$ to $C(c).ns$ do
            if $C(c).S[i] \notin \mathbb{S}$ then
                $push(C(c).S[i], \mathbb{S})$
                $N_t \leftarrow N_t + 1$
            end if
        end for
    end for
    for $c = 1$ to $N_c$ do
        $C(c).is \leftarrow C(c).is + C(c).ns$
    end for
    $N_m \leftarrow N_m + (N_s - N_t)/N_c$
end while
```

$C(c).av$ represent the availability of $C(c)$. If $C(c)$ is available, $C(c).av = 1$, otherwise $C(c).av = 0$. Collection of shots in $C(c)$ is denoted as $C(c).S$. Since shots are ranked as mentioned above, $C(c).S[1]$ is supposed to be the most relevant shot and $C(c).S[C(c).ts]$ should be the least relevant one in cluster $C(c)$. $\mathbb{S}$ is the collection of selected shots.

4. Experiments and Results

4.1 Experimental Setup

We conduct two experiments: dataset construction and action recognition to validate the efficiency of our method. For dataset construction, we use 11 actions defined in UCF YouTube Action (UCF11) dataset [11]: "basketball shooting", "biking/cycling", "diving", "golf swinging", "horse riding", "soccer juggling", "swinging", "tennis swinging", "trampoline jumping", "volleyball spiking", and "walking with a dog". Note that in this experiment, we do not use videos of that dataset. Our videos are automatically collected from Web source (YouTube) as described in Sect. 3.2. As for action recognition experiment, we use videos of that dataset which contains a total of 1168 videos. The dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination conditions. We train three SVM multi-class classifiers: one based on our collected data, one based on data retrieved by the baseline [2] and one based on standard training data. Finally, we use these classifiers to perform action recognition on the standard test data.

Our baseline is our previous work [2]. According to this method, first videos are ranked based on usage frequencies of tags. Shots are collected from videos which have tags with high co-occurrence frequencies. Next shots are ranked using VisualRank [8] which is a ranking method with a visual feature based similarity matrix. Since it became hard to obtain tag information, we could not perform tag co-occurrence based video ranking step as proposed in the baseline. Here we use our method of shot collection and apply VisualRank to shot ranking to compare the baseline with our proposed method of shot selection which composed of diversity based shot clustering and LOF based shot ranking. We show that our method proposed in this paper can obtain higher precision rate for most of experimented actions and importantly, our results look more diverse than those by the baseline in all cases.

As distance metric, we use Rank-order distance [26] which has been demonstrated as a better density measurement than commonly used Euclidean distance [17]. We train a temporal ConvNet using UCF101 dataset [11] (split 1) following the approach proposed in [13] except that we insert a normalization layer between pool2 layer and conv3 layer. Using this modified network architecture, we obtained slightly better performance than the original one: 82.1% versus 81.2% [13] on UCF101 (split 1). We use 2048 dimensional full7 features extracted using the trained temporal ConvNet. MinPts is set as T/n where T is total number of shots for a concept obtained after shot collection step (Sect. 3.1) and n is a constant. Ten values of n are experimented: $10, 20, \ldots, 100$.

4.2 Dataset Construction

In this experiment, we want to validate the quality of automatically constructed dataset regarding precision and diversity. Precision rate is calculated as percentage of relevant shots among top N shots. Three values of N are taken into consideration: 30, 50, 100 following the baseline [2]. We evaluate relevance of top ranked shots manually.

First we examine the effect of parameter settings on the performance of our proposed approach. Figure 3 shows average precision rates in different cases of n with $N = 100$. According to our empirical results, $n = 50$ obtained best performance. All results related to our proposed method that we report from here on refer to the case of $n = 50$. Note that n = 50 was only optimal regarding average precision of action concepts considered in this paper. It is hard for us to explain why n = 50 obtained the best average precision. Actually, n = 50 was not optimal for all experimented action concepts. The results vary for each concept due to their diversity as well as many different characteristics of their data including the total number of shots obtained (T), the quality of the raw data (some concepts are popular on the web so their data are rich while some are not) and so on.

Figure 4 compares average precision rates in different

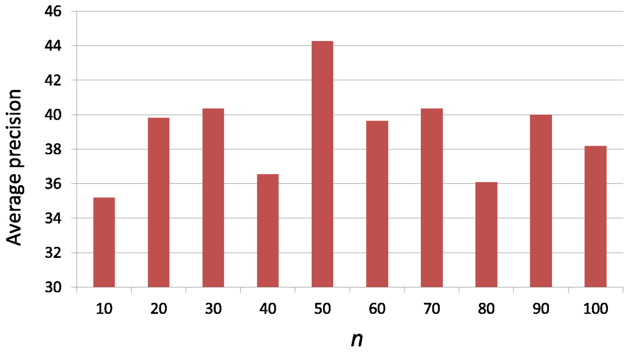

Fig. 3 Average precisions by the proposed method in different cases of n when $N = 100$.

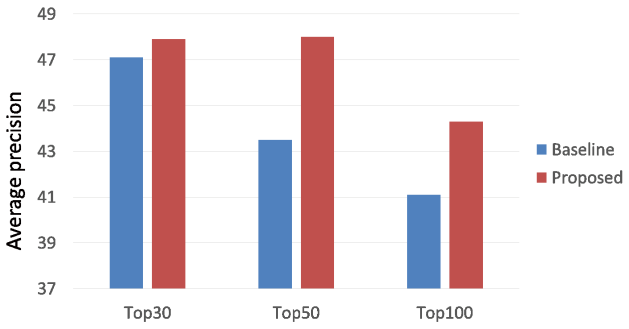

Fig. 4 Average precisions by the proposed approach and the baseline in different cases of N.

Table 1 Precision rates of 11 action keywords with $N = 100$.

Action	Proposed	Baseline
basketball	**50**	35
biking	**23**	17
diving	**35**	28
golf_swing	52	**54**
horse_riding	**50**	42
soccer_juggling	**68**	63
swing	**36**	31
tennis_swing	47	**51**
trampoline_jumping	**54**	**54**
volleyball_spiking	58	**69**
walking	**14**	9
Average	**44.3**	41.1

cases of N between the proposed approach and the baseline. As shown in this figure, the proposed approach outperformed the baseline in all cases of N, especially when $N \geq 50$. In all of our results, "Baseline" corresponds to VisualRank based shot ranking with our shot collection and "Proposed" means our approach with $n = 50$. Note that n was optimized using not the testing data but the collected data. In our paper, testing data imply testing data of standard data (manually collected data) which were employed only in action recognition experiments (Sect. 4.3). Precision for all actions when $N = 100$ are shown in Table 1.

As shown in Table 1, for most of the actions, more relevant shots could be ranked to the top using our method. In many cases, top ranked shots by the baseline are although relevant to the concept but actually look similar to each other (See Fig. 7 for some example results). Even though average precision is not significantly improved, shots retrieved by our proposed method look much more diverse as shown in the followings.

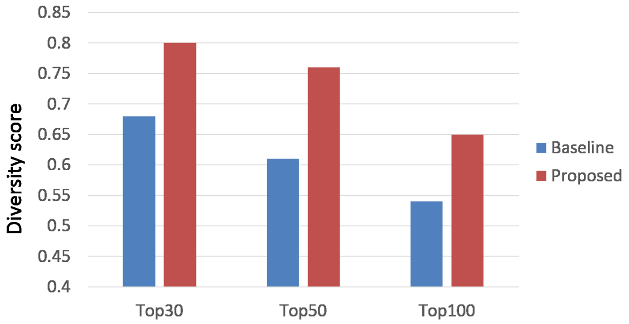

Fig. 5 Results of diversity evaluation. As opposed to the baseline, our approach retrieved more diverse shots from various videos. This explains for significant improvements in recognition performance (Fig. 6).

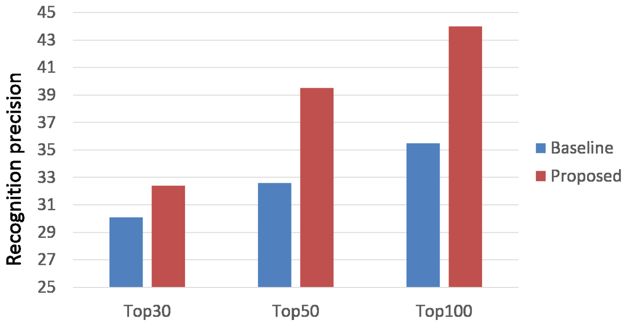

Fig. 6 Results of action recognition with automatically obtained training data. As shown in this figure, action model trained with shots obtained by the proposed method achieved better recognition rates in all cases of N.

Regarding evaluation for diversity of ranking results, we use evaluation method as described in [7]. Diversity score of a ranking result is defined as the ratio of the number of identical videos in its top ranked N video shots to N. This definition is based on the fact that shots from the same video tend to look similar. The results of diversity evaluation are summarized in Fig. 5. As shown in Fig. 5, overall, the diversity score was significantly enhanced by using the proposed method. Figure 7 shows some examples of relevant shots among top 15 shots obtained by our method and the baseline.

4.3 Action Recognition

In this experiment, we validate performance of our automatically collected training data for the task of recognition on standard test data. To evaluate recognition performance, we follow the original setup [11] using leave one out cross validation for a pre-defined set of 25 folds. Average accuracy

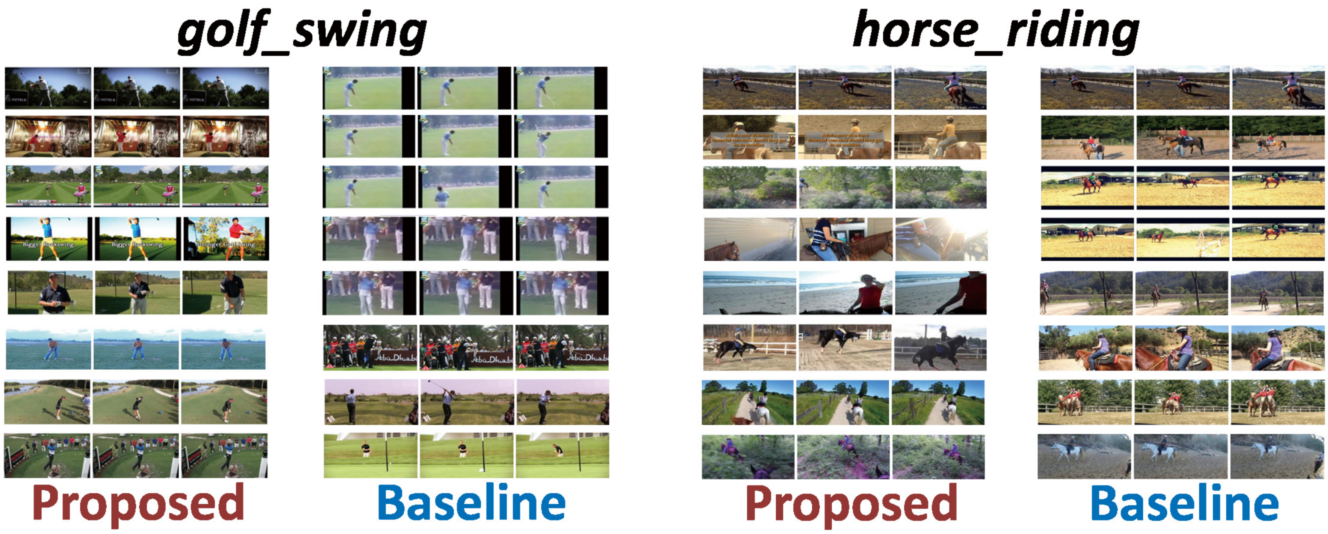

Fig. 7 Relevant shots among top 15 shots retrieved by our proposed method and the baseline for “golf_swing” and “horse_riding”. As seen here, shots by the baseline tend to look similar while shots by our method are taken from various view points against different background.

over all classes is reported as performance measure. We use top ranked N shots to train action classifiers. Similar to the previous experiment, three values of N are taken into consideration: 30, 50 and 100.

Figure 6 shows recognition accuracy rates by the proposed approach and the baseline in all cases of N. As shown in this figure, we obtained significant precision gain compared to the baseline. The recognition rate was boosted from approximately 35% to 44% in case of $N = 100$. This can be explained mostly by the improvements regarding diversity in the results (Sect. 4.2). Since many shots obtained by the baseline look similar, the information we can gain from them is much less than that from shots retrieved by our proposed approach. These results verified the fact that a high quality action database should reflect as well as possible the diversity of the concepts. The precision rate is further improved as more top-ranked shots are used to train.

In case of using standard training data instead of our automatically collected training data, we obtained recognition rate 81.5%. This result is comparable to other approaches on the same dataset [18], [27]. [27] with probabilistic fusion of multiple motion descriptors and scene context descriptors achieved 73.2%. Especially, the state-of-the-art motion hand-crafted features (dense trajectory based MBH) [18] achieved 80.6%. To the best of our knowledge, there are no reports with CNN features on this dataset for us to compare. These results still show a huge gap between standard data (automatically collected data) and manually collected data. However, our method offers the advantage of a fully automatic, scalable learning which is expected to encourage the development of approaches for automatic construction of large-scale action databases.

5. Conclusions

In this paper, we proposed a fully automatic approach for action dataset construction with noisy Web videos. Our approach aims to solve the problem of limitation in quantity of training data for the task of action recognition. In our experiments, we first constructed a database for 11 actions in UCF11 dataset using YouTube videos with our proposed approach. We then employed this database to train action classifiers and applied them to classify standard test data of UCF11. Even though our collected training data are still far from manually collected training data, our method offers the advantage of a fully automatic, scalable learning and shows the potential for development of approaches for automatic construction of action databases.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 26011435.

References

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” Proc. IEEE Computer Vision and Pattern Recognition, Alaska, USA, pp.1–8, June 2008.
[2] N.H. Do and K. Yanai, “Automatic extraction of relevant video shots of specific actions exploiting Web data,” Computer Vision and Image Understanding, vol.118, no.1, pp.2–15, 2014.
[3] A. Ulges, C. Schulze, M. Koch, and T.M. Breuel, “Learning automatic concept detectors from online video,” Computer Vision and Image Understanding, vol.114, no.4, pp.429–438, 2010.
[4] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” Proc. IEEE Computer Vision and Pattern Recognition, Florida, USA, pp.2929–2936, June 2009.
[5] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, “Automatic

annotation of human actions in video," Proc. IEEE Computer Vision and Pattern Recognition, Florida, USA, pp.1491–1498, June 2009.

[6] N.H. Do and K. Yanai, "Automatic Construction of an Action Video Shot Database using Web Videos," Proc. IEEE International Conference on Computer Vision, Barcelona, Spain, pp.527–534, Nov. 2011.

[7] H.N. Do and K. Yanai, "VisualTextualRank: An extension of VisualRank to Large-Scale Video Shot Extraction Exploiting Tag Co-occurrence," IEICE Transactions on Information and Systems, vol.E98-D, no.1, pp.166–172, 2015.

[8] Y. Jing and S. Baluja, "VisualRank: Applying PageRank to Large-Scale Image Search," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30, no.11, pp.1870–1890, 2008.

[9] M. Ankerst, M.M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," Proc. ACM SIGMOD International Conference on Management of Data, New York, USA, pp.49–60, June 1999.

[10] A.L.M. Chiu and A.C. Fu, "Enhancements on local outlier detection," Proc. IEEE Database Engineering and Applications Symposium, HongKong, China, pp.298–307, July 2003.

[11] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," Proc. IEEE Computer Vision and Pattern Recognition, Florida, USA, pp.1996–2003, June 2009.

[12] S. Khurram, R.Z. Amir, and S. Mubarak, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild," CoRR, vol.abs/1212.0402, 2012.

[13] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," Proc. Advances in Neural Information Processing Systems, pp.568–576, Canada, Dec. 2014.

[14] X. Chen, A. Shrivastava, and A. Gupta, "Neil: Extracting visual knowledge from web data," Proc. IEEE International Conference on Computer Vision, Sydney, Australia, pp.1409–1416, Dec. 2013.

[15] B. Collins, J. Deng, K. Li, and L. Fei-Fei, "Towards Scalable Dataset Construction: An Active Learning Approach," Proc. European Conference on Computer Vision, pp.86–98, Marseille, France, Oct. 2008.

[16] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting Image Databases from the Web," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.33, no.4, pp.754–766, 2011.

[17] Y. Xia, X. Cao, F. Wen, and J. Sun, "Well Begun Is Half Done: Generating High-Quality Seeds for Automatic Image Dataset Construction from Web," Proc. European Conference on Computer Vision, Zurich, Switzerland, pp.387–400, Sept. 2014.

[18] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," Proc. IEEE International Conference on Computer Vision, Sydney, Australia, pp.3551–3558, Dec. 2013.

[19] D. Oneata, J. Verbeek, and C. Schmid, "Action and Event Recognition with Fisher Vectors on a Compact Feature Set," Proc. IEEE International Conference on Computer Vision, Sydney, Australia, pp.1817–1824, Dec. 2013.

[20] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action Recognition with Stacked Fisher Vectors," Proc. European Conference on Computer Vision, Zurich, Switzerland, vol.8693, pp.581–595, Sept. 2014.

[21] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2625–2634, 2015.

[22] J.Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4694–4702, 2015.

[23] E. Golge and P. Duygulu, "Conceptmap: Mining Noisy Web Data for Concept Learning," Proc. European Conference on Computer Vision, Zurich, Switzerland, pp.439–455, Sept. 2014.

[24] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," Proc. ACM Computing Surveys, vol.41, no.3, pp.15:1–15:58, 2009.

[25] M.M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander, "OPTICS-OF: Identifying local outliers," Proc. European Conference on Principles of Data Mining and Knowledge Discovery, Prague, Czech Republic, vol.1704, pp.262–270, Sept. 1999.

[26] C. Zhu, F. Wen, and J. Sun, "A rank-order distance based clustering algorithm for face tagging," Proc. IEEE Computer Vision and Pattern Recognition, Colorado Springs, USA, pp.481–488, Juny 2011.

[27] K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," Machine Vision and Applications, vol.24, no.5, pp.971–981, 2013.

Nga Hang Do received B.Eng., M.Eng. and D.Eng degress from the University of Electro-Communications, Tokyo in 2011, 2013 and 2015, respectively. She is currently a PDRF at Department of Informatics, the University of Electro-Communications, Tokyo.

Keiji Yanai received B.Eng., M.Eng. and D.Eng degrees from the University of Tokyo in 1995, 1997 and 2003, respectively. From 1997 to 2006, he was a research associate at Department of Computer Science, the University of Electro-Communications, Tokyo. He is currently a professor at Department of Informatics, the University of Electro-Communications, Tokyo. His recent research interests include object recognition and Web multimedia processing. He is a member of IEEE Computer Society, ACM, JSAI and IPSJ.