The University of
Electro-Communications

Master Thesis

Automatic Construction and Analysis of Large-scale Action Shot Database Exploiting Web Data

Author:
Nga Hang Do

Supervisor:
Prof. Keiji Yanai

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Media Science and Engineering Program
Department of Informatics

January 30 2013

Abstract

Department of Informatics
The University of Electro-Communications

Master of Science

Automatic Construction and Analysis of Large-scale Action Shot Database Exploiting Web Data

by Nga Hang Do

Video sharing websites have recently become a tremendous video source, which is easily accessible without any costs. This has encouraged researchers in the action recognition field to construct action database exploiting Web sources. However Web sources are generally too noisy to be used directly as a recognition database. Thus building action database from Web sources has required extensive human efforts on manual selection of video parts related to specified actions. In this paper, we introduce a novel method to automatically extract video shots related to given action keywords from Web videos according to their metadata and visual features. First, we select relevant videos among tagged Web videos based on the relevance between their tags and the given keyword. After segmenting selected videos into shots, we rank these shots exploiting their visual features in order to obtain shots of interest as top ranked shots.

Especially, we propose to adopt Web images and human pose matching method in shot ranking step and show that this application helps to boost more relevant shots to the top. Furthermore, we introduce a novel ranking method called VisualTextualRank which analyzes the visual link structures between video shots simultaneously with the textual link structures between videos and their tags and appply this method to shot ranking step.

Our unsupervised system of extracting action video shots only requires the provision of action keywords such as “surf wave” or “bake bread” at the beginning. We have done large-scale experiments on various kinds of human actions as well as non-human actions and obtained promising results.

Acknowledgements

This research thesis would not have been possible without the support of many people. I wish to express my gratitude to my supervisor, Prof. Keiji Yanai who was abundantly helpful and offered invaluable assistance, support and guidance. Special thanks also to all my graduate friends for sharing the literature and invaluable assistance. Not forgetting to my best friends who always been there. Especially, I would also like to convey thanks to the Ministry and Faculty for providing the financial means and laboratory facilities. Finally, I wish to express my love and gratitude to my beloved families; for their understanding & endless love, through the duration of my studies.

Contents

Dedicated To My BimBo and ChimCu

Chapter 1

Introduction

The explosion of the Internet as well as the rising need of sharing information between people on the Internet has made it a huge and unstoppably growing data source. People upload to the Internet every kind of data including images, music and videos. YouTube is one of the most popular video sharing websites which allow people to easily upload their own videos and access to those of others. By using Web API like YouTube API, we can obtain a large number of videos of various topics from Web sources without any difficulties. Especially, many of the topics are related to human and human actions; therefore, there is an increasing tendency for action recognition researchers to construct human action database exploiting Web videos.

When users upload their videos, they usually attach to the videos keywords called as “tags”- useful metadata for video retrieval. However, in general, tags are annotated to the whole video sequence, not to specific scenes. Therefore, it can not be determined which tag corresponds to which part of the video. For example, some videos tagged “eat” might include not only the eating scene but also such other scenes as entering restaurants, ordering foods, or drinking something (See Figure 1.1). People who want to search for eating scenes have to manually skip the scenes of no interest while carefully watching the whole video. Moreover, in some cases such as tags are irrelevant to the keyword since they are supplied subjectively by general users, or the keyword itself is ambiguous, the tag based search results

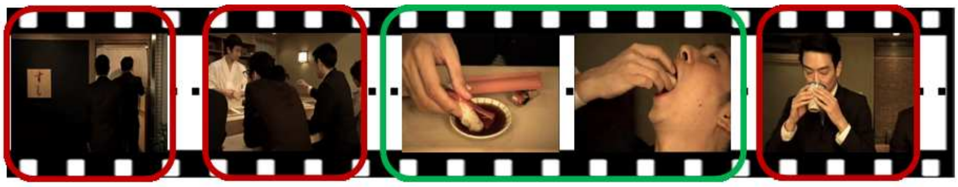

FIGURE 1.1: A video obtained by searching with "eat sushi" keyword on YouTube. It contains scenes of interest (scenes with green bounding box) describing "eat sushi" action as well as irrelevant scenes (scenes with red bounding box) describing actions of entering restaurant, ordering sushi and drinking tea (respectively from the left to the right). Researchers who need only training data for "eat sushi" action must watch the whole video carefully to find its relevant scenes. Extracting relevant scenes in an unsupervised way is the purpose of this paper.

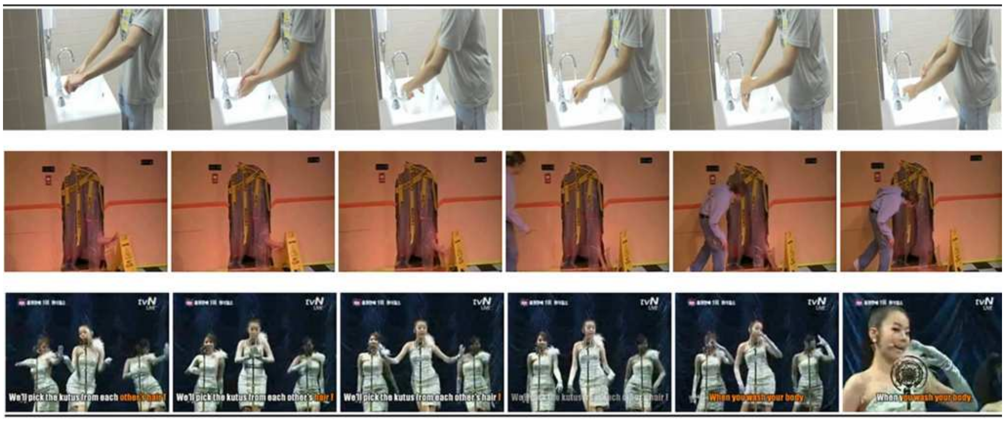

FIGURE 1.2: Some videos obtained by searching with "wash hand" keyword on YouTube. We expected videos which contain scenes of people washing their hands like the top one. Howerver, many search results like the other ones have no scenes of interest even though they have "wash hand" as their tags. The second video is a part of a comedy with title as "Employees Must Wash Hands...Before Murder". The bottom video is a video clip of the song named as "Wash your hands too".

may include many unrelated videos (See Figure 1.2). Due to above reasons, Web videos based database construction for specific actions become a very troublesome and time-consuming task.

In this paper, we propose a new method to automatically extract from tagged Web videos relevant video shots of specific actions using metadata as well as visual context of these videos. Note that video shots here refer to small fragments of a video obtained by separating it at each point of a scene or camera change. Our unsupervised method requires only the provision of action keywords at the beginning.

As for keywords, we mainly focus on words related to human actions. Our list of human action keywords contains sport activities such as “serve volleyball” or “row dumbbell” as well as activities of daily living like “shave mustache” and “tie shoelace”. The list also includes some music related activities like “play trumpet” and “dance flamenco” or emotion related activities like “slap face” and “cry” as the consequence of “being angry” and “being sad” respectively. Moreover, we also tried several non-human actions such as “flowers bloom” or “leaves fall”. We want to demonstrate that our proposed system can be applied to extract relevant video shots of various types of actions from the Web.

If video shots corresponding to any “action verb” can be acquired automatically from unconstrained videos like Web videos, we can build easily training database for action recognition. So far, as mentioned above, the construction of action training data has been known as exceptionally expensive work, which is totally different from building object recognition database. In fact, while object recognition categorizes up to 10,000 objects, the largest widely used action recognition dataset includes only 51 human action categories[1]. On the other hand, by applying our method, video shots associated with unlimited types of actions can be easily collected from Web sources. In addition, the proposed method can be applied to improve Web video searching and tagging.

Our main idea is at first, selecting relevant videos among thousands of Web videos for specified action and then, extracting the most related shots from selected videos. The video selection step is based on our assumption that videos tagged with many relevant words have high probability of being relevant videos so they should be selected. For the extraction of corresponding shots, we apply an efficient unsupervised ranking method called VisualRank[2]. We made large-scale experiments on 100 human action keywords and 12 non-human action keywords. The experimental results reflect the effectiveness of our system as we achieved high precision for many keywords. Note that here precision is considered as the percentage of relevant shots among top ranked 100 shots (Precision@100).

Furthermore, we proposed to take still Web images corresponding to given actions into account, with the intuition that the shots with more similarity to related action images have higher probability of being relevant shots, thus they should be biased in shot ranking. In fact, recent works[3–6] show that action recognition exploiting still images is possible. We collect images related to the given actions automatically via Web image search engines based only on provided keywords and measure visual resemblances between video shots and selected images. Shots with higher similarity scores will have higher chance to be ranked to the top. Note that these Web images involved processes also do not require any supervision, therefore the automaticity of the whole framework can be preserved. We verify the efficiency of introducing Web images by applying Web images exploited framework on 28 human actions and 8 non-human actions with precision achieved by original framework respectively lower than 20% and 15%. The results demonstrate that exploiting Web action images can significantly improve the performance of the original system.

Moreover, we further enhanced our system by exploting efficiently the correlation between the videos and their tags. Our intuition is that, when tags are noisy, exploiting solely their co-occurrence frequencies to evaluate their relevance to the given keyword is not effective enough. Tags are supposed to be more efficiently employed if they are ranked considering their correlation with corresponding videos. For example, if we find that a video shot is important, or in other words, related to given action keywords, so that words tagged to its video have high chance to be important as well. And vice versa, if a tag was found as being relevant to the keyword, it is highly probable that the videos annotated with it are also relevant. We conducted co-analysis of visual links among video shots along with textual links between videos and their tags. This analysis can be done through the shot ranking step and we call our ranking method which performs this analysis as VisualTextualRank (abbreviated as VTR). This novel ranking method of ours extends [7], [8], [2] and [9], and effectively uses both textual information and visual information extracted from Web videos. Based on our experimental results, we demonstrate that VTR can significantly improve the precision of ranking results.

Our proposed system can help reduce tremendous human effort on building database for action recognition. Although a few modest manual scanning may still be needed to use these video shots as training data, there is no doubt that human effort can be significantly reduced in comparison to fully manual database construction. The contributions of this thesis can be enumerated as follows:

(1) An automatic system of extracting relevant video shots of specific actions from the Web which enables us to construct large-scale action video shot database

(2) A novel ranking method which analyzes simultaneously visual links among video shots along with textual links between videos and their tags

In the next chapter, we refer some related works on action recognition, web image mining and tag ranking, respectively. In Chapter 3, we describe our unsupervised system of extracting relevant video shots from Web videos. We report the results of this thesis in Chapter 4 and finally, conclude our work in Chapter 5.

Chapter 2

Related work

2.1 Action Recogntion

For the past five years, ST (Spatio-Temporal) features which describe both spatial and temporal description of movement, and their BoF (Bag of Features) representation, due to their effectiveness, have been exploited by many researches on human action recognition and content-based video analysis. By using BoF of ST features, action recognition problem can be almost regarded as the same problem as object recognition except for feature extraction process. One of recent works which adopt this methodology is Jones et al.'s work[10]. The novelty in their work is that the results are refined based on users' relevance feedback following previous works in the image domain.

Nevertheless, the BoF model suffers from some limitations, one of which is the loss of some discriminative information in both spatial and temporal dimensions. As one of other effective models of human action recognition, a dense representation proposed by Zhen et al.[11] takes into account the motion and structure information simultaneously. In this work, high dimensional features are first extracted and then embedded into a compact and discriminative representation by DLA (Discriminative Locality Alignment) method. On the other hand, instead of using all frames in the video sequence, Liu et al.[12] proposed to learn the most representative frames called as key frames by AdaBoost algorithm and represent action by the probabilistic distribution and temporal relationships of these frames.

The above mentioned works aimed to label to the whole content of each test video sequence one of the pre-defined categories, while our objective is to search among a large number of Web videos for only video parts which are associated with the given keywords. Moreover, while our proposed system is unsupervised, all of the above works apply supervised learning method which implements SVM (Support Vector Machines) as the final classifiers. SVM is widely used in computer vision and machine learning in general and pattern recognition in particular.

Beside the supervised methods, there have been several attempts in unsupervised action recognition. Niebles et al.[13] categorized action videos in KTH datasets and their original ice-skating video data adopting the PLSA (Probabilistic Latent Semantic Analysis) model. Niebles et al.[14] also proposed a method to extract human action sequences from unconstrained Web videos. Cinbis et al.[15] proposed a method to learn action models automatically from Web images gathered via Web image search engines, and recognize actions for the same video dataset as[14]. Although Cinbis et al. 's work is the most similar to our work, they exploit only Web images and static features as a training source, while Web videos and spatio-temporal features are also adopted in our work. In addition, while works by both Niebles et al. and Cinbis et al. aim to recognize only human actions, our method does not restrict its applicability within any type of actions. Our method might be applied to collect relevant video shots of non-human actions such as "airplane fly" and "tornado".

As in another similar work, Ballan et al.[16] proposed a method to add tags to video shots by using Web images obtained from Flickr as training samples. Meanwhile, Laptev et al.[17–19] proposed methods to automatically associate movies and movie scripts. Their methods also enable the construction of an action shot database in an unsupervised manner, although targeted videos are limited to only the movies with available scripts.

2.2 Web Image Mining

Regarding still images, many works on automatic construction of image database exploiting images gathered from the Web have been carried out so far[20–25]. Most of these works use object recognition methods to select relevant images to specified keywords from "raw" Web images collected using Web image search engines. In fact, Web images acquired through Web image search engines like Google Image Search are not really so "raw" regarding their relevance to the given keywords. Google so far have applied a technology called VisualRank[2] to a group of initial search results. According to VisualRank, images found to share the most visual characteristics with the group at large would shall be determined as the most relevant ones and brought to the top of search results. To apply this idea to action video shots detection is our initial motivation of this work. Hence our work can be regarded as video shot version of the automatic search for relevant Web data of a given keyword.

2.3 Tag Ranking

In this paper, we perform tag analysis to compute tag based relevance scores. Having tags is a common characteristic of CGM (Consumer Generated Media) data on the Web. Users are recommended to tag their uploaded media data with some words related to the data content so that other users can search for them. As efforts on tag ranking considering their relevance, Yang et al.[26] proposed a method to evaluate a tag relevance score on each tag based on tag co-occurrence statistics which is called as "Web 2.0 Dictionary". This method requires only tag analysis and no visual information. We apply this method to initially search for relevant Web videos. As a similar method which does not require visual features, Dong et al.[27] proposed a method to evaluate tag relevance score by combining the probabilistic relevance score estimation and random walk-based refinement. Although this two-step method is similar to ours, they use only tag information, while we exploit both tag and visual features.

As another related work, Liu et al.[9] presented a Web video topic discovery and tracking method via bipartite graph which represents the correlation between videos and their tags. Actually, their idea is the motivation of our improvement on our adoption of tags. However, they tried to find relevant videos of a topic, while detection of relevant video shots is our objective. Moreover, the main difference between their work and our work in terms of methods is that they used only textual information, while we use both textual and visual features. In this paper, we propose a new method, which is based on random walk over bipartite graph to integrate visual information of video shots and tag information of Web videos effectively.

Chapter 3

Proposed Framework

In this chapter, we introduce our enhanced system which automatically extracts from tagged Web videos video shots corresponding to specific actions. This sytem was built based on our previous paper[7]. We first present the objective as well as the overview of the whole system in Section 3.1 and then go to the detail of our proposed methodology.

3.1 Overview of Proposed System

The objective of the proposed system is explained explicitly in Figure 3.1. From abundant Web videos of an action keyword, we exploit their visual features as well as textual information to obtain only relevant video shots of that keyword. Figure 3.2 illustrates the overview of the proposed system. Our system consists of four following processing steps:

1. Video selection and video-tag relevance calculation
2. Shot segmentation and shot similarity measurement
3. Image selection and shot-image similarity calculation (option)
4. Shot ranking

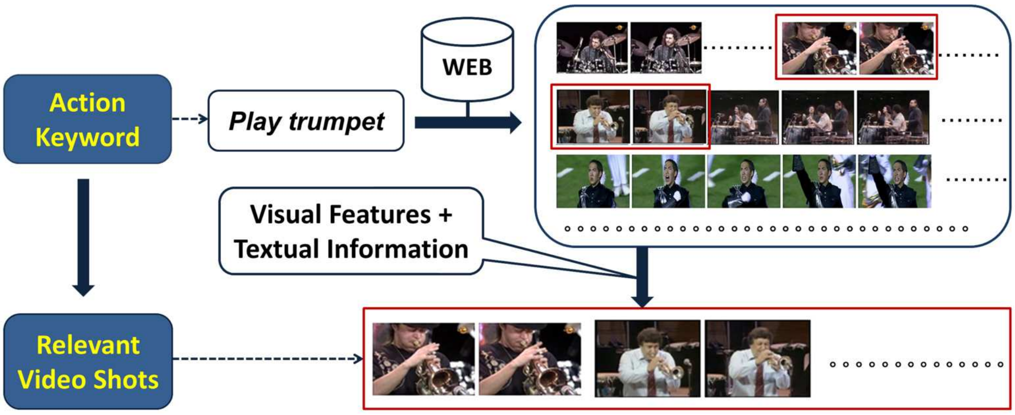

FIGURE 3.1: Illustration of our objective. When we search for videos of a given action keyword (as "play trumpet" in this example) using a video search engine like YouTube, we can obtain bunch of videos including relevant ones as well as irrelevant ones. Even the relevant ones may contain unrelated parts. In this example, playing trumpet is just one section of an instrumental performance. Thus the videos may consist of many irrelevant sections such as playing drum and playing piano. Our objective is to extract only relevant video parts of the given keyword (parts which are surrounded by red bounding box) *in an unsupervised manner*.

In the first step, video IDs and tags for at most 1000 Web videos of search results for the action keyword are collected via Web API. The co-occurrence frequencies among tags are exploited to build a database of tag relevance information. Then videos are ranked in the descending order of their tag relevance scores with the keyword. Only the top ranked videos are downloaded since they are considered as action related videos. Meanwhile, the relevance scores of the videos to their tags are also calculated by similar way to calculate relevance of videos to the keyword.

In the second step, the downloaded videos are segmented into video shots using color information. Spatio-temporal features are extracted from all shots and used to calculate similarity matrix of shots.

The third step is an option. In this step, firstly, hundreds of top results of image search for given action keywords are downloaded using Bing API. Then, Web action images are automatically selected based on human detection method. Finally, similarity scores between shots and images are measured according to their static features. Note that human detected images are selected and images with no human detected are discarded only in case of human actions. In case of non-human actions, images directly retrieved by Bing API are adopted. The third step can be performed in one of two modes: for shots and images, (1) SURF features are extracted, and shot-to-image similarities are measured using feature matching. (2) Simple but efficient pose features which simulate the orientation of human body parts are extracted, and shot-to-image similarities are measured by comparing their pose features.

Both modes can be applied to human actions while the first mode is restricted to non-human actions only. Note that as for shots, we do not extract pose features from all of their frames but only one frame at every second since normally there is no significant change in one second. This also helps to reduce the cost of calculation.

In the final step, we rank video shots by our proposed ranking method which employs both textual similarities between videos and tags as well as visual similarities between shots. In our improved system, action relatedness scores of video shots and tags are updated iteratively by employing not only co-occurrence relevance between tags and corresponding videos but also more trustworthy information, content-based features which are extracted from videos. In the end, we can obtain video shots corresponding to the given keywords in the upper rank of the video shot ranking results.

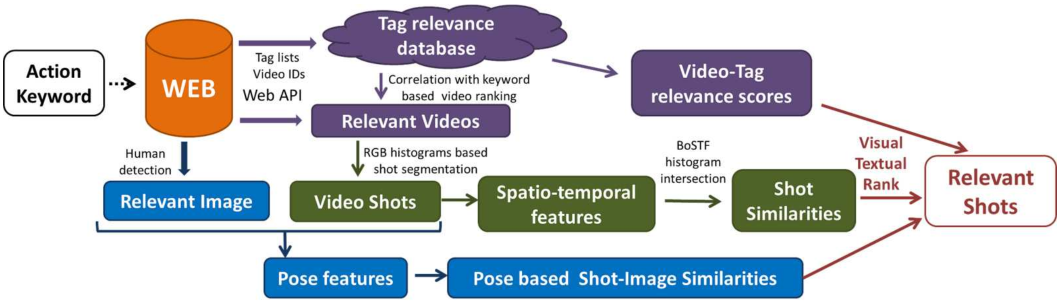

Figure 3.2: Overview of unsupervised system of extracting corresponding video shots for specific actions from Web videos. We modified our previous system[7] by introducing Web images and a novel ranking method to enhance shot ranking process.

3.2 Methodology

3.2.1 Tag-based Video Selection

Web videos associated with the given keywords can be obtained easily by using Web API. In case of YouTube, they provide YouTube API to search in their video database for the videos tagged with the given query words. However, since tags are assigned subjectively by the uploaders, sometimes tags are only weakly related or unrelated to the corresponding videos. The objective of this step is to select the more query-related videos to download.

Firstly, the given keywords are sent to the Web API to collect sets of video IDs and tags. Then the relevance scores of Web videos to the given keyword are calculated according to co-occurrence relationships between their tags. To this end, we apply the "Web 2.0 Dictionary" method proposed by Yang et al.[26] with some modifications in relevance measurement. "Web 2.0 Dictionary" corresponds to statistics on tag co-occurrence, which we need to construct in advance using a large number of tags gathered from the Web. This method is based on an idea that tags other than the query are supporters of the query, and the query can be regarded as being relevant to a video whose tags are its strong supporters.

Assume that $N(t)$ is the number of the videos tagged with word t among all the Web videos, and $\mathcal{T}$ is a set of all the words other than t tagged to all the Web videos. The correlation of parent word t and its child word $t_i \in \mathcal{T}$ is defined as

$$w(t, t_i) = \frac{F(t, t_i)}{N(t)} \tag{3.1}$$

where $F(t, t_i)$ is the number of videos tagged with both word t and word t_i at the same time. Let $\mathcal{T}_V$ represent a set of tags for video V excluding t, we estimate relevance score of video V for word t, $P(V|t)$, by substituting $\mathcal{T}_V$ for V and $w(t, t_i)$

for $P(t_i|t)$ as follows:

$$\begin{aligned} P(V|t) &\propto P(\mathcal{T}_V|t) \\ &= \prod_{t_i \in \mathcal{T}_V} P(t_i|t) \\ &= \prod_{t_i \in \mathcal{T}_V} w(t, t_i) \end{aligned} \tag{3.2}$$

The above equations to calculate relevance of an image video to the given keyword are obtained by applying [26]. However if we multiply all the correlation values between the query tag and the rest of the tags within one video, the value of Equation 3.2 becomes smaller as the number of tags increases. To prevent this, we modify Equation 3.2 so that the number of co-occurrence words used for calculation is limited to m at most, and define the relevance score $Sc_t(V)$ using average log likelihood as follows:

$$\begin{aligned} S(V|t) &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 w(t, t_i) \\ &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} (\log_2 F(t, t_i) - \log_2 N(t)) \\ &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) - log_2 N(t) && (3.3) \\ Sc_t(V) &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) && (3.4) \end{aligned}$$

where $\mathcal{T}'$ contains at most the top m word t_i in the descending order of $w(t, t_i)$, and n $(n \leq m)$ represents $|\mathcal{T}'|$. Since the second term of Equation 3.3 is always the same in the video set over the same action keyword, we omit it and define the relevance score $Sc_t(V)$ as shown in Equation 3.4. In the experiment, we set m as 10, and select the most relevant 200 videos to the given keyword from the 1000 videos returned by the Web API. This tag-based selection in the first step is important to allow only promising videos to go to the next step which requires more costly processes such as feature extraction and similarity calculation.

Note that in case of compound keywords such as “drink coffee”, we regard $N(t)$ as the number of the videos including all of the element word of the compound keyword in their tag sets and $w(t, t_i)$ as the number of videos having all the words of t and t_i even if t_i is also a compound word. We ignore videos which do not have any co-occurrence tag since we can not calculate their relevance scores.

In the experiments, as seed words, we prepared 150 sets of verbs and nouns which are related to such actions as “ride bicycle” or “launch shuttle”. We gathered 1000 video tags for each seed word, and extracted all the tags. As a result, we obtained 12,471 tags which appear more than five times among all the collected tags. For each of 12,471 words, we gathered 1000 video tags again, and constructed “our Web 2.0 Dictionary” by counting tag co-frequencies according to Equation 3.1.

3.2.2 Relevant Shot Extraction: VisualTextualRank

Here we introduce our novel ranking method which aims to rank the relevant video shots to the top exploiting both their visual features and metadata (tags). The basic idea of our ranking method is that the relevant tags are used to annotate relevant videos; the relevant video shots are from videos annotated with relevant tags and visually similar to each other. Thus tags and video shots are co-ranked so that at each iterative ranking step, ranks of shots are refined using their visual similarities as well as their relevance with corresponding tags, and then, ranks of tags are updated based on their relevance with video shots in conjunction with refined ranking positions of video shots. Figure 3.3 sketches our idea.

Our ranking method, VisualTextualRank, is an extension of VisualRank[2] with idea inherited from Liu et al.’s work[9]. In [9], tags and videos are also co-ranked using their correlation to refine their relevance with specific topic. However, in [9], relevance of the whole video, not particular scene, is evaluated and more importantly, content-based features of videos are totally ignored. On the other hand, VisualRank exploits only a visual linkage between images and does not take textual information into account. In VisualRank, the ranking position of the image which looks similar to more images having high ranking position becomes higher and higher after iterative processing.

VTR employs both visual and textual features of Web videos. By applying the anlysis results of visual structure between video shots to tag ranking, we can avoid selecting tags simply because they appear frequently. Thus the effect of noisy tags becomes minor and more relevant tags are employed on shot selection.

The proposed co-ranking method employing the mutual relationships of videos-tags along with video shots-video shots can be represented by following iteration processes:

$$\boldsymbol{RS}_k = \alpha \times SM^* \times SC^* \times \boldsymbol{RT}_k + (1-\alpha)\boldsymbol{p} \tag{3.5}$$

$$\boldsymbol{RT}_{k+1} = (SC')^* \times \boldsymbol{RS}_k \tag{3.6}$$

RS and RT are vectors which represent rank positions of shots and tags, respectively. Let the number of shots be n_s and the number of tags be n_t, the dimension of RS will be $n_s \times 1$ and the dimension of RT will be $n_t \times 1$. SM refers to shot-shot similarity matrix where $SM[i][j]$ means visual similarity score between shot i and shot$j$; SM^* is its column-normalized matrix with size as $n_s \times n_s$. SC represents shot-tag similarity matrix where $SC[i][k]$ measures textual relevance score between the video of shot i and tag k; SC^* is its $n_s \times n_t$ column-normalized matrix. SC' refers to the transpose matrix of SC which represents tag-shot similarity matrix and SC'^* is its column-normalized matrix. Shot ranking uses histogram intersection between spatio-temporal features of shots as visual similarities. Textual relevance scores between videos and tags are calculated using similar method of calculating relevance scores between videos and keyword. Note that since the textual features, here refer to tag co-occurrence, are considered as being noisier than content-based features, we rank video shots first and use their refined ranking positions to update ranks of tags.

RT is initially defined as a uniform vector. At each ranking step, after ranking positions of video shots are updated based on their visual similarities and their correlation with tags following Equation 3.5, video shots cast their votes for tags through Equation 3.6. Thus relevant shots will cast important votes for tags which are strongly connected with them. And then at the next iterative step, those tags again help boost ranking positions for video shots which are tight linked with them. Gradually, video shots and tags having few of important votes casted will go to the bottom. Our co-ranking method extends VisualRank by adopting tags as textual information and keeps reducing the negative effects of noisy tags by using visual feature based relevance update strategy.

Following VisualRank, we also introduce damping factor α and damping vector p into shot ranking. α has been found as holding minor impact on global ordering in ranking results[28]. Thus following our previous paper[7] we choose α as 0.85. As for p we experiment both following definition ways:

$$\boldsymbol{p}_i^{(1)} = [1/n] \tag{3.7}$$

$$\boldsymbol{p}_i^{(2)} = \frac{\exp(\mathrm{SI}(S_i))}{\sum_{j=1}^{n} \exp(\mathrm{SI}(S_j))} \tag{3.8}$$

The uniform damping vector presented in Equation 3.7 is used when we do not employ optional third step. The nonuniform damping vector presented in Equation 3.8 is used when we take Web images into account. So that in this case video shots has similar visual characteristics with corresponding images will be biased during ranking computation. $\boldsymbol{p}_i^{(2)}$ is proportional to corresponding shot-image similarity score $\mathrm{SI}(S_i)$. For the computation of shot-image similarity scores, please refer to Section 3.2.4.

3.2.3 Shot-Shot Similarity Matrix Calculation

In this subsection, we describe how to estimate the similarity matrix which appears in Equation 3.8. In our work, this similarity matrix holds Spatio-Temporal (ST) feature based similarity scores between shots. We first divide each downloaded video into several shots and extract ST features from all the shots. We then

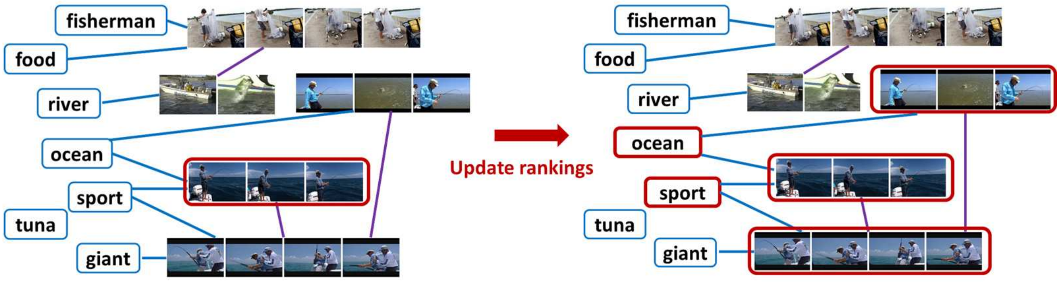

Figure 3.3: Illustration of our proposed co-ranking method. We show here an example of “catch+fish” action. Blue links represent relevance between video shots and tags. Purple links refer to visual similarities between shots. Weak links are not shown here. At first (the right of this figure), the shot marked with red bounding box is considered as being important, so that it will cast its vote for tags which are strongly linked with it. And then at the next step of ranking processes, those tags again cast their votes for video shots which are tight connected with them. Finally we have evaluation results as the left of this figure. Our method iteratively updates ranking positions of both shots and tags in this manner and obtain top ranked shots as relevant ones after converging.

represent each shot as a Bag-of-Spatio-Temporal-Features (BoSTF) histogram and calculate similarity between shots as their histogram intersection.

3.2.3.1 Shot Segmentation

After downloading the most relevant 200 videos to the given keyword regarding tag relevance scores, we segment downloaded videos into video shots based on their RGB histograms. We simply calculate 64 dimensional RGB histogram for each frame and record one segmentation point between two consecutive frames if their histogram intersection is larger than our predefined threshold. As the result, we obtain 10 shots per video on average. However, there are some shots whose duration is too short or too long. It is hard for us to recognize what happens in a shot which lasts too short. In contrast, excessively long shots are supposed to be uninformative since there is no significant change in them. We consider a shot as too short one if its duration is smaller than one second, or too long one if it lasts more than one minute. Thus we select only shots which last longer than one second and shorter than one minute to go to the next step.

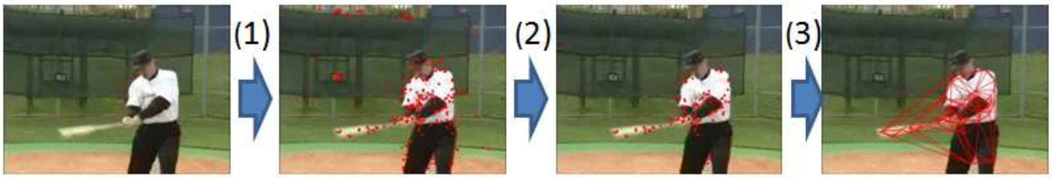

Figure 3.4: Steps to extract the ST feature. (1) detected SURF points, (2) detected SURF points with motion, and (3) obtained Delaunay triangles. (Cited from [29])

3.2.3.2 Spatio-Temporal Feature

Following the method described in our previous paper[29], firstly, interest points are detected using the SURF method[30], and then moving interest points are selected applying the Lucas-Kanade method[31]. Since ST features are supposed to represent movements of objects, only moving interest points are considered as ST interest points and static interest points are discarded. After detection of ST interest points, triples of interest points which hold both local appearance and motion features are formed applying Delaunay triangulation. Then changes of flow directions of interest points as well as the sizes of the triangles are tracked within five consecutive frames. This tracking enables us to extract ST features not from only one point but from a triangle surface patch. Thus the features are expected to be more robust and informative. The ST features are extracted from every five frames. Our proposed method of ST feature extraction is relatively faster than the other methods such as cuboid based method, since it employs SURF detector[30] and Lucas-Kanade detector[31] which are comparatively fast detectors. Figure 3.4 shows an example of the process for extracting the ST features from a video shot of action “batting”.

Our proposed ST features has been demonstrated as being not only fast and easy to implement but also comparative to the-state-of-art[29].

3.2.3.3 Shot-to-Shot Similarity Matrix

To apply our ranking method, we need to compute the similarities among all the shots to find out the shots sharing the most visual characteristics with others. To this end, we first vector-quantize them and convert them into the bag-of-features (BoF). While the standard BoF represents the distribution of local features within one image, the BoF employed in this paper represents the distribution of features within one shot which consists of several frame images. We call our BoF as bag-of-frames (BoFr). In the experiment, we set the size of the codebook as 5000.

The similarity between two shots is measured as their histogram intersection:

$$s(H_i, H_j) = \sum_{l=1}^{|H|} min(h_{i,l}, h_{i,l}) \tag{3.9}$$

where H_i, $h_{i,l}$ and $|H|$ represents the BoFr vector of the i-th shots, its l-th element and the dimension number of the BoFr vector, respectively.

3.2.4 Shot-Image Similarity based Damping Vector Calculation

Remind that employing Web images is an optional step based on our intuition that the shots which are more similar to corresponding action images have higher probability of being relevant shots. So the idea here is to select action images from Web images, calculate the similarities between shots and images, and then bias the shots with high similarities in the shot ranking step. We already reported our work which implements this idea in our workshop paper[8] and our under-review journal article[1].

[1] Do Hang Nga and Keiji Yanai. Automatic Extraction of Relevant Video Shots of Specific Actions Exploiting Web Data. Computer Vision and Image Understanding.

3.2.4.1 Image Selection

When an action keyword is queried on a Web image search engine, thousands of images might be returned. However, in general, even top results may be not relevant images of the queried action due to the wide variety of keyword's meaning as well as the action itself, especially in the case of human action. Here we want to filter the returned results of Web image search engine so that the fewer irrelevant images the better. On the other hand, we also want to preserve the automaticity of our framework, thus manual selection is not preferred here. We postulate two assumptions: (1) the set of retrieved images contains relevant images of the queried action and (2) humans or body parts should be seen in human action images.

It is reasonable to consider that in case of human actions, images which contain humans are more likely related images than images in which humans do not appear. Based on these assumptions, we select a collection of action images by applying a human detection method[32, 33] on Web images. For non-human actions, we simply select the first images returned by Web search engine and evaluate shot-image similarities by local feature matching (See Section 3.2.4.3). Note that in the first proposed mode of shot-image similarity calculation, we only care if images contain humans or not and compute similarities between human detected images and shots based on SURF matching. On the other hand, the second mode requires more detailed analysis of human movements and adopts human pose estimation method (See Section 3.2.4.2 and Section 3.2.4.4).

In the first mode, we use Poselets method[32] to detect humans. Poselets are demonstrated as effective body part detectors trained by 3D human annotations. We apply Poselets detector tools which are officially offered by the authors[2] on the set of retrieved Web images using default parameters. Figure 3.5 illustrates some examples of selected Web images using Poselets-based human detection.

[2] http://www.cs.berkeley.edu/%7Elbourdev/poselets/

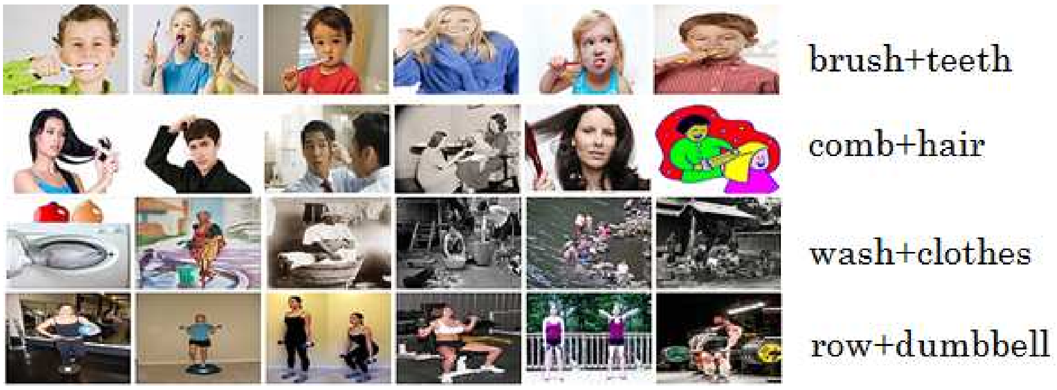

Figure 3.5: The top six Web images after Poselets-based image filtering.

Note that as shown in our previous work[8], the appropriate number of images to use in shot-similarity calculation step should be 20 to 30. Here we use 30 first human detected images.

3.2.4.2 Pose Feature

In case of human action recognition, not only low-level features such as SURF and our proposed spatio-temporal feature but high-level features like human pose should be also adopted. Even though actions may depend on actors or situations which they are taken, the basic poses for humans to perform them in general are similar. Based on this idea, we extract features of human poses detected in shots and images, and compare poses using these features. We suppose that the similarity calculation based on pose comparison can achieve better performance than local-feature-matching-based calculation.

As for the characteristics of a pose, we pay attention to relations of body parts' orientation or in other words, to their connection. We apply pose estimation models proposed by Y.Yang et.al[33] which are flexible mixture models for capturing contextual co-occurrence and spatial relations between body parts. For each pose, their full body model[3] detects 26 human body elements where 2 elements correspond to head, 4 elements relate to each limb and 8 elements point out torso (See Figure 3.6).

[3] http://phoenix.ics.uci.edu/software/pose/

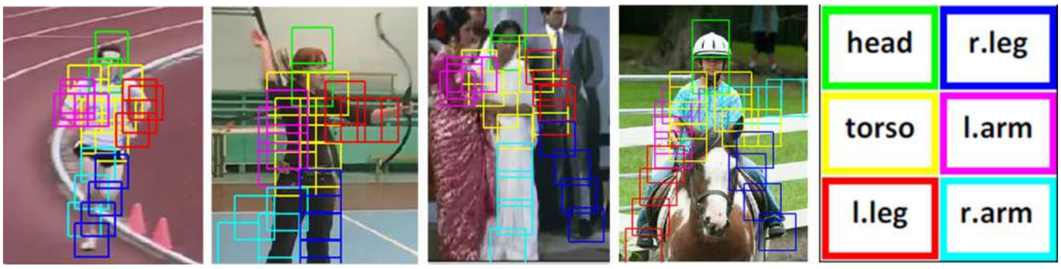

FIGURE 3.6: Examples of results by full body model

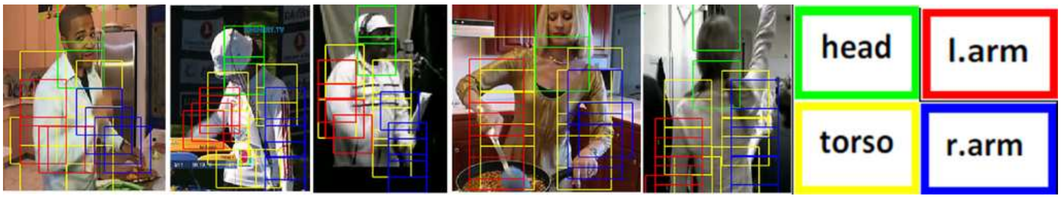

FIGURE 3.7: Examples of results by upper body model

Since our action category list contains some actions like "play piano" or "eat" which are most frequently taken when only upper bodies of actors appear, we also employ upper body model. In case of upper body pose estimation, the upper body model detects 2 elements of head, 4 elements of each of 2 arms, and 8 elements of torso (See Figure 3.7).

From each detected pose, we simply extract inner orientation and correlation orientation of its parts as its features. Inner orientation here is defined as direction of a body part such as an arm or a torso. Correlation orientation here refers to spatial relations between a pair of body parts such as a head and a leg. Following is how we calculate inner orientation and correlation orientation.

$$\begin{aligned} O_{in}(P) &= [dx_1 \;\; dy_1 \quad . \;\; . \;\; . \quad dx_{n-1} \;\; dy_{n-1}], \\ & \quad \text{where } \; dx_i = x_i - x_{i+1}, \; dy_i = y_i - y_{i+1} \qquad (3.10) \\ O_{co}(P_i, P_j) &= [X_{P_i} - X_{P_j} \;\; Y_{P_i} - Y_{P_j}] \qquad (3.11) \end{aligned}$$

where $O_{in}(P)$ means inner orientation of part P and $O_{co}(P_i, P_j)$ refers to correlation orientation between part P_i and part P_j. (x_i, y_i) represents position of element i of part P which has n elements. (X_P, Y_P) is defined as center position of part P. Finally, for each detected pose, we obtain a 70 dimensional feature. Note that for an image or a shot frame, first we apply the full body model. If the full body model fails to detect human pose, we then try the upper body model. If the upper body model succeeds to detect an upper pose, we calculate its orientation except for leg related orientation which will be regarded as 0. This enables us to compare poses even in case that they are detected by different body models.

3.2.4.3 Local Feature Matching Based Shot-to-Image Similarity Calculation

For shot-image similarity calculation, we first extract SURF local features[30] from all action images of selected set and each one frame per five consecutive frames of all the shots. For each shot, we count matching points between SURF local features extracted from each frame and each Web image by thresholding Euclidean distances between SURF feature vectors. The similarity $\mathrm{SI}(S_i)$ between a shot S_i which has M frame images ($F_j(j = 1..M)$) and an image set $\mathcal{I}$ which has N images ($I_k(k = 1..N)$) is calculated by the following equations:

$$\mathrm{SI}(S_i) = \sum_{k=1}^{N} \max_{j=1} \mathrm{SI}(F_j|I_k), \tag{3.12}$$

$$\text{where } \mathrm{SI}(F_j|I_k) = \frac{2 * \mathrm{MatchPoint}(F_j, I_k)}{(\mathrm{Point}(F_j) + \mathrm{Point}(I_k))}, \tag{3.13}$$

$\mathrm{MatchPoint}(F_j, I_k)$, $\mathrm{Point}(F_j)$ and $\mathrm{Point}(I_k)$ represent the number of matched points between a frame image F_j and a Web image I_k, the number of extracted SURF features from F_j and the number of extracted SURF features from I_k, respectively.

3.2.4.4 Pose Comparison Based Shot-to-Image Similarity Calculation

Like the above mode of shot-image similarity calculation, the similarity between a shot and a set of images is regarded as the similarity of its frame with the highest similarity score, and the similarity between a frame and a set of images is equal to normalized total similarity of that frame to all images in the set. Here we simply define pose comparison based similarity between a frame and an image as

Euclidean distance between the poses. However, in case of comparison between the upper body pose and the full body pose, we disregard leg associated elements. That means we only compare upper parts of the poses in this case. Moreover, since calculation of distance between two full poses will result in higher value than other cases due to extra leg related distance, we normalize it as follows:

$$\mathrm{SI}'(\mathrm{F}|\mathrm{I}) = \mathrm{SI}(\mathrm{F}|\mathrm{I}) * \frac{\text{number of elements unrelated to legs}}{\text{total number of elements}} \tag{3.14}$$

In this calculation of ours, the number of orientation elements unrelated to legs and total number of orientation elements equal to 40 and 70, respectively.

Chapter 4

Experiments and Results

To examine effectiveness of the proposed system, we conducted various experiments under different conditions with 100 kinds of human action keywords and 12 kinds of non-human action keywords. In Section 4.1, we explain our evaluation method and describe briefly about our experiments. Each experiment and its results will be expressed in detail in the next sections.

4.1 Experimental Setup and Evaluation Method

In our experiments, we used YouTube videos as our data source. We collected video metadata including video IDs and tags using YouTube Data API. To examine the effectiveness of our proposed method, we make large-scale experiments on 100 human action categories and 12 non-human action categories with video metadata analysis on 112,000 YouTube videos and spatio-temporal feature analysis on 22,400 YouTube videos. In each experiment, we obtained rankings of 2000 shots in average for each action, since as we mentioned above, we downloaded 200 videos for each action and each video is segmented into 10 shots in average. For the evaluation of recognition results, average precision is widely used. However, here we use the precision rate over top ranked 100 shots since we expect that they are qualified to be used for action database construction while commonly used datasets such as KTH dataset[34] and "in-the-wild" YouTube dataset[35] have approximately 100 video shots per action[1]. That means in each experiment, we simply count the number of relevant shots among 100 top ranked shots NR and the precision achieved in that experiment is computed as $NR/100$.

We carried out 5 experiments with various settings. We defined our experiments as follows:

(1) Exp.1: Original Framework (without both Web images and VisualTextualRank)

(2) Exp.2: Framework adopts Web images with local feature matching based shot-similarity calculation method

(3) Exp.3: Framework adopts Web images with pose comparison based shot-similarity calculation method

(4) Exp.4: Framework adopts VisualTextualRank

(5) Exp.5: Full Framework (with both Web images and VisualTextualRank)

The objective of Exp.1 is to verify the performance of the original framework which neither exploits Web images nor applies VisualTextualRank. On the other hand, Exp.2 and Exp.3 show the effectiveness of adopting Web images while Exp.4 and Exp.5 demonstrate the efficiency of our ranking method, VisualTextualRank. Especially, Exp.5 applies our proposed framework with full steps.

[1] KTH dataset has 599 shots for 6 actions, and "in-the-wild" dataset has 1168 shots for 11 actions.

4.2 Performance of the Original Framework

The purpose of the first experiment is to validate our original framework when Web action images are not taken into account and VisualTextualRank is not applied. We call this experiment as Exp.1. This means in Exp.1, shot selection step involves only spatio-temporal features and biases the top k shots regarding tag relevance scores (Equation 3.7). We conduct Exp.1 on our full action category set which consists of 100 human action categories and 12 non-human action categories. The results for human actions and non-human actions are summarized in Table 4.1 and Table 4.2, respectively.

As shown in Table 4.1, the mean of the precision at 100 shots over 100 human actions was 36.6%, and the precision varies from 2 to 100 depending on each action category. Top 34 actions regarding precision obtained 66 relevant shots among top ranked 100 shots in average and 14 actions achieved precision higher than 70%. Figure 4.1 shows some example results of some of successful action categories. However, the original framework failed to extract relevant shots for some actions (Figure 4.2). In the case of “boil egg”, some shots are actually related to “egg” but few of them describe exactly “boil egg” action. In cases of actions like “smile”, the action itself is too ambiguous to recognize. “Smile” is one of facial expressions which are mostly researched by emotion recognition works. Our proposed original framework can not distinguish “smile” and other facial actions. As for action keywords like “jog”, we could not select relevant videos of theirs due to tag noise as well as the variety in meaning of the keywords. Downloaded videos of “jog” mainly consist of videos about TV shows, movies or even motorbikes called as “jog”.

As for non-human actions, we obtained 14.9% as average precision. While some categories like “flower blooming” or “tornado” obtained quite a number of relevant shots at the top, some categories such as “leaves falling” and “waterfall” detected just very few relevant shots (Figure 4.3). In fact, for “leaves falling” or “waterfall” categories, most of collected videos are unrelated to the actions. The main reason is that tag noise led to the failure in relevant video selection.

Table 4.1: Precision@100 of 100 human actions (%)

soccer+dribble	100
fold+origami	96
crochet+hat	95
arrange+flower	94
paint+picture	88
boxing	86
jump+parachute	82
jump+trampoline	82
do+exercise	79
do+aerobics	78
do+yoga	77
surf+wave	75
shoot+arrow	73
massage+leg	72
fix+tire	67
batting	66
basketball+dribble	64
blow-dry+hair	64
knit+sweater	64
ride+bicycle	62
curl+bicep	58
shoot+ball	58
tie+shoelace	57
laugh	50
dive+sea	49
harvest+rice	49
ski	49
iron+clothes	47
twist+crunch	47
dance+flamenco	45
dance+hiphop	43
eat+ramen	42
dance+tango	41
play+trumpet	41
AVG. (1-34)	**65.9**

play+drum	40
skate	37
swim+crawl	36
cut+hair	35
run+marathon	35
count+money	33
paint+wall	33
shoot+football	33
draw+eyebrows	32
fieldhockey+dribble	32
hit+golfball	32
lunge	32
play+piano	32
row+boat	32
sing	32
chat+friend	31
clean+floor	31
cut+onion	31
shave+mustache	31
pick+lock	30
plaster+wall	30
blow+candle	29
wash+face	29
walking+street	29
brush+teeth	28
catch+fish	28
drive+car	28
plant+flower	28
play+guitar	28
lift+weight	27
raise+leg	27
hang+wallpaper	26
jump+rope	26
AVG. (35-67)	**31.0**

climb+tree	24
ride+horse	24
roll+makizushi	24
sew+button	24
fry+tempura	23
slap+face	20
read+book	19
squat	19
row+dumbell	16
wash+clothes	15
wash+dishes	15
comb+hair	14
drink+coffee	14
swim+breaststroke	13
cry	12
eat+sushi	12
serve+teniss	11
tying+tie	11
boil+egg	9
head+ball	9
swim+backstroke	9
take+medicine	8
serve+volleyball	7
swim+butterfly	7
bake+bread	6
cook+rice	6
grill+fish	5
jog	5
slice+apple	5
peel+apple	5
bowl+ball	4
smile	4
kiss	2
AVG. (68-100)	**12.2**
AVG. (ALL)	**36.6**

Table 4.2: Precision@100 of 12 non-human actions (%)

aircraft +landing	tornado	blooming +flower	airplane +flying	earthquake	shuttle +launching	
30	39	44	14	7	18	

leaves +falling	snow +falling	typhoon	heavy +rain	waterfall	explosion	**AVG.**
3	14	4	0	5	0	**14.9**

4.3 The Effectiveness of Exploiting Web Images

To examine the efficiency of introducing Web action images, we validate our modified system including the optional step on 28 human action categories and 8 non-human action categories which showed the lowest precision in the first experiment. Note that the local feature matching based framework (Exp.2) can run on both human actions and non-human actions while pose comparison based mode works (Exp.3) can only run on human actions. We show results of these experiments for human actions and non-human actions in Table 4.3 and Table 4.4 respectively. For human actions dataset, we want to evaluate the effectiveness of adopting Web action images and compare two modes of shot-similarity calculation.

Table 4.3: Results of 28 human action categories depending on how to exploit Web images. Exp.1: Web images unexploited, Exp.2: Web images + local feature matching exploited, Exp.3: Web images + Pose matching exploited

Action	Exp.1	Exp.2	Exp.3
slap+face	20	13	36
read+book	19	23	22
squat	19	32	37
row+dumbbell	16	24	33
wash+clothes	15	10	31
wash+dishes	15	25	40
comb+hair	14	12	20
drink+coffee	14	9	19
swim+breaststroke	13	31	11
cry	12	5	5
eat+sushi	12	11	15
serve+tennis	11	15	24
tie+necktie	11	23	24
boil+egg	9	6	14
head+ball	9	7	7
swim+backstroke	9	14	3
take+medicine	8	7	8
serve+volleyball	7	31	35
swim+butterfly	7	31	14
bake+bread	6	18	18
cook+rice	6	15	16
grill+fish	5	26	26
jog	5	21	10
pick+apple	5	9	2
slice+apple	5	2	13
bowl+ball	4	15	17
smile	4	18	26
kiss	2	3	3
Average	10.1	16.3	18.9

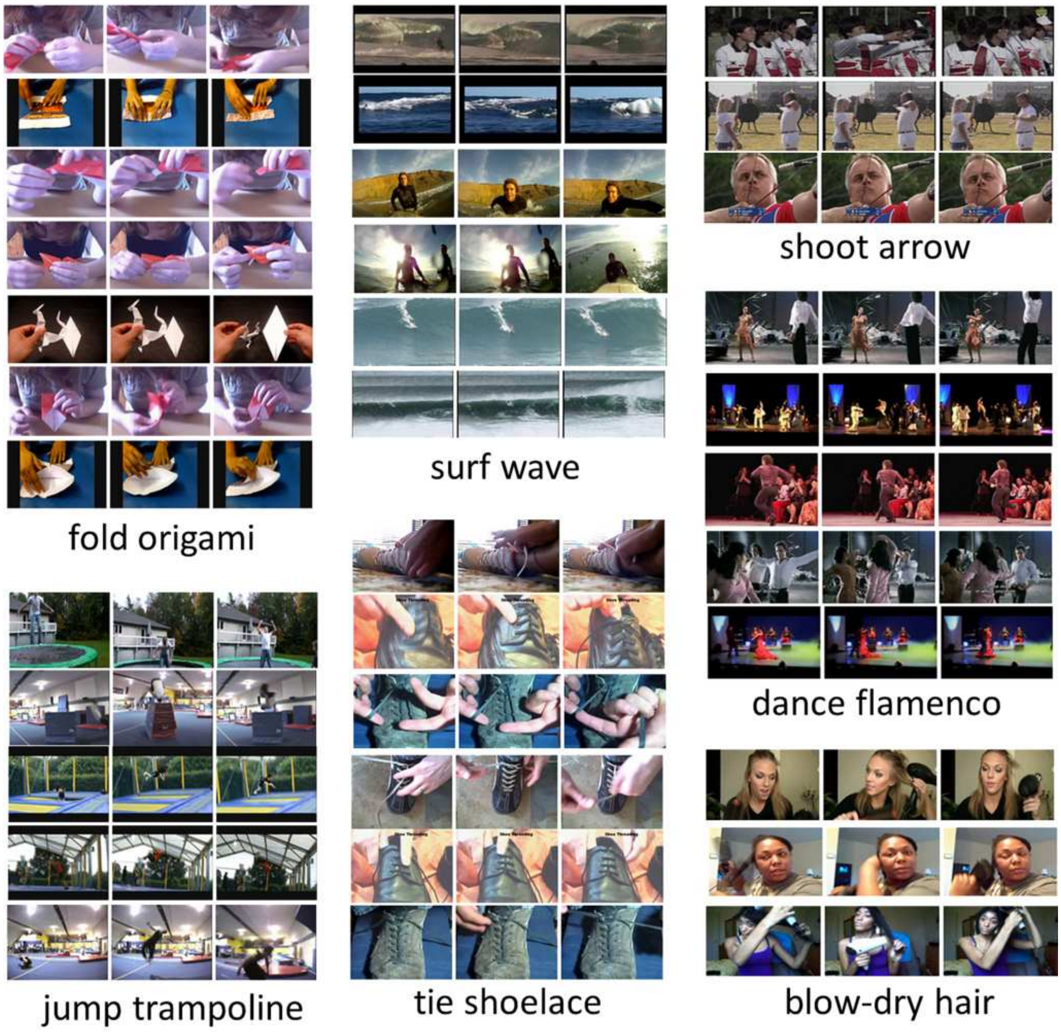

Figure 4.1: Relevant shots obtained in top 10 ranked shots of some categories which achieved high precision. Many of relevant shots are boosted to the top in these cases.

Table 4.4: Results of 8 non-human action categories of experiment on validating effectiveness of Web image introduction. Exp.1: Web image unexploited, Exp.2: Web image exploited (local feature matching based similarity calculation)

Action	Exp.1	Exp.2
explosion	0	5
falling+leaves	3	16
snow+falling	14	22
typhoon	4	29
airplane+flying	2	32
earthquake	7	25
heavy+rain	0	3
waterfall	5	17
Average	4.4	18.6

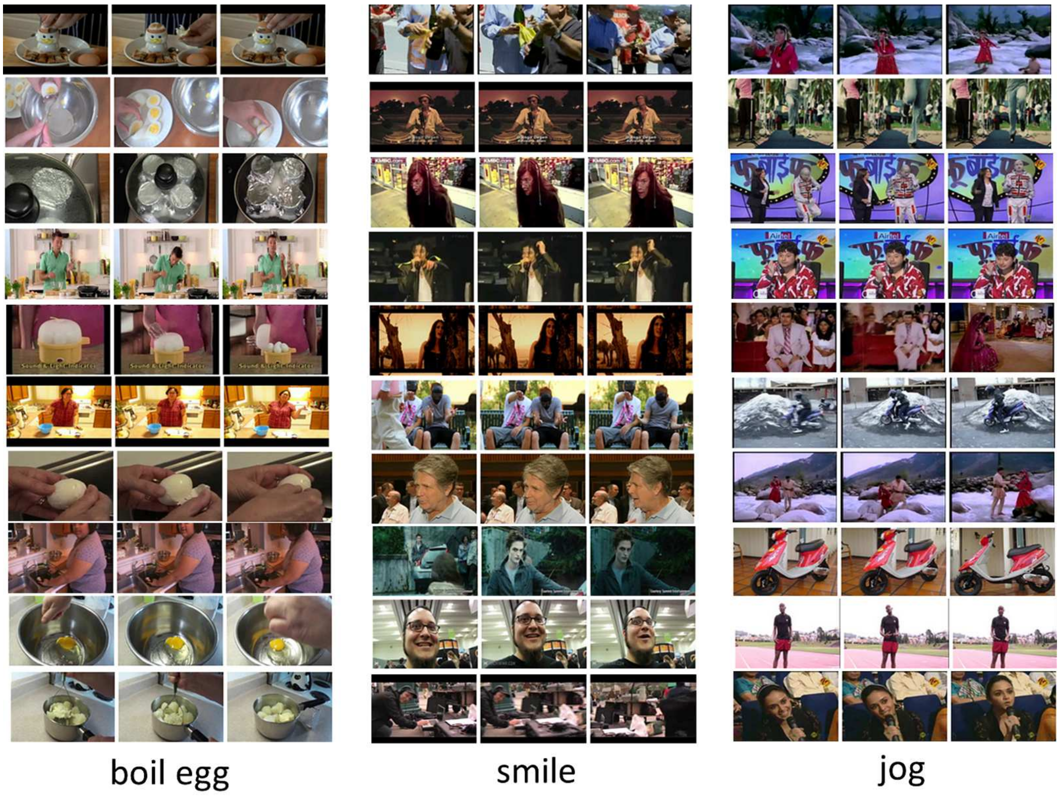

Figure 4.2: 10 shots among top 30 ranked shots of some low precision categories. As for "boil egg", "eggs" app ear in many shots but few shots describe exactly "boil egg" action. Especially, single action keywords such a s "smile" or "jog" are too ambiguous to obtain good candidate videos.

As shown in Table 4.3, introducing Web images helps to enhance the performance for human actions by 6.2% and 8.8% in average in case of exploiting local feature matching mode and pose matching mode respectively. For non-human actions, experimental results (Table 4.4) demonstrate that by introducing Web images into shot ranking, we can improve the precision from 4.4% to 18.6% in average. That means even in case where the tag noise led to the selection of irrelevant videos, our proposed method still can extract from those videos a number of action related video shots. Figure 4.4 and Figure 4.5 respectively shows some relevant shots which were detected by taking Web images into account in case of human actions and non-human actions.

We realized that local feature matching based method improved the performance in average but degraded it in cases of several categories such as "slap face", "wash clothes" and "comb hair". On the other hand, exploiting shot-to-image similarity

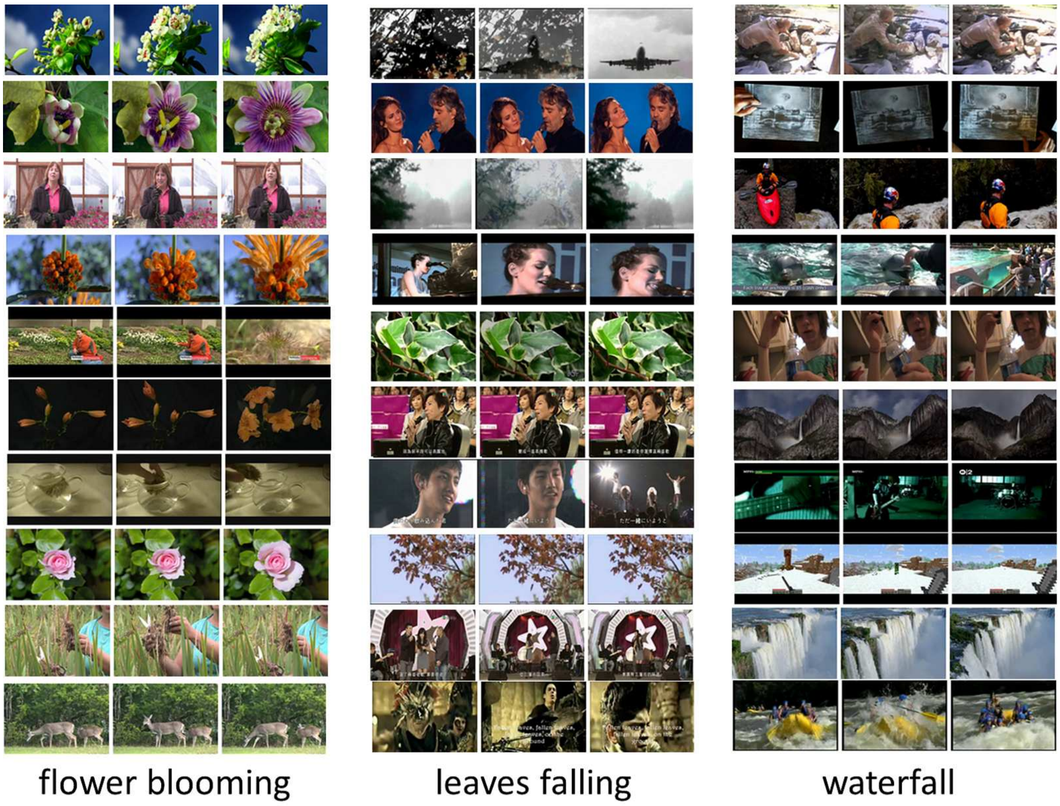

FIGURE 4.3: 10 shots among top ranked 50 shots. Nearly half of shots for "flower blooming" are expected shots. In the cases like "leaves falling" or "waterfall" tag noise caused selection of irrelevant videos. Particularly, "leaves falling" became tag of many music related clips so most of downloaded videos are not related to "leaves falling" scene but to music.

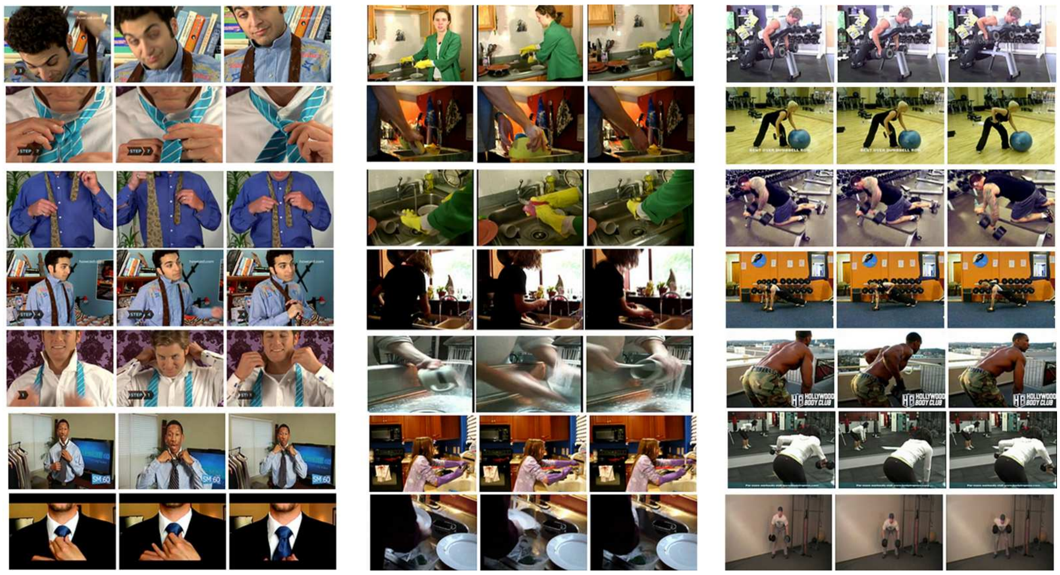

FIGURE 4.4: Some relevant shot that framework without optional step failed to detect were obtained by introducing Web images for human actions.

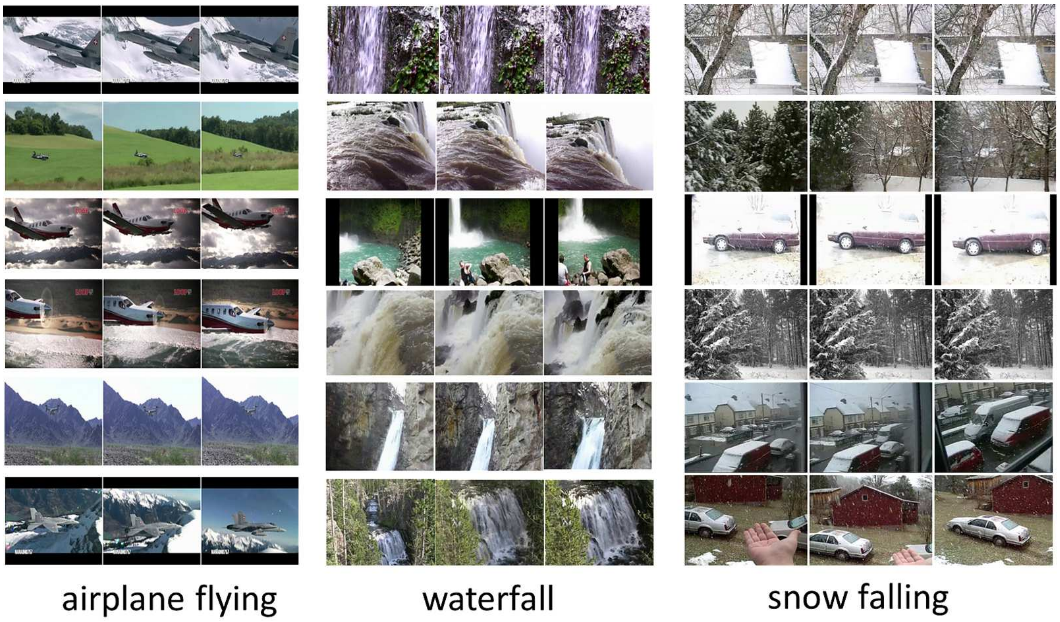

FIGURE 4.5: Some relevant shot that framework without optional step failed to detect were obtained by introducing Web images for non-human actions.

measurement based on pose comparison not only obtained the highest precision in average but also outperformed Web images unexploited framework for most actions except for “swim” related ones. In case of “swim”, human pose estimation failed to detect humans in water, hence obtained shots are mostly human detected shots such as medal rewarding, interviewing. (Figure 4.6). These results match with our expectation that in general, for human action learning, human poses hold very informative clues that should be exploited (Figure 4.7) and applying human pose matching to measure similarities between human action images can achieve better results than using low-level features only.

To confirm this hypothesis, we further conducted more experiments on other human action categories using pose matching between video shots and images introduced framework. We selected randomly 17 human action categories from actions which showed precision higher than 20% but lower than 35% in image unexploited framework. As expected, the performance was remarkably improved as it rose from 26.8% to 36.8% in average and the full system outperforms Web images unexploited system in most of categories. The results are summarized in Table 4.5 and result examples are shown in Figure 4.8.

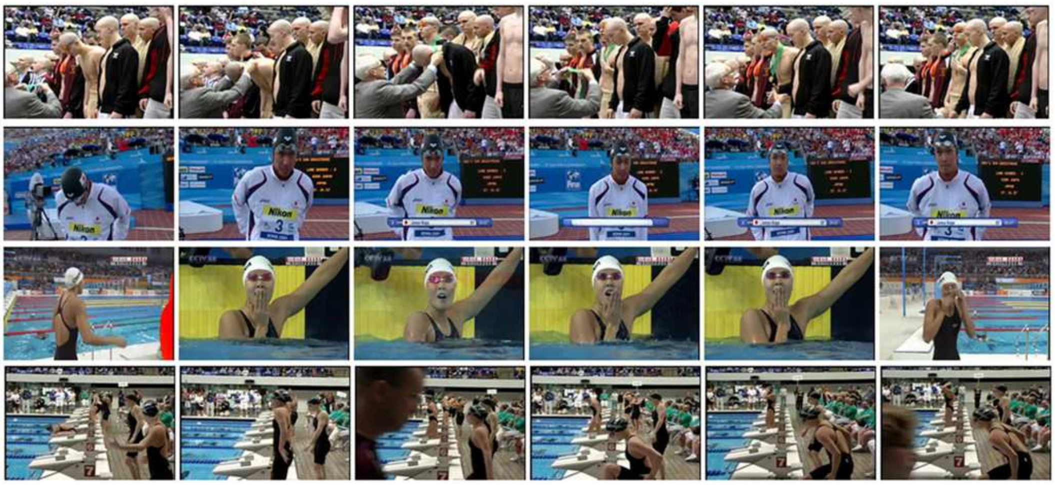

Figure 4.6: Top results of “swim backstroke”. Since humans could not be detected while swimming in the water so only human detected scenes like medal rewarding, interviewing, result notifying or warming-up (respectively from top to bottom) were obtained. This is one of few cases that human pose comparison based method does not work well.

Table 4.5: Results of 17 human action categories of experiment on validating effectiveness of proposed pose matching method

Action	Exp.1	Exp.3
blow+candle	29	35
clean+floor	31	38
jump+rope	26	39
roll+makizushi	24	26
sew+button	24	40
drive+car	28	35
ride+horse	24	35
catch+fish	28	45
play+guitar	28	38
shave+mustache	31	28
chat+friend	31	38
draw+eyebrows	32	35
play+piano	32	27
plaster+wall	30	38
brush+teeth	28	34
row+boat	32	28
wash+face	29	30
Average	28.6	36.8

4.4 The Efficiency of VisualTextualRank

In this subsection, we want to examine the efficiency of our proposed ranking method which refines simultaneously video shots and tags by employing not only co-occurrence relevance between tags and corresponding videos but also trustworthy content-based features extracted from videos. First, we conduct two experiments (Exp.4 and Exp.5) on 20 human action categories which above frameworks failed to detect relevant video shots. Exp.4 employs tags, spatio-temporal features and damping vector defined in Equation 3.7. That means Exp.4 is similar to Exp.1 except for the shot ranking method: Exp.1 applies VisualRank while Exp.4 uses our proposed method, VisualTextualRank. On the other hand, Exp.5 considers also human pose information during ranking processes and thus uses damping vector defined in Equation 3.8. Hence Exp.5 applies our full framework including optional step and VisualTextualRank. We chose randomly 20 categories among 45 failed categories by previous experiments and as the result of that choosing, the dataset of these two experiments consists of: 7 categories with precision between 20% and 30%, 10 categories with precision between 10% and 20%, and the remainder with precision below 10%.

Since until now, the precision obtained by experimental conditions in Exp.3 is the highest, we will compare the performance of Exp.4 and Exp.5 with that of Exp.3. In terms of overall performance on tested categories, as shown in Table 4.6, the framework with VTR outperforms previous ones and improves precision of most categories especially in cases of “blow+candle”, “jump+rope”, “catch+fish”, “play+guitar”, “wash+dishes”, “drive+car”, “slap+face”, “squat”, “serve+tennis”, “tie+necktie” which are enhanced significantly by more than 15%. Figure 4.9 provides some examples of relevant shots which were detected by VTR while previous frameworks failed to detect them.

About the effectiveness of introducing human pose feature, we can see that it depends on categories. Experimental results show that this kinds feature helps to improve some categories such as “serve+tennis” or “row+dumbbell” but degrades VTR in some categories such as “blow+candle”, “eat+sushi” and “drive+car”.

Table 4.6: Results of 20 human action categories compared between the framework with and without VTR. All of these categories have precision lower than 30% by Exp.3. Exp.4 computes VTR with a uniform bias vector (Equation 3.7) while Exp.5 uses pose features and damping vector Equation 3.8 to compute VTR.

Action	Exp.3	Exp.4	Exp.5
blow+candle	29	44	35
climb+tree	24	24	24
eat+sushi	12	23	15
jump+rope	26	49	47
catch+fish	28	59	54
read+book	19	21	20
boil+egg	9	11	14
grill+fish	5	13	19
play+guitar	28	41	43
wash+clothes	15	29	31
wash+dishes	15	39	39
drive+car	28	40	34
slap+face	20	45	44
squat	19	35	36
serve+tennis	11	27	30
cook+rice	6	11	15
comb+hair	14	26	27
roll+makizushi	24	36	32
row+dumbbell	16	30	33
tie+necktie	11	28	27
Average	17.9	31.5	30.9

Our explaination for these results is that in fact, pose features works better when human poses are taken in full body and without large occlusion. This case corresponds to “serve+tennis” and “row+dumbbell”. However, in cases like “blow+candle” or “eat+sushi”, in general, only upper bodies appear and they even are obscured by tables (See Figure 4.10). Thus we could not extract pose features properly and employing them in our method led performance of VTR down. This problem of pose features is also discussed in our previous paper[8].

Interestingly, we found that VTR also improve VisualRank in the sense that it increase the variety of ranking results. Since VisualRank employs only visual features, visually similar images are often ranked to the top. In case of shot ranking, applying VisualRank proposed sometimes boosts shots from the same videos to the top since they are generally look similar. On the other hand, VTR additionally exploits the correlation between videos and tags so that not only

Table 4.7: Additional experiment to demonstrate the effectiveness of VTR. The experimental settings here are used for Exp.4. Categories experimented here have precision higher than 30% by applying the framework in Exp.3.

Action	Exp.3	Exp.4
harvest+rice	49	46
play+trumpet	41	59
ski	49	60
dance+hiphop	43	68
play+drum	40	45
shave+mustache	31	30
dance+flamenco	45	53
clean+floor	31	38
pick+lock	30	28
swim+crawl	36	49
Average	39.5	47.6

visually similar video shots but also video shots having strong textual links with relevant shots also have chances to be ranked high as well (See Figure 4.11).

To furthur quantify the performance of proposed VTR, we apply our framework with VTR to other 10 actions randomly chosen from categories whose precision higher than 30% by the framework in Exp.3. It is clear that VTR reduces the problems caused by tag noise but how about categories which the previous frameworks quite successfully obtained relevant shots? We expect that VTR can improve our system as well since it was demonstrated as effective method exploiting both tags and content based features. As VTR using uniform damping vector is conducted more easily but achieved better results than VTR using pose feature based bias vector in terms of overall performance according to results of the previous experiments, we use the former in this experiment. Its results are presented in Table 4.7.

The results in Table 4.7 show that VTR enhances our system on all tested categories excepting for "harvest+rice". Precision of "play+trumpet", "ski", "dance+hiphop", "swim+crawl" is boosted significantly. The average precision is improved by approximately 8%. This means in case of "good" categories, VTR also helps to detect more relevant shots. Some examples of successfully detected relevant shots by VTR are presented in Figure 4.12.

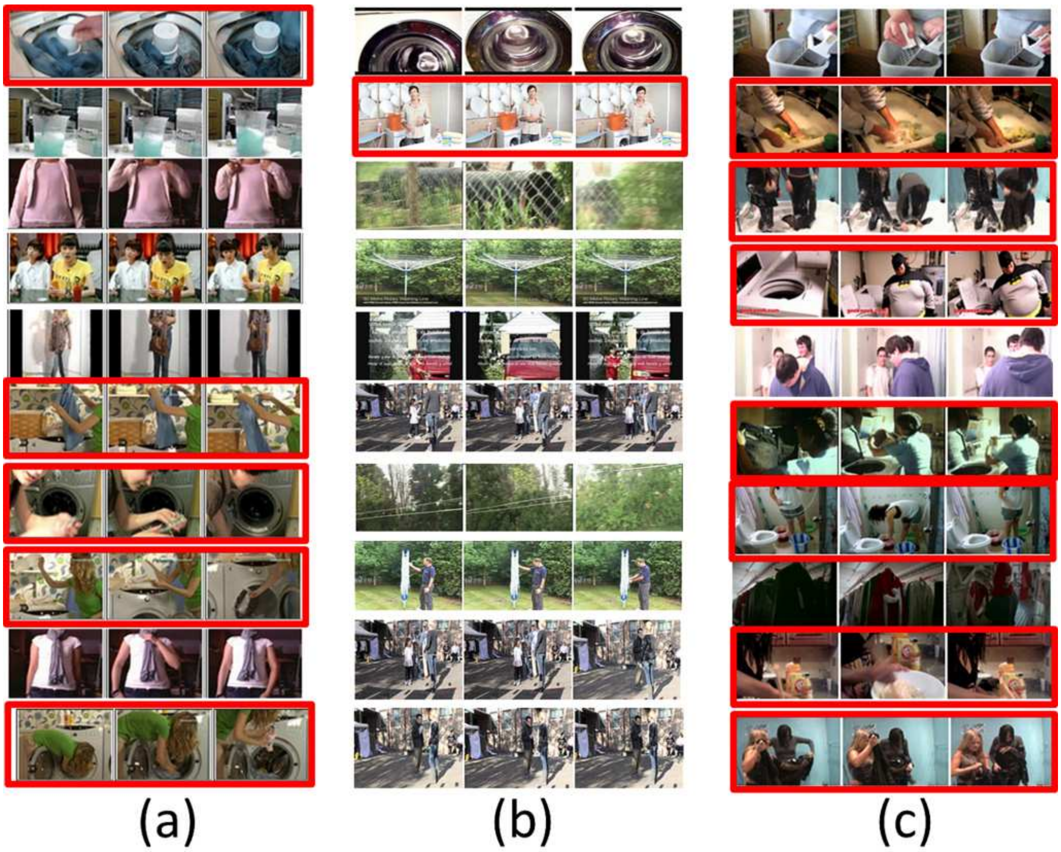

Figure 4.7: Top 10 ranked results for "wash clothes" by (a) Web images unexploited framework, (b) local feature matching exploited framework, (c) Pose matching exploited framework. Relevant shots are bounded with red boxes. As shown here, while local feature matching based method ranked less relevant shots to the top, pose comparison based framework biased to the shots which have human poses of washing clothes so it performed better.

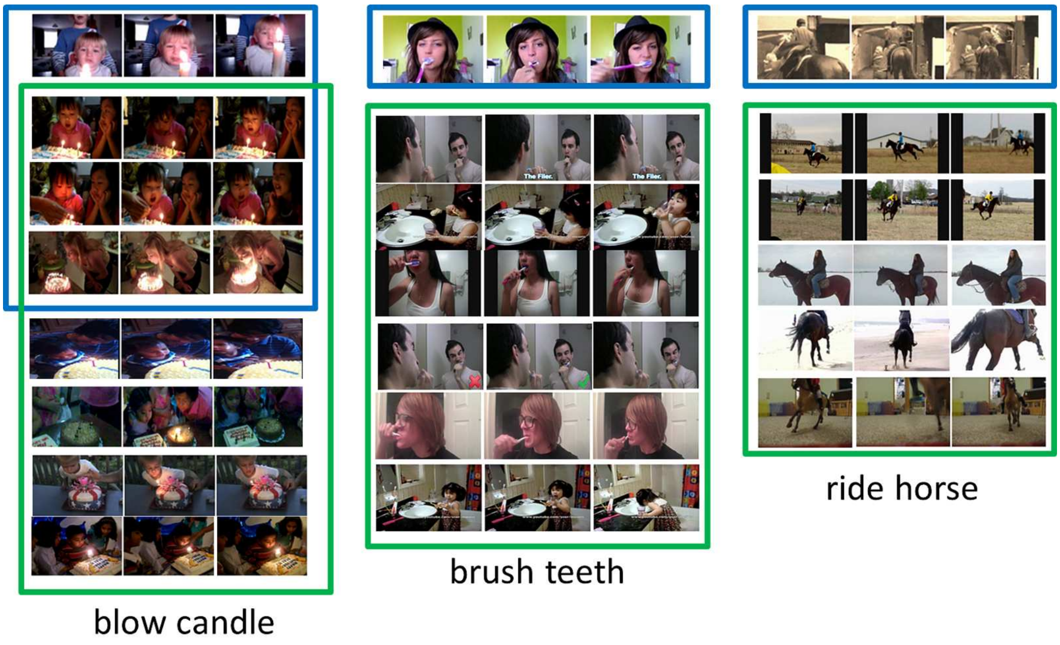

FIGURE 4.8: Relevant shots among top 15 ranked shots for "blow candle", "brush teeth" and "ride horse". Relevant shots which were extracted by Web images unexploited framework and Web image exploited framework with pose matching based shot-image similarity calculation are enclosed by blue and green bounding box respectively. These results demonstrate that exploiting Web images helps to boost more relevant shots to the top.

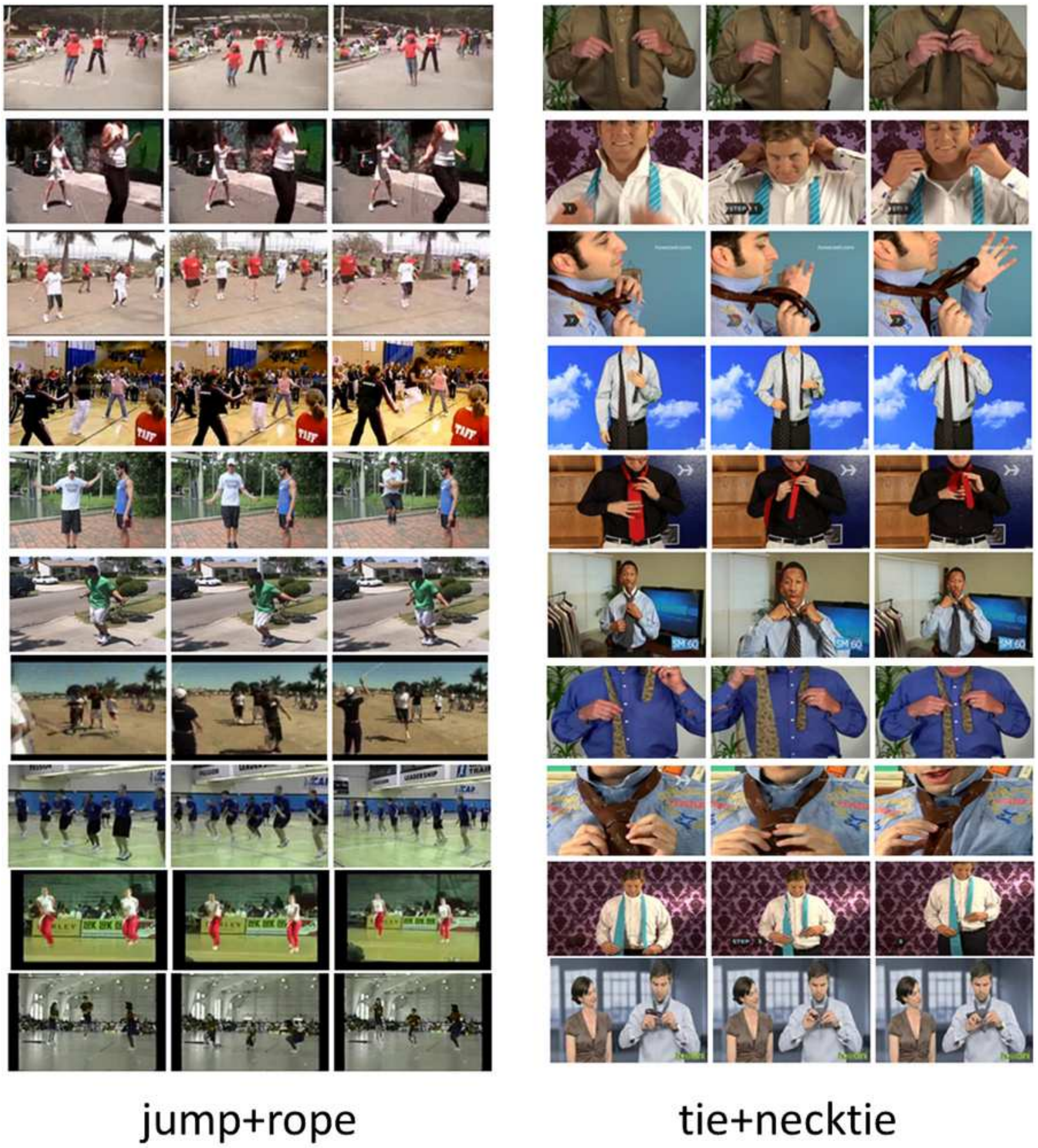

FIGURE 4.9: 10 shots among relevant shots that framework with VTR successfully extracted but framework without VTR did not. Categories here have precision lower than 30% by the previous framework.

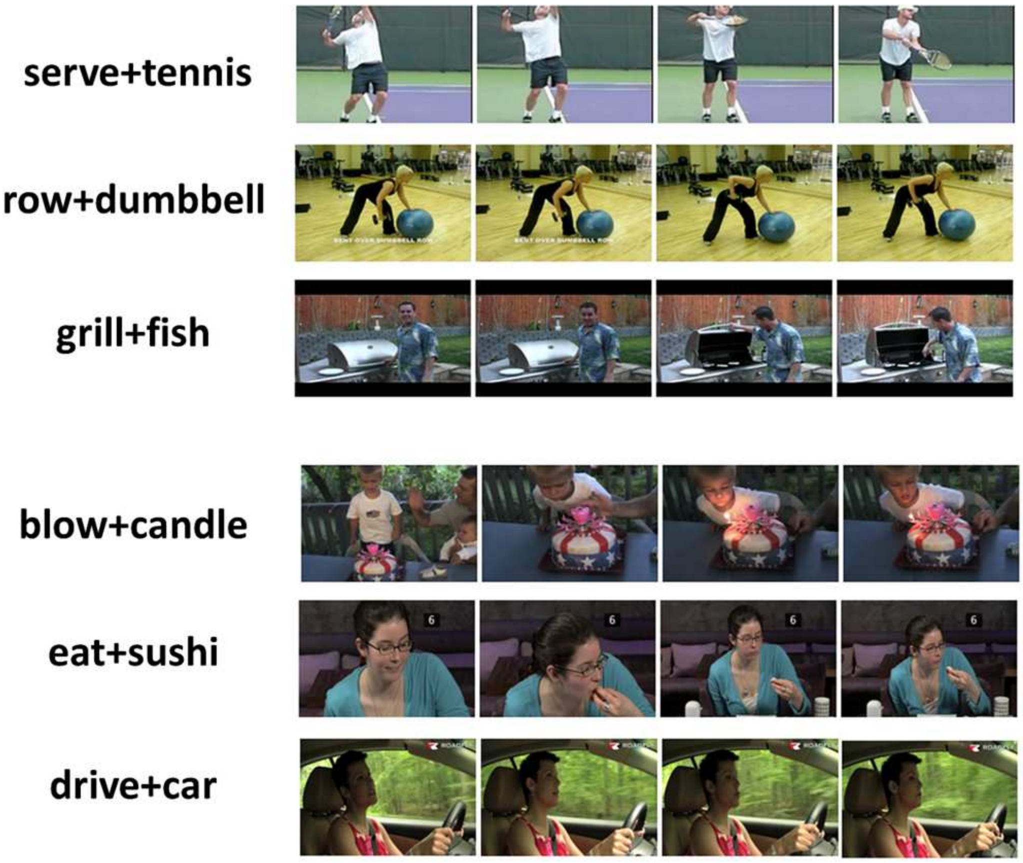

Figure 4.10: Effectiveness of introducing pose feature. Top three categories: “serve+tennis” and “row+dumbbell” (whole body seen), “grill+fish” (upper body clearly seen) are examples of ones which Pose-VTR obtained better results. Bottom three categories: “blow+candle”, “eat+sushi”, “drive+car” (upper body occluded by many objects) are cases when pose can not be detected and hence pose exploited VTR performed worse than VTR with uniform bias vector.

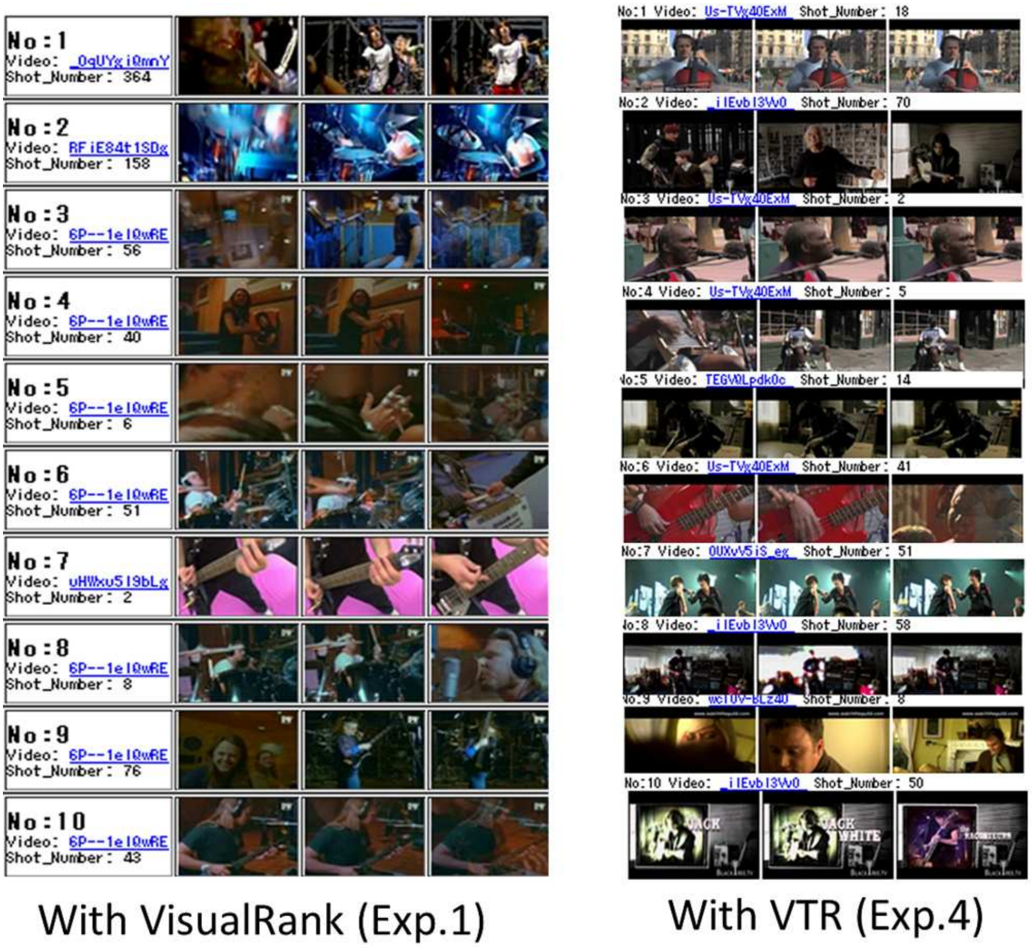

FIGURE 4.11: Diversity of results obtained by our system with(right) and without(left) VTR. The category here is "play+guitar". As in the original framework (Exp.1), more than half of top 10 shots are from the same video with ID "6P–1elQwRE". Besides, VTR can select relevant shots from different videos since it effectively employs both textual and visual features of them.

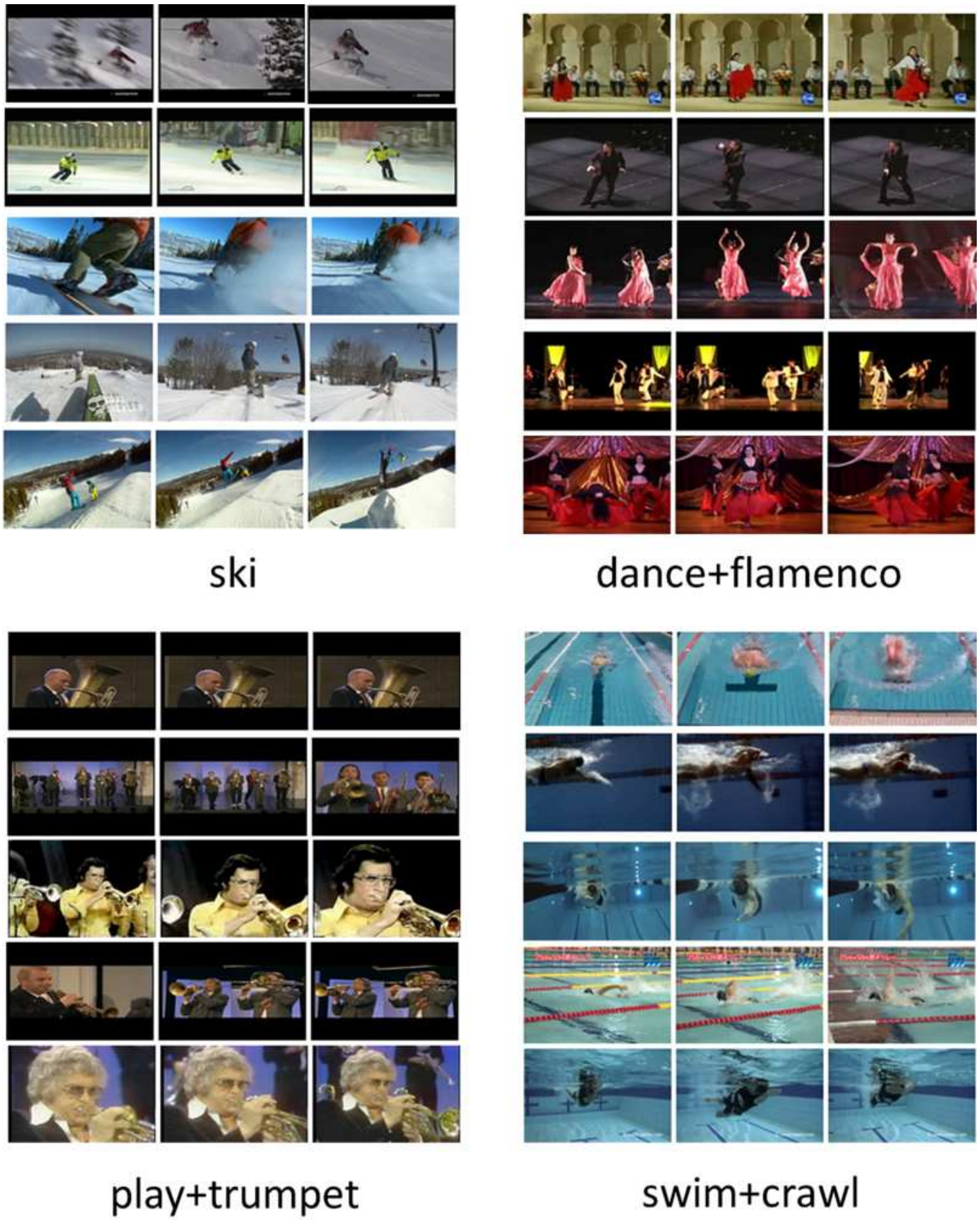

Figure 4.12: 5 shots among relevant shots that the framework exploiting VTR successfully extracted but the original framework did not. Categories here have precision lower than 30% by Exp.3.

Chapter 5

Conclusions and Discussion

In this paper, we proposed a method of automatically extracting from Web videos video shots corresponding to specific actions by only providing action keywords. To the best of our knowledge, we are the first to aim at automatic construction of such a large-scale database for action recognition. The empirical results show that the performance of proposed framework depends on the action categories and selection of action keywords. For some actions the original method worked very well. For example, precision rates of the best 24 and 35 actions exceed 50% and 40%, respectively, by the original framework which neither exploits Web images nor applies our proposed ranking method.

However, in the cases of many categories, exploiting action images helps significantly enhance performance of shot ranking step. Particularly, exploiting proposed shot-image pose matching method improved precision rates of most of experimented human categories.

Especially, the effectiveness of proposed novel ranking method, VisualTextualRank, which performs co-ranking of video shots and tags employing both visual links between video shots along with textual links between videos and their tags, can be seen in many categories.

As future works, first, we plan to improve video selection step and adopt more context features such as human-object interactions or scene information to our framework. Our framework can help reduce tremendous human effort on building database for action recognition. Although a few modest manual scanning may still be needed to use these video shots as training data, there is no doubt that human effort can be significantly reduced in comparison to fully manual database construction. What is the benefit of constructing huge action database without difficulty? So far there has been no visual analysis of verbs using Web videos. The reason is that we need an immense database to conduct that analysis while building such database has been considered as an exhausted work. By applying our system, it is possible to build large-scale database with much effort. Our future work is to quantify the relationship between the concepts of verbs and the features of their corresponding video shots or more precisely, the characteristics of the actions performed in these shots. Understanding this relationship can help us categorize any verb based on the actions related to it. For instance, "type" as in "type keyboard" and "play piano" are totally unrelated according to their definition but in fact, they are "visually similar" - the action to perform them look similar. (see Figure 5.1). Moreover, we can also classify objects based on the way human interact them as the results of visual analysis between verb phrases which are comprised of their nouns and a specific verb. For example, we have "udon", "ramen", "onigiri", "hamburger" as the nouns which we want to categorize and we have "eat" as our verb. By analyzing the visual relationship between "eat udon", "eat ramen", "eat onigiri" and "eat hamburger", we can classify "udon" along with "ramen" to a group of food, and "onigiri" along with "hamburger" to another group (see Figure 5.2).

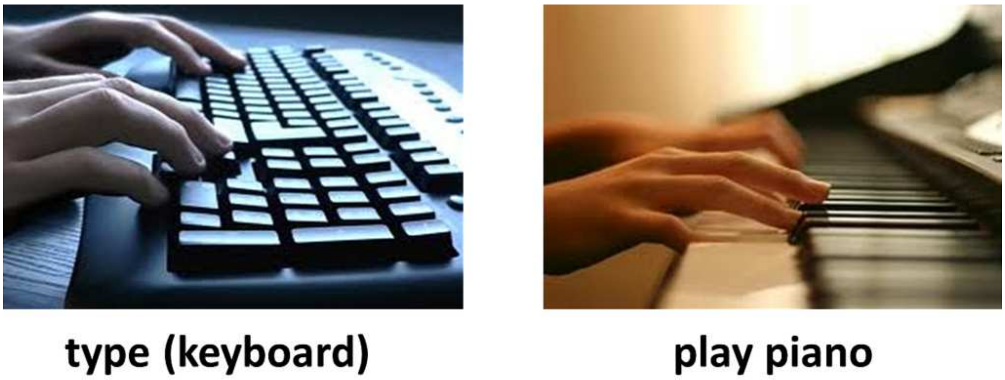

Figure 5.1: Images of "type" (left) and "play piano" (right). Even though they are textually unrelated, they should be categorized into the same action group since the action to perform them look really similar. While typing or playing piano, people usually use their fingers as shown in this figure.

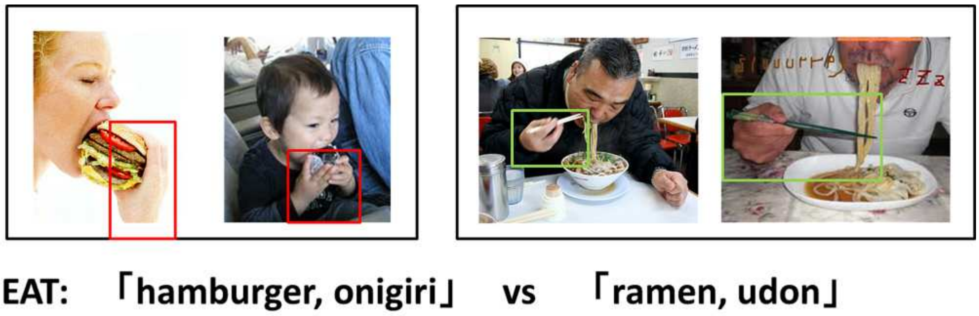

Figure 5.2: Images of people eating "hamburger", onigiri, "ramen", "udon" (respectively from left to right). When people eat "udon" or "ramen", they usually use chopsticks while for "hamburger" and "onigiri", they commonly use their own hands. Thus we can classify "udon" along with "ramen" to a group of food, and "onigiri" along with "hamburger" to another group.

Bibliography

[1] H. A. Jhuang, H. A. Garrote, E. A. Poggio, T. A. Serre, and T. . Hmdb: A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2011.

[2] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 (11):1870–1890, 2008.

[3] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[4] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010.

[5] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[6] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[7] D. H. Nga and K. Yanai. Automatic construction of an action video shot database using web videos. In *Proc. of IEEE International Conference on Computer Vision*, 2011.

[8] D. H. Nga and K. Yanai. Automatic collection of web video shots corresponding to specific actions using web images. In *CVPR Workshop on Large-Scale Video Search and Mining*, 2012.

[9] L. Liu, L. Sun, Y. Rui, Y. Shi, and S. Yang. Web video topic discovery and tracking via bipartite graph reinforcement model. In *Proc. of the ACM International World Wide Web Conference*, pages 1009–1018, 2008.

[10] S. Jones, L. Shao, J. Zhang, and Y. Liu. Relevance feedback for real world human action retrieval. *Pattern Recognition Letters*, 2012.

[11] X. Zhen, L. Shao, D. Tao, and X. Li. Embedding motion and structure features for human action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012.

[12] L. Liu, L. Shao, and P. Rockett. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, 2012.

[13] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. of British Machine Vision Conference*, 2006.

[14] J. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting moving people from internet videos. In *Proc. of European Conference on Computer Vision*, pages 527–540, 2008.

[15] N. I. Cinbins, R. G. Cinbins, and S. Sclaroff. Learning actions from the web. In *Proc. of IEEE International Conference on Computer Vision*, pages 995–1002, 2009.

[16] L. Ballan, M. Bertini, A. D. Bimbo, M. Meoni, and G. Serra. Tag suggestion and localization in user-generated videos based on social knowledge. In *Proc. ACM MM WS on Social Media*, pages 3–7, 2010.

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[18] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. of IEEE International Conference on Computer Vision*, 2009.

[19] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1491–1498, 2009.

[20] K. Yanai. Generic image classification using visual knowledge on the web. In *Proc. of ACM International Conference Multimedia*, pages 67–76, 2003.

[21] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. of European Conference on Computer Vision*, pages 242–255, 2004.

[22] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. of IEEE International Conference on Computer Vision*, pages 1816–1823, 2005.

[23] K. Yanai and K. Barnard. Probabilistic Web image gathering. In *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 57–64, 2005.

[24] L. Li and L. Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2007.

[25] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proc. of IEEE International Conference on Computer Vision*, 2007.

[26] Q. Yang, X. Chen, and G. Wang. Web 2.0 dictionary. In *Proc. of ACM International Conference on Image and Video Retrieval*, pages 591–600, 2008.

[27] D. Liu, X.S. Hua, L. Yang, M. Wang, and H.J. Zhang. Tag ranking. In *Proc. of the ACM International World Wide Web Conference*, pages 351–360, 2009.

[28] K. Yanai. Automatic Web image selection with a probabilistic latent topic model. In *Proc. of the ACM International World Wide Web Conference*, 2008.

[29] A. Noguchi and K. Yanai. A surf-based spatio-temporal feature for feature-fusion-based action recognition. In *ECCV WS on Human Motion: Understanding, Modeling, Capture and Animation*, 2010.

[30] B. Herbert, E. Andreas, T. Tinne, and G. Luc. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, pages 346–359, 2008.

[31] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[32] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. of IEEE International Conference on Computer Vision*, 2009.

[33] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2011.

[34] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proc. of International Conference on Pattern Recognition*, pages 32–36, 2004.

[35] J. Liu, J. Luo, and M. Shah. Recognizing realistic action from videos. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2009.