UEC at TRECVID 2005 High Level Feature Task

— Web Images Meet TRECVID —

Keiji Yanai, Liu Ounan and Yuki Tsujita

Department of Computer Science, The University of Electro-Communications, JAPAN
yanai@cs.uec.ac.jp

ABSTRACT

We participated in the TRECVID high level feature extraction task for the first time. Our initial policy for TRECVID is using visual knowledge collected from the World-Wide Web instead of using common annotation data. Since image data on the Web are weakly annotated by HTML files, there is possibility that we can use them as training data without any human annotation labor. Currently, we are pursuing this new concept. The aim that we participated in TRECVID is investigating if high level concepts in TV news movies can be detected with visual knowledge on the Web.

In this paper, we describe about our first attempt for TRECVID. We applied the GMM-based generative model we originally designed for Web image gathering task to the high level feature extraction task with no modification. We trained the models with three kinds of training data, which are Web images, common annotation data and their combination. We made three runs with Web images in three different setting in terms of the number of GMM components, one run with common annotation data, and one with their combination. For all runs we used the same method.

1. INTRODUCTION

Our main research interest is generic object recognition. Recently we are working on gathering images from World Wide Web and applying them to generic object recognition tasks as visual knowledge [6]. Learning of image concepts from the Web is paid attention as the new framework to avoid human labor for making training image sets [4, 5]. We regard the TRECVID tasks as a kind of generic object recognition tasks in our first participation this year. Therefore, what we are interested in among TRECVID tasks is the high level feature extraction task. To treat with the task as image recognition, we used only keyframe images provided by the TRECVID committee ignoring ASR text data and motion information which can be extracted from movie files.

This year we applied the GMM-based probabilistic model which is designed for an Web image gathering task [7] to the TRECVID 2005 high level feature extraction task without any modification. We submitted only the "waterscape/waterfront" and "mountain" tasks out of the 10 high level feature extraction tasks this year. In the experiment for [7] presented at ACM MIR 2005, we have already gathered images of "mountain" and "beach". We applied the models trained during the probabilistic Web image gathering for them to the TRECVID task. Unfortunately, the results we submitted were not as good as we expected. This comes from the difference between the nature of images extracted from TV news and the nature of images gathered from the Web greatly, and indicates that our model for Web images is needed to be modified for TV news images at TRECVID.

In addition, we would like to point out that TRECVID 2005 test collection has a problem that both develop collections and test collections include a lot of identical shots most of which are commercial shots.

2. METHOD

We applied the probabilistic image classification method employed in our probabilistic Web image gathering system [7] with no special modification for the TRECVID task.

In our Web image gathering scheme, we gather several hundreds of Web images relevant to a given concept. At first, we provide keywords which represent the visual concept of images we like to obtain. For example, "mountain", "beach" and "sunset". Using Web image/text search engines, we gathered "raw" images related to the given concept from the World Wide Web. The "raw" image always includes many irrelevant images, the ratio of which is 50% or more on average.

Next, we employ a probabilistic method to select relevant images from all the raw images. In general, to use a probabilistic method or other machine learning methods to select true images, we need labeled training images. However, we do not want to pick up relevant images by hand. Instead, we regard images which are highly evaluated by the HTML text analysis as training images, although they include some irrelevant images. In our probabilistic framework, we allow training data to include some irrelevant data and we can remove them by repeating both estimation of a model and selection of relevant regions of images from all the regions of raw images. We use a generative model based on the Gaussian mixture model to represent models associated to keywords, and estimate models with the EM algorithm. After estimating the model, we "recognize" relevant region out of all regions in all the raw images with the model. We repeat this model estimation and region selection. After the second iteration, we use regions selected in the previous iteration as training data for estimating a model.

2.1. Segmentation and Image Feature Extraction

To extract image features from each region, we carry out the region segmentation in advance. In the experiments, we used JSEG [2]. After segmentation, we extract image features from each region whose size is larger than a certain threshold. As image features, we prepare three kinds of features: color, texture and shape features, which include the average RGB value and its variance, the average response to the difference of 4 different combination of 2 Gaussian filters, region size, location, the first moment and the area divided by the square of the outer boundary length. An image feature vector we use in this paper is totally 24-dimension. We need to do such pre-processing for all the TRECVID keyframe images of test data as well as all of the Web raw images. It took about 7 days for all of the keyframe images of TRECVID test data with 10 PCs.

2.2. Overview of Probabilistic Approach

As a method to select images, we adopt a probabilistic method with a Gaussian mixture model. This approach is based on the method for learning to label image regions from images with associated text without the correspondence between words and images regions [3, 1]. That method uses a mixture of multi-modal components, each combining a multinomial for words and a Gaussian over image features. Here, we simplify things a bit, and build models of the distribution of image features for a given concept for regions which are obtained by a region segmentation algorithm.

To get a model of regions associated to a certain concept, we need training images. As mentioned before, our basic policy is no human intervention, so that we use images which are highly evaluated by the HTML analysis as training images. Most of such images are relevant ones, but they always includes outliers due to no supervision. Moreover, in general, images usually include backgrounds as well as objects associated with the given concept. Therefore, we need to eliminate outlier images and regions unrelated to the concept such as backgrounds, and pick up only the regions strongly associated with the concept in order to make a model correctly. We use only the regions expected to be highly related to the concept to estimate a model. In our new method, we need negative training images in addition to "raw" images. We prepare about one thousand

images by fetching them from the Web randomly as negative training images in advance.

Our method to find regions related to a certain concept is an iterative algorithm similar to the expectation maximization (EM) algorithm applied to missing value problems. Initially, we do not know which region is associated with a concept "X", since an image with an "X" label just means the image contain "X" regions. In fact, with the images gathered from the Web, even an image with an "X" label sometimes contains no "X" regions at all. So at first we have to find regions which are likely associated with "X". To find "X" regions, we also need a model for "X" regions. Here we adopt a probabilistic generative model, namely a mixture of Gaussian, fitted using the EM algorithm.

In short, we need to know a model for "X" regions and which regions are associated with "X" simultaneously. However, each one depends on each other, so we proceed iteratively. Once we know which regions corresponds to "X", we can regard images containing "X" regions as "X" images, and therefore we can compute the probability of an "X" image for each image. Finally, we select the images which have the high probability as final results.

2.3. Applying Trained Models to TRECVID data

To detect images relevant to the concepts given in TRECVID high level feature extraction, we apply the trained models in the same way as region selection during the training stage, and estimate the probability of "X" ($P(X|R)$) for all the regions extracted from all the keyframes. Finally, we regard the mean of the probability of "X" of top T regions within each image as the probability of "X" ($P(X|I)$) for each image. In the experiment, we set T as 2. We sort the keyframes in the order of $P(X|I)$, and pick up top 2000 keyframe images as a result.

In the experiment, we used only "mountain" and "beach" model. Note that we used the "beach" model to detect"waterfront/waterscape" shots.

2.4. Algorithm

To summarize our method we described above, the algorithm is as follows:

(1) Carry out region segmentation for all the images and extract image features from each region of each image.
(2) At the first iteration, regard images in group A as positive training images which are associated with the concept "X" and images gathered from the Web with non-noun keywords in advance as negative training images. Note that "images in group A" are images highly evaluated by the HTML analysis.
(3) Select n "X" regions randomly from positive images, and select n "non-X" regions randomly from negative images, respectively (Figure 1 (4)).
(4) Applying the EM algorithm to the image features of regions which are selected as both positive and negative regions, compute the Gaussian mixture model for the distribution of both "X" and "non-X" (Figure 1 (5)).
(5) Find the components of the Gaussian mixture which contributes "X" regions or "non-X" regions greatly. They are regarded as "X" components or "non-X" components, and the rest are ignored. The mixture of only "X" regions is a model of "X" regions, and the mixture of only "non-X" is a model of "non-X" regions.
(6) Based on the mixture of "X" components and the mixture of "non-X" components, compute $P(X|r_i)$ and $P(nonX|r_i)$ for all the regions which come from "X" images, where r_i is the i-th region.
(7) Select the top n regions in terms of $P(X|r_i)$ as new positive regions and the top $\frac{1}{3}n$ regions in terms of $P(nonX|r_i)$ as new negative regions. Add $\frac{1}{3}n$ regions randomly selected from the negative training images to new negative regions.
(8) Repeat from (4) to (7) with newly selected positive and negative regions (Figure 1 (7)).
(9) After repeating several times, apply the trained model to the TRECVID test data set. Calculate $P(X|r)$ for all the regions extracted from

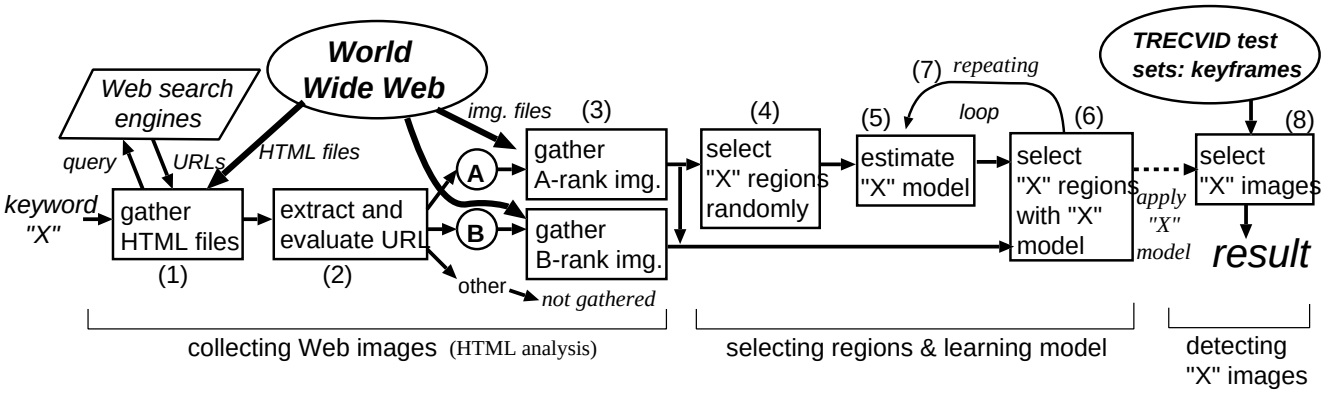

Figure 1: Processing flow of Web image gathering and detecting "X" images from the TRECVID test data sets.

all the keyframe images of the TRECVID test set, and then obtain $P(X|I)$. Finally select top 2000 images in terms of $P(X|I)$ as a output result (Figure 1 (8)).

The detail of the algorithm is described in [7].

3. EXPERIMENTAL RESULTS

Table 1 shows our all results. We made five runs in the different setting. For three out of five runs, we used Web images as training data. They are represented in the table as "Web1", "Web2" and "Web3", respectively. The difference among them is the number of Gaussian components of the GMM models. We set 200, 250 and 300 as the number of Gaussian for "Web1", "Web2" and "Web3", respectively.

In Table 1, "Web+Com" represents the result when both Web images and common annotation data are used as training data. "Common" means the result in case that only common annotation data are used as training data with the same method as the method for Web images. In addition, the table includes the result of an additional run, "Hist", which we made after the submission deadline. "Hist" is the result by color histogram and k-nearest neighbor in case of using common annotation sets as training data.

All the results are not as good as we expected. This comes from the difference between the nature of images extracted from TV news and the nature of images gathered from the Web, and indicates that our model for Web images is needed to be modified for TV News at TRECVID.

In our model, we need negative training images in addition to "raw" images gathered from the Web. In all the experiments, we used images randomly gathered from the Web as a negative training data set. Negative images are important to boost the difference between relevant images and irrelevant images. Therefore, in case of TRECVID, we should have used randomly selected keyframes from the development data as negative data in addition to images from the Web.

4. A PROBLEM ON TRECVID HIGH LEVEL FEATURE TASKS

As we pointed out in the previous section, for the "mountain" and "waterscape" task the very simple color histogram was very effective. Then, we examined if the results by color histogram include the identical shots of both develop and test data keyframe image set.

We examined top 200 keyframe images of the color histogram results for "mountain" and "waterscape". They includes 165 and 200 relevant images within the top 200 images, which means top 200 precision are 0.825 and 1.000, respectively. They are extremely high precision as for the very simple method. We found that all of these im-

Table 1: The mean average precision and the number of detected relevant shots within top 2000 shots for our 5 runs and an additional run by color histogram and k-nearest neighbor.

concepts	min	median	max
runs	MAP	# detected shots	
43. Waterscape	0.001	0.187	0.493
Web1	0.0068	102/868	
Web2	0.0142	143/868	
Web3	0.0176	161/868	
Web+Com	0.0140	148/868	
Common	0.0146	141/868	
Hist	0.4315	441/868	
44. Mountain	0.001	0.155	0.458
Web1	0.0254	135/752	
Web2	0.0237	128/752	
Web3	0.0228	123/752	
Web+Com	0.0172	108/752	
Common	0.0187	120/752	
Hist	0.2481	252/752	

ages had identical keyframe images in the training data. That is, all of the relevant keyframe images within the top 200 of "mountain" and "waterscape" have the exactly same keyframes in the common annotation training sets. Figure 2 shows some of identical "mountain" image pairs. The bigger images are images from a test set, and the small images below the bigger images are identical images found in the common annotation sets of "mountain". Most of them seem to come from commercial shots, although we did not examine all of the identical shots.

Table 2 shows the results for 10 concepts by color histogram and k-nearest neighbor which are evaluated with the official evaluation program provided by the TRECVID committee. This shows that the identical shots affect the results of "building exterior", "waterscape/waterfront" and "mountain" greatly.

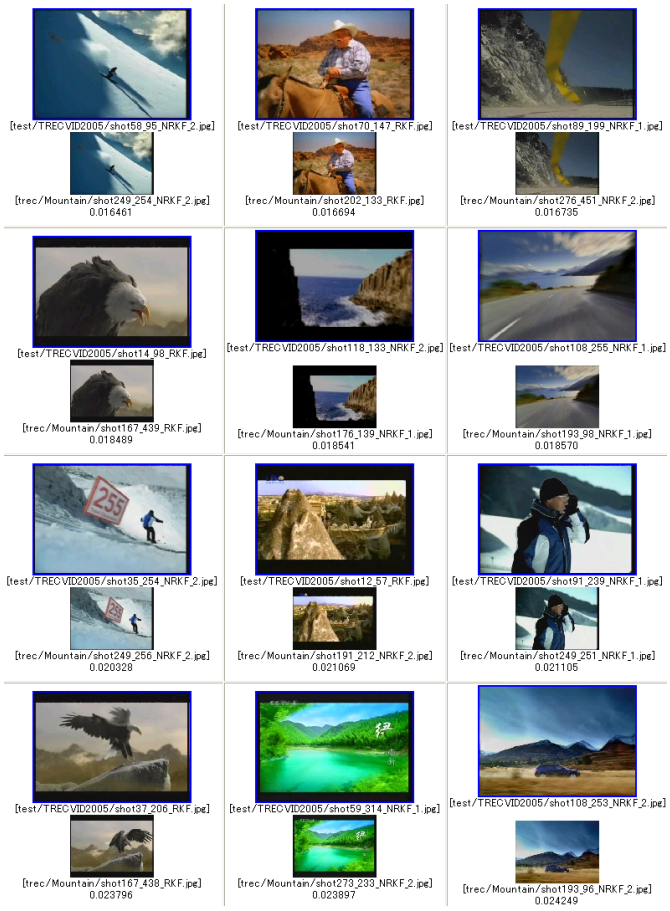

Figure 2: Identical "mountain" image pairs. The bigger images are images coming from test sets, and the small images are identical images found in the common annotation sets of "mountain".

5. CONCLUSION

We described our first attempt for TRECVID with the visual knowledge on the Web. Although the results were not good, we learned much about TRECVID. We believe that you can improve much at TRECVID 2006.

We think it is a big problem that many identical shots are included in both develop and test data sets. We cannot tell progress of methods from bonus results due to identical shots while using this data set. We hope the TRECVID committee treats with this problem appropriately at TRECVID 2006. We think one of the best ways is excluding all of the commercial shots from the TRECVID data set.

Table 2: The mean average precision of the results by the color histogram and k-nearest neighbor.

concepts	median	max	CH+kNN
38. People walking	0.145	0.346	0.0925
39. Explosion/fire	0.037	0.129	0.0031
40. Map	0.185	0.526	0.1485
41. US flag	0.071	0.253	0.0032
42. Building	0.236	0.511	**0.2248**
43. Waterscape	0.187	0.493	**0.4315**
44. Mountain	0.155	0.458	**0.2481**
45. Prisoner	0.001	0.056	0.0005
46. Sports	0.231	0.521	0.1075
47. Car	0.181	0.369	0.1036

6. REFERENCES

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[3] P. Duygulu, K. Barnard, J. d. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *The Seventh European Conference on Computer Vision*, pages IV:97–112, 2002.

[4] 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*, 2005.

[5] 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.

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

[7] K. Yanai and K. Barnard. Probabilistic web image gathering. In *Proc. of 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2005.