NHK STRL at TRECVID 2012: Semantic Indexing

Yoshihiko Kawai † Mahito Fujii †

†Science and Technical Research Laboratories, NHK,
1–10–11 Kinuta, Setagaya–ku, Tokyo, Japan

1 Introduction

The recent spread of digital video recording devices has made it commonplace for individuals to store large amounts of video in their equipment. This situation brings out the need for an effective retrieval method of finding desired scenes from accumulated video data. Broadcasting organizations also need efficient retrieval technology to enable effective use of their huge video archives.

To efficiently retrieve a desired video resource, it is important to analyze its semantic content rather than just its physical characteristics such as color or texture. The semantic indexing task in TRECVID aims at detecting semantic concepts such as objects and events. The studies about this task are also referred to as generic object recognition. One of the most common techniques for semantic indexing is the bag-of-visual-words (BoVW) method [1, 2], which calculates feature vectors on the basis of the occurrence frequency of local features such as SIFT [3] and SURF [4] and classifies them by a machine learning algorithm such as the support vector machine. The effectiveness of the BoVW approach has been verified in many previous studies [5].

In this paper, we propose a detection method of semantic concepts based on the BoVW approach. The proposed method uses global features that take a wider region into consideration, in addition to the local features of the conventional BoVW method. The feature vectors are calculated for block regions of various sizes in order to obtain robustness against variations in the size or position of objects. The random forests method [6] is used to determine whether the shot contains a specific concept. Semi-supervised learning is used to improve the quality of the learning data. Results of an evaluation experiment demonstrated that the proposed method had a sufficient accuracy in the detection of 346 different concepts.

2 Semantic Indexing Task

An overview of the proposed method is shown in Fig. 1. First, the input video is divided into shots, and then a keyframe is extracted from the beginning of each shot. The proposed method uses these keyframes as representative images for each shot. Next, local features and global features are calculated from the keyframes to obtain a feature vector. Local features are obtained by using the BoVW approach based on two types of algorithm: SIFT [3] and SURF [4]. Global features are calculated using color moments, the Haar wavelet, and local binary patterns (LBPs) [7]. These local and global features are aggregated over block regions of various sizes and then connected to form a feature vector for the entire frame. Finally, the feature vectors are classified by a random forest classifier to determine whether the target concept appears in the shot. The random forest classifier uses learning data labeled with concept names and is assumed to have been trained in advance. The details of each process are described below.

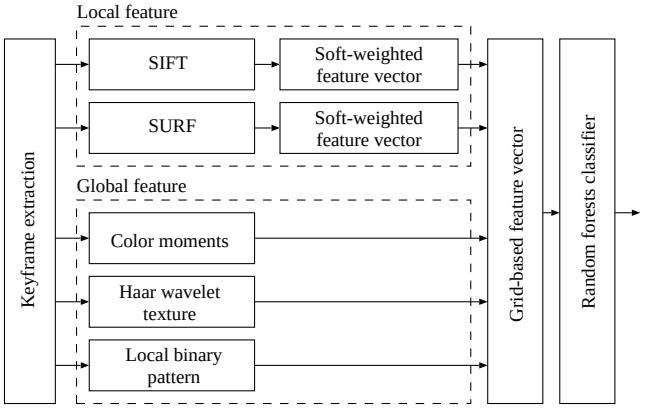

Figure 1: Overview of semantic indexing method.

2.1 Local Feature

The proposed method calculates local features based on the combination of SIFT and SURF algorithms to capture the visual characteristics of keyframes more accurately. The visual vocabularies are generated by using the k-means method to cluster feature descriptors extracted from the training data. Separate visual vocabularies are prepared for SIFT and SURF. The local feature vectors are calculated using a weighting method on the basis of the distances between a visual vocabulary and feature descriptors [8]. This differs from the conventional approach, where a single feature point is allocated to a single visual vocabulary item, and instead allows a single

feature point to be associated with multiple visual vocabulary items. If K is the total number of items in the visual vocabulary, it calculates a K-dimensional feature vector $T = (t_1, ..., t_k, ..., t_K)$. The vector elements t_k are calculated with the formula

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(p_j, w_k), \quad (1)$$

where M_i is the total number of feature points having w_k as a visual vocabulary item whose distance is close to the i-th item and $sim(p_j, w_k)$ is the degree of similarity between feature point p_j and visual vocabulary item w_k. N is a constant expressing how many of the closest visual vocabulary items should be considered. We set $N = 4$, as in reference [8].

2.2 Global Features

The proposed method uses three types of global feature.

2.2.1 Color Moments

The proposed method transforms the input image into the HSV and Lab color spaces and then calculates the average pixel value μ_c, the standard deviation σ_c, and the cube root of skewness s_c for each component c ($c \in \{h, s, v, l, a, b\}$). The calculation formulae are as follows:

$$\mu_c = \frac{1}{HW} \sum_x \sum_y f_c(x, y), \quad (2)$$

$$\sigma_c = \left\{ \frac{1}{HW} \sum_x \sum_y \{f_c(x, y) - \mu_c\}^2 \right\}^{1/2}, \quad (3)$$

$$s_c = \left\{ \frac{1}{HW} \sum_x \sum_y \{f_c(x, y) - \mu_c\}^3 \right\}^{1/3}, \quad (4)$$

where $f_c(x, y)$ represents the pixel value of a component c at coordinates (x, y) and $HandW$ are the height and width of the image region.

2.2.2 Haar Wavelet Texture

First, a Haar wavelet transform is applied in three levels to the image region. We then calculate the variance of the pixel values in each sub-band region, and these are concatenated to form the feature quantities.

2.2.3 Local Binary Pattern

The local binary pattern $L_{P,R}$ from P pixels on a circle of radius R is formulated as

$$L_{P,R}(x, y) = \begin{cases} \sum_{p=0}^{P-1} \delta_{P,R}(x_p, y_p), & if\ U_{P,R}(x, y) \leq 2 \\ P + 1, & otherwise \end{cases} . (5)$$

Here, $\delta_{P,R}$ represents the magnitude relationship of intensity values between a particular pixel (x, y) and the surrounding pixels $(x + x_p, y + y_p)$ and is calculated as

$$\delta_{P,R}(x_p, y_p) = \begin{cases} 1, & f(x+x_p, y+y_p) - f(x, y) \geq 0 \\ 0, & otherwise \end{cases} . (6)$$

The values of x_p and y_p are given by

$$\begin{cases} x_p = R \cos \frac{2\pi p}{P} \\ y_p = R \sin \frac{2\pi p}{P} \end{cases} \quad (0 \leq p \leq P - 1) . \quad (7)$$

The function $U_{P,R}$ in Equation (5) represents the total number of locations where there is a change between 0 and 1 in the sequence $\delta_{P,R}$ for the surrounding pixels, and is calculated by

$$U_{P,R}(x, y) = |\delta_{P,R}(x_{P-1}, y_{P-1})| + \sum_{p=1}^{P-1} |\delta_{P,R}(x_p, y_p) - \delta_{P,R}(x_{p-1}, y_{p-1})|. \quad (8)$$

The proposed method calculates $L_{P,R} (0 \leq L_{P,R} \leq P+1)$ for all the pixels in the image region and obtains their frequency histogram. To ensure robustness against changes of resolution, frequency histograms are calculated for each $L_{P,R}$ with $(P, R) = (8, 1), (16, 2)$, and $(24, 3)$ [7].

2.3 Calculation of Feature Vectors

The conventional method [9] partitions the keyframe images horizontally and vertically into 2×2 and 1×3 grid regions, and the average feature vectors for each grid region are concatenated to obtain a feature vector for the whole image. To calculate feature vectors more robustly with respect to differences in the size and position of objects, the proposed method partitions the frame images into grid regions of various sizes. In addition, to deal with objects that cross over the boundaries between grid regions, neighboring grid regions overlap each other. The average feature vectors are calculated for each grid region, and these are concatenated to form the feature vector of the entire keyframe. The specific size of the grid regions is obtained by dividing the frame image horizontally and vertically into 1×1, 2×2, 3×3, 3×1, 1×3, and 4×4 regions. Figure 2 shows the grid region sizes used in the proposed method. The amount of overlap between neighboring grid regions is 50% of the grid width in the vertical direction and 50% of the grid height in the horizontal direction.

2.4 Random Forests Classifier

The random forests method [6] is used to determine whether an input keyframe has a specific concept. Random forests is a kind of ensemble learning, and it provides highly accurate classifications by using a combination of decision trees (CART) [10]. Some researchers assert that random forests is superior to methods such as bagging

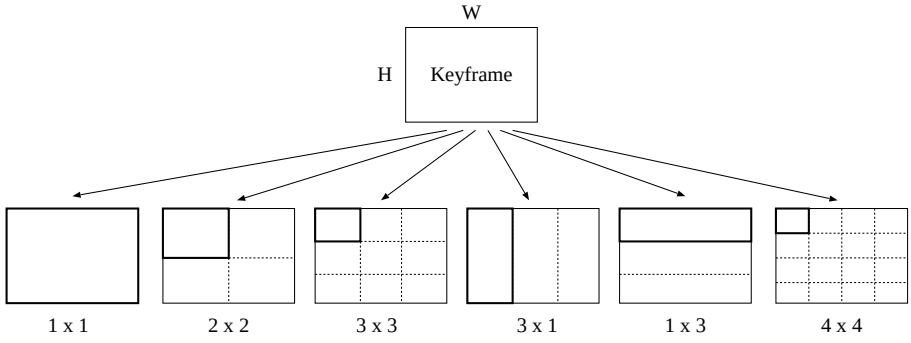

Figure 2: Calculation of grid-based feature vector.

or boosting in certain cases. In addition, random forests can complete the learning process in a short time even for high-dimension feature vectors by searching for the best feature for the branching node in a subset of vector elements.

The random forests algorithm works well when the training data for two classes (including and not including the concept) are roughly the same in number, but the classification error rate is rather uneven when one class is much larger than the other. The conventional method attempts to resolve this problem by applying a higher weight to the smaller class [6]. However, the bootstrap samples generated by the conventional method contain few data items with a high weight and many data items with a low weight, and this situation could cause overtraining. Thus, we propose a new sampling method for creating the bootstrap samples that ensures each class is selected with equal probability. The data is sampled with replacement and is not weighted. If the number of bootstrap samples is small relative to the amount of training data, various data items are additionally selected from the minority class, making it possible to generate a classifier with high generalization capability.

3 Experiments

3.1 Settings

For the full task of detecting 346 concepts, our team submitted four types of run (shown in Table 1). The training type was "A" in each case. These runs differed with regard to the methods used for feature vector calculation and classifier learning. The settings of each run are shown in Table 2. For the feature vectors, we used our proposed block-based feature vector for runs 1 and 3, and the conventional SIFT-BoVW feature vector for runs 2 and 4. For the learning method, runs 1 and 2 create random forest (RF) classifiers by using semi-supervised learning with collaborative annotation as a starting point, while runs 3 and 4 create classifiers by using the results of collaborative annotation directly. Run 4 is the baseline method.

Table 1: System ID and training type of each run.

Run	System ID	Training type
1	NHKSTRL1	A
2	NHKSTRL2	A
3	NHKSTRL3	A
4	NHKSTRL4 (baseline)	A

Table 2: Evaluation results of each run.

Run	Feature type	Training method	infAP
1	Block-based BoVW	Semi-supervised RF	0.106
2	SIFT BoVW	Semi-supervised RF	0.101
3	Block-based BoVW	RF	0.102
4	SIFT BoVW	RF	0.099

3.1.1 Experimental Results

The evaluation results are shown in Table 2. Each method obtained evaluation results with an inferred average precision (infAP) of around 0.1. The lowest accuracy was obtained with the baseline method (run 4), for which the infAP was 0.099. Next was run 2, which used semi-supervised learning and achieved an accuracy of 0.101. Runs 1 and 3, which used our block-based features, achieved better accuracy than the conventional SIFT-BoVW method. Specifically, the accuracy of run 3 (which used an ordinary learning method) was 0.102, while run 1 (which used semi-supervised learning) achieved the greatest accuracy, which was 0.106.

Table 3 shows the results of a detailed comparison between run 1, which achieved the highest infAP, and run 4, which used the baseline method. The comparison result shows that run 1 achieved greater accuracy in 31 out of 46 concepts. This increase in accuracy was particularly large for the concepts "101 Scene_Text" and "128 Walking_Running". In these concepts, we conclude that the image features were well reflected by the proposed feature quantities. In contrast, run 4 achieved greater accuracy for the concepts "4 Airplane_Flying" and "198 Civilian_Person". This over-fitting problem might occur in some concepts due to the increased number of dimensions

in the feature vectors. It may be necessary to investigate potential alternatives such as switching the feature quantity calculation method for some concepts.

4 Conclusion

In this paper, we proposed a method for calculating feature vectors that is robust against variations in the size or position of objects by combining image features calculated for block regions of various sizes. We also proposed a method for using semi-supervised learning to tidy up the training data. Evaluation results showed that the proposed method has a better detection accuracy than the conventional method. In our future work, we intend to continue improving the image feature vector in order to further improve the accuracy and to incorporate audio features as well as video features.

References

[1] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," In Proc. ICCV'03, 2003.

[2] G. Csurka, C. Bray, C. Dance and L. Fan, "Visual categorization with bags of keypoints," in Proc. ECCV Workshop on Statistical Learning in Computer Vision, pp.59–74, 2004.

[3] D.G. Lowe, "Object recognition from local scale-invariant features," In Proc. ICCV'99. vol.2. pp.1150–1157, 1999.

[4] H. Bay, A. Ess, T. Tuytelaars and L.V. Gool, "SURF: speeded up robust features," Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.

[5] "TREC video retrieval evaluation notebook papers and slides," http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[6] L. Breiman, "Random forests," Machine Learning, vol.45, pp.5–32, 2001.

[7] T. Ojala M. Pietikaninen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24, no.7, pp.971–987, 2002.

[8] Y.-G. Jiang, C.-W. Hgo and J. Yang, "Towords optimal bag-of-features for object categorization and semantic video retrieval," In Proc. ACM CIVR'07, 2007.

[9] S.-F. Chang, J. He, Y.-G. Jiang, E.E. Khoury, C.-W. Ngo, A. Yanagawa and E. Zavesky, "Columbia University/VIREO-City/IRIT TRECVID2008 high-level feature extraction and interactive video search," In Proc. TRECVID 2008 Workshop, 2008.

[10] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, "Classification and regression trees," Wadsworth and Brooks, 1984.

[11] S. Ayache and G. Quenot, "Video corpus annotation using active learning", In Proc. ECIR'08, 2008.

Table 3: Evaluation results for each concept.

Concept		infAP	
		Run 1	Run 4 (Baseline)
3	Airplane	0.099	0.094
4	Airplane_Flying	0.147	0.173
9	Basketball	0.022	0.031
13	Bicycling	0.011	0.007
15	Boat_Ship	0.051	0.050
16	Boy	0.027	0.024
17	Bridges	0.018	0.026
25	Chair	0.033	0.039
31	Computers	0.009	0.004
51	Female_Person	0.194	0.187
54	Girl	0.021	0.010
56	Government-Leader	0.162	0.123
57	Greeting	0.051	0.056
63	Highway	0.030	0.035
71	Instrumental_Musician	0.093	0.080
72	Kitchen	0.010	0.009
74	Landscape	0.334	0.328
75	Male_Person	0.616	0.620
77	Meeting	0.084	0.063
80	Motorcycle	0.057	0.044
84	Nighttime	0.042	0.049
85	Office	0.076	0.080
95	Press_Conference	0.030	0.025
99	Roadway_Junction	0.034	0.022
101	Scene_Text	0.073	0.011
105	Singing	0.018	0.029
107	Sitting_Down	0.002	0.002
112	Stadium	0.076	0.090
116	Teenagers	0.041	0.030
120	Throwing	0.018	0.018
128	Walking_Running	0.222	0.163
155	Apartments	0.059	0.056
163	Baby	0.038	0.033
198	Civilian_Person	0.733	0.756
199	Clearing	0.089	0.097
254	Fields	0.175	0.161
267	Forest	0.117	0.090
274	George_Bush	0.087	0.054
276	Glasses	0.051	0.050
297	Hill	0.105	0.090
321	Lakes	0.124	0.137
338	Man_Wearing_A_Suit	0.169	0.138
342	Military_Airplane	0.085	0.082
359	Oceans	0.170	0.173
434	Skier	0.108	0.074
440	Soldiers	0.046	0.045
Average		0.106	0.099