TokyoTech's TRECVID2006 Notebook

Taichi Nakamura, Yuichi Miyamura, Koichi Shinoda and Sadaoki Furui
Department of Computer Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552 Japan
{nakamura,miyamura}@ks.cs.titech.ac.jp
{shinoda,furui}@cs.titech.ac.jp

In this notebook we describe our TRECVID 2006 experiments. We TokyoTech team participated in shot boundary detection and high-level feature extraction tasks.

1 Shot Boundary Detection

Our approach to shot bondary detection uses SVMs with generic features. Using the radial kernel for SVMs, we ignore the difference among the types of gradual transitions (i.e. FOI, DIS, and OTH).

We classify shot boundaries into the following three categories.

- Cuts(CUT)
- Gradual transitions with five frames or less (Short Gradual; SG)
- Gradual transitions with more than five frames (Long Gradual; LG)

We prepare a kernel function and a feature set for each of these categories.

1.1 Cut Detection

Since shot boundaries with less than five frames are classified as "cuts" in the TRECVID evaluation, the results for SG are added to the results in CUT, and are submitted as the results for "cuts". For the cut detection, we use two linear kernel SVMs (one for CUT and the other for SG) with different feature sets. The features for a CUT-SVM are activity ratio (the ratio of "dynamic" pixels to all pixels, where each dynamic pixel has larger difference than a predetermined threshold), the opticalflow, the change in the Hue-Saturation color histogram and edge. The features for SG-HMM are the activity ratio and the change in the Hue-Saturation color histogram.

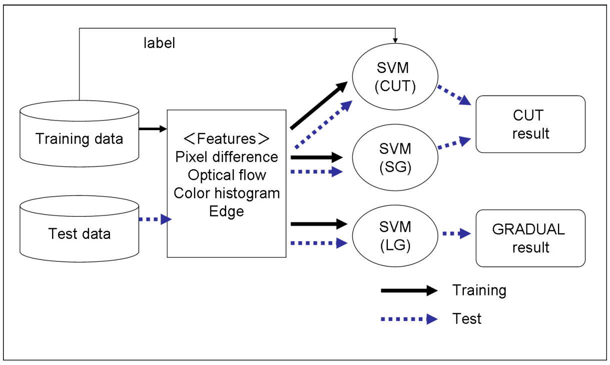

Figure 1: The shot boundary detection framework.

run	training data portion	SG cost	LG cost	number of thresholds for activity ratio	ALL		CUT		GRAD		GRAD(frame)	
					R	P	R	P	R	P	R	P
1	first half	1	2	2	79.7	82.2	87.0	81.9	59.9	83.7	66.1	94.0
2	last half	1	2	2	79.9	82.3	87.2	81.9	60.2	83.7	65.4	94.1
3	all	1	2	2	79.8	82.1	87.1	81.8	60.2	83.6	65.4	94.1
4	all	2	2	2	81.3	81.4	89.1	81.0	60.2	83.0	65.4	94.1
5	all	1	2	1	79.1	83.3	85.9	83.3	60.5	83.4	65.4	94.1
6	all	2	2	1	80.6	82.6	88.0	82.6	60.4	82.7	65.5	94.1
7	all	1	3	2	80.8	81.0	87.2	81.3	63.4	79.8	67.3	92.6
8	all	2	3	2	82.2	80.3	89.2	80.6	63.3	79.3	67.4	92.6
9	all	1	3	1	80.0	82.1	86.0	82.8	63.7	79.7	67.3	92.6
10	all	2	3	1	81.5	81.4	88.1	82.1	63.5	79.1	67.5	92.6

Table 1: The conditions and performance of submitted ten runs

Gradual transition	Recall	Precision
SVMs	60.93	78.26
SVMs + additional detector	70.80	81.23

Table 2: Comparison of the performance of the main and additional experiments.

1.2 Gradual Detection

For gradual detection, we use one radial kernel SVM (LG-SVM). The features for this SVM are the difference (an average of absolute differences between successive two frames), the optical flow, the change in the Hue-Saturation color histogram and edge. When the frames detected as CUT/SG are overlapped with those detected as LG, we classify them as CUT/SG.

1.3 Experiment

We used test data of TRECVID 2005 shot boundary detection to train these SVMs. We selected feature type and parameters of SVMs based on 2-fold cross-validation. Our 10 runs differ in the portion of data used for training, cost-factors of SVMs, and the number of thresholds for activity ratio. By changing the cost-factor of SVM, we can emphasize the positive examples (i.e, shot boundary).

Table 1 shows the whole results of our submission. Comparing our detection results with correct labels, we found that many of the LG transitions our system failed to detect involved luminance changes (e.g, FOI). This is probably because we employed the Hue-Saturation color space to obtain roboustness against luminance variation (e.g, flashlights). To overcome this problem, we prepared an additional simple detector using the Saturation-Value color space dedicated to these transitions, and conducted an additional experiments with 2-fold cross-validation on the TRECVID 2005 test data. Table 2 compares the performance of our former system and that of the system with the additional detector. The additional detector significantly improved recall rates while maintaining precision rates.

2 High-level Feature Extraction

For the high-level feature extraction task, we use only visual features of keyframe images. Maximum entropy models [1] are employed to model these visual features.

2.1 Visual Features

We use a sparse image representation [2] based on affine-invariant regions, which are thereafter represented by visual words and color features. Figure 2 briefly shows the visual feature extraction process.

Affine-invariant regions are detected by Harris-Affine and Hessian-Affine detectors. To obtain affine invariance and rotation invariance, the regions are normalized to a circular region, then rotated to the

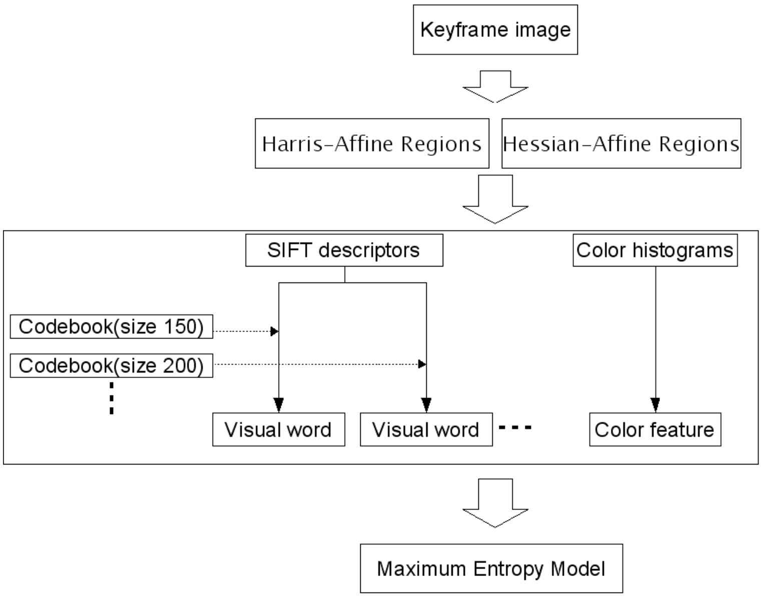

Figure 2: Overview of our visual feature extraction process

direction of the dominant gradient orientation. Finally we compute two kinds of features representing the region in the following way.

Visual words The region is first described with a 128 dimensional SIFT descriptor. This descriptor is then quantized with codebooks, which are constructed in advance for each high-level feature. We call this quantized descriptor “visual word”.

Color features For each region, a 100-dimensional histogram in the Hue-Saturation color space is computed. Each region is considered to have a color feature when more than half of its pixels are in the single corresponding bin of the color histogram.

2.2 Classifiers

We use a maximum entropy model (MEM) [1] to classify the presence/absence of each of the high-level features. A MEM estimates the posterior distribution of label (presence or absence) given the features of a keyframe image. We use the implementation of MALLET [3].

2.3 Feature Selection

Color information is effective for some high-level features, but not so effective for others. Similarly, each high-level feature has a suitable codebook size. Therefore, for each high-level feature, we tried to select the combination of feature types (e.g, color features, visual words from codebook size 150, visual words from codebook size 200, and so on) by 5 fold cross-validation. In this cross-validation, 70% of the videos were randomly selected as a training set, and remaining 30% were used for testing.

2.4 Experiments

We used the TRECVID 2005 training data set to train the MEM. The A_Tech1_2 run was trained on the complete training set using all the features. On the other hand, for A_Tech1_1 run, the feature selection technique mentioned above was used. We constructed a codebook for each high-level feature by clustering SIFT descriptors from keyframe images having the high-level feature. When there existed more than 2000 relevant keyframes, randomly sampled 2000 keyframes were used. Figure 3 shows the classification performance of our 2 runs (A_Tech1_1 and A_Tech1_2). Most of our performance is below

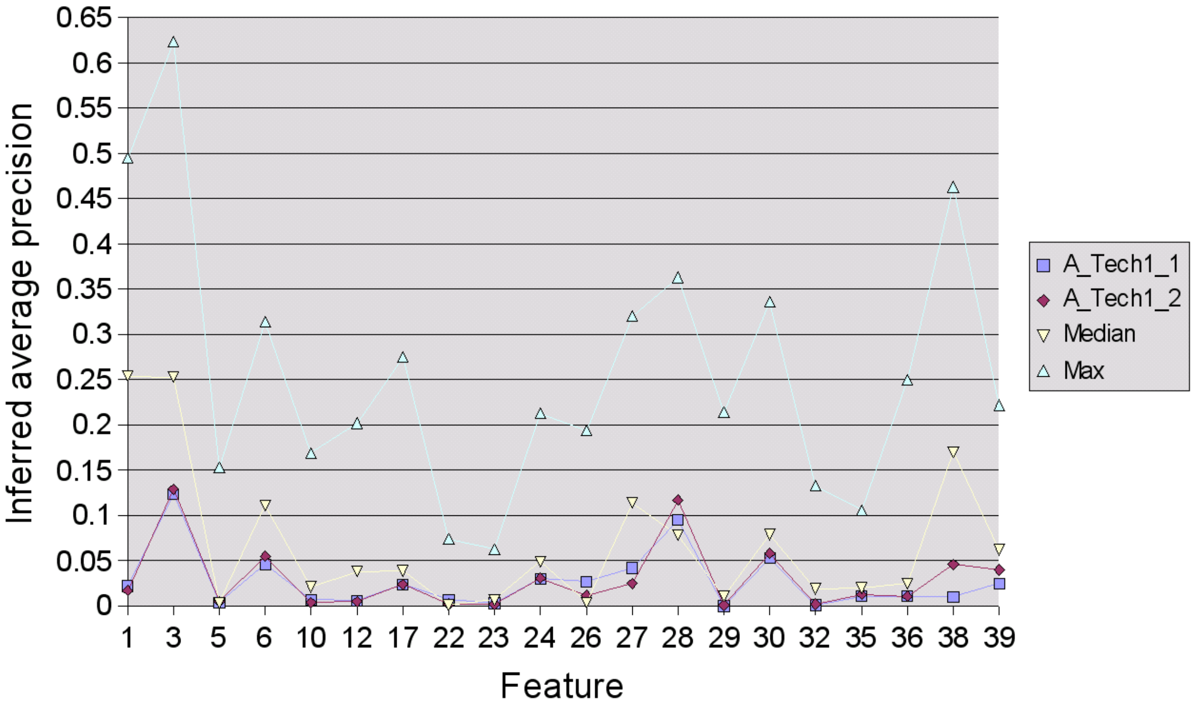

Figure 3: Performance of A_Tech1_1 and A_Tech1_2 in terms of inferred average precision

the median except flag-US and animal. We also found that there is no big difference of performance between the two runs.

2.5 Conclusions and Future works

In this paper, we reported experimental results for our preliminary system. To improve the performance, our future work includes the following topics.

- To enhance the visual features (e.g, motion, shape context descriptor).
- To utilize the transcription of speech data.
- To model the temporal correlation and the inter-class relation of high-level features.

References

[1] A. Berger, Stephen A. Della Pietra, , and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22(1), pp. 39–71, 1996.

[2] S. Agarwal and D. Roth, Learning a sparse representation for object detection, In *ECCV*, pages, 113130, 2002.

[3] McCallum, Andrew Kachites, "MALLET: A Machine Learning for Language Toolkit.", http://mallet.cs.umass.edu. 2002.