UEC at TRECVID 2008 High Level Feature Task

Zhiyuan Tang and Keiji Yanai
Department of Computer Science, The University of Electro-Communications, JAPAN
{tou-s,yanai}@mm.cs.uec.ac.jp

Abstract

In this paper, we describe our approach and results for high-level feature extraction task (HLF) at TRECVID2008. This year, our focus is to develop a framework which fuses a number of features effectively. In our paper, color , face, motion, text, and local pattern features were extracted. After that, a simply-modified version of Adaboost algorithm was implemented as a late fusion to combine all these features.

Description of our submitted runs is as follows:

- *(Run1)UEC_fusion_ver6,*
  (Run2)UEC_fusion_ver2,
  (Run3)UEC_fusion_ver1,
  (Run4)UEC_fusion_ver5:

  fusion of color, face, motion, text and Bag-of-Features (BoF) model of local pattern features by using a simple version of Adaboost. use different setting to compute error rate in the fusion phase.

- *(Run5)UEC_fusion_c_bdd_ver6,*
  (Run6)UEC_fusion_c_bdd_ver2:

  fusion of color, face, BoF model of local pattern features by using a simple version of Adaboost. use different setting to compute error rate in the fusion phase.

Run1∼Run4 are the same to combine color, face, motion, local pattern features by using our algorithm. In our experiment, we changed some parameters when computing the error measure in the fusion algorithm, this makes the 4 runs different from each other. By the analysis of the results of these 4 runs, we noticed that motion and text did not help us at all, so we also tried to fuse only color, face, local pattern features as Run5 and Run6 with different error measure computation. As a result, Run5 yielded the best performance (infAP=0.0314) of these our 6 runs.

1. Introduction

Since TRECVID provides not only a large video date set but also a systematic protocol for evaluating video concept detection performance, it is appreciated by the researchers in the field of video/image recognition. Using this valuable date set, we have been testing our system in recent years.

For the HLF task in TRECVID2006, we extracted some single type visual features from the keyframes (for example, color histogram, edge histogram, etc.), and classified test frames by the support vector machine (SVM). From the results, we realized that a certain feature cannot satisfy all the concepts. For TRECVID2007, we attempted to adopt a kind of fusion to combine some features to get a result that is effective for any kind of concept. What we did is to apply SVM to the extracted features respectively, and then to fuse these SVM classifiers by linear combination with weights selected by cross validation. This method is more effective, however it is intractable to implement when more than 3 kinds of features are extracted.

For the TRECVID2008 HLF task, we still use the thought of developing a framework to fuse a number of features to get more effective performance. This time we added some new features. In addition, inspired by some papers [2, 8], we implemented a simple version of Adaboost [6] algorithm as a late fusion.

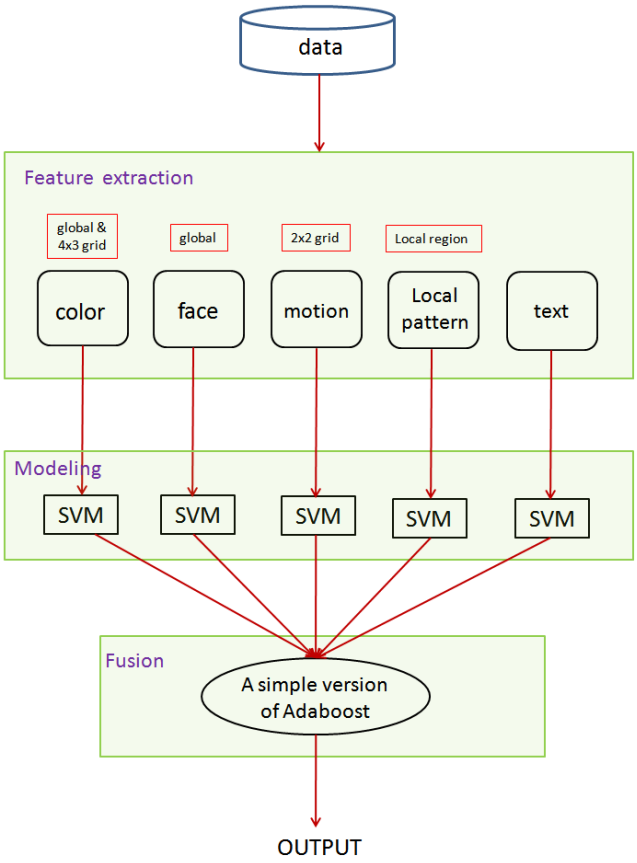

Figure 1. UEC HLF framework.

This method can choose the suitable weights automatically no matter how many kinds of features there are.

2. Framework

The framework is shown in Figure 1. At the first stage, color, text, face, motion and local pattern features of the learning/test data are extracted from different granularity of global scale, local region and grid segmentation. Then SVM is applied to modeling all the features respectively. At last stage, a simple version of Adaboost is implemented as a late fusion to combine all the SVM classifiers to obtain the final output.

2.1. Features

2.1.1. Color

In the experiment, we use a normal color histogram as the color feature. The axises of RGB color space are divided in quarters and a 64-bin histogram is generated. For getting some location information, besides extracting from global scale of the image, we also tried to extract a 768 bins histogram by dividing the image to 4×3 grid segments.

2.1.2. Text

Automatic speech recognition (ASR) text data is provided by the sponsor every year. We use this ASR text to make a text feature. We choose 2000 representative words. Then we count their global frequency in the whole text data and the local frequency in every shot. At last a 2000 bins histogram is generated by using tf-idf algorithm.

2.1.3. Face

We perform a face detection by using Haar-like features [7]. The number of faces is expected to help handle with "Two People" concept.

2.1.4. Motion

The Lucas-Kanade's optical flow [4] is used as our motion feature. We extract the frames 0.5 seconds before and after each keyframe and choose 500 interest points from them. The circle (360 degrees) is divided to 12 equal parts. And the motion feature is generated by voting the magnitude of the optical flow of each point to the corresponding region according to their angular degree.

2.1.5. Local pattern

We use SIFT [3] as the local pattern feature. The local patches are detected by three ways : (1) DoG (2) random sampling [5] (3) grid (shown in Figure 2). The bag-of-keypoints [1] model is used to represent the whole image. The codebooks are obtained by performing the k-means clustering and the vector is generated by voting the SIFT descriptors of each image to the codebook pattern. In our experiment, the codebooks are computed for each concept respectively and every codebook size is 1000.

2.2. Fusion

In TRECVID2007, we fused the SVM classifiers by linear combination with weights selected by cross val-

Figure 2. 3 kinds of local patch detection.

idation. It is intractable to implement when more than 3 kinds of features are extracted. So this year, we implemented a method that using boosting scheme. This method can choose the suitable weight automatically whatever how many kinds of features there are.

Boosting is a powerful technique for combining multiple classifiers to produce a new one whose performance can be significantly better than that of any of the components. Adaboost is one of the representations of Boosting. It trains a new classifier according to the performance of the previously trained classifier so as to give greater weight to the misclassified data points. Finally when the desired number of classifiers have been trained, they are combined to form a great classifier using coefficients that give different weight to different component classifiers.

In our experiment, we implement a simple version of Adaboost. The algorithm is shown in Figure 3.

While in the original Adaboost re-weighting for training data and re-training are performed in its loop, in our simply-modified Adaboost algorithm we only train SVM using all the learning data before the boosting loop and fuse the result with the error measure.

Given: Training data X ($|X| = n$), with its positive subset X_1 and X_0, where $y(x_i) = 1$ if $x_i \in X_1$ and $y(x_i) = -1$ if $x_i \in X_0$.

1. Initialize the data weight w :

$$w_1(x_i) = \begin{cases} \frac{1}{2|X_1|} & if\ x_i \in X_1 \\ \frac{1}{2|X_0|} & if\ x_i \in X_0 \end{cases} \quad (1)$$

2. Train SVM classifier h_j with X for all the features, where j is the number of the features.

3. For t = 1, . . . ,T:
   - Normalize the weighting coefficients :
   $$w_{t,i} = \frac{w_{t,i}}{\sum_{m=1}^{n} w_{t,m}} \quad (2)$$
   - Evaluate the error coefficients for all the features:
   $$e_i = |h_j(x_i) - y_i| \quad (3)$$
   $$\epsilon_j = \sum_i w_i e_i \quad (4)$$
   - Select the classifier h_t which has the minimum error ϵ_t.
   - Update the weighting coefficients :
   $$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \quad (5)$$
   $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i} \quad (6)$$

4. Output the combined classifier:

   for test data x_k :

$$h(x_k) = \sum_{t=1}^{T} \alpha_t h_t(x_k) - \frac{1}{2}\sum_{t=1}^{T} \alpha_t \quad (7)$$

$$\text{where } \alpha_t = log\frac{1}{\beta_t}$$

Figure 3. The simple version of Adaboost.

3. Experiments

We made 6 runs as shown in Table 1. Firstly, we tried to combine all the features we had with our framework.

In our feature extraction phase, there was some problems: (1) Since some 0.5 seconds separated frame was extremely different, the motion feature could not be computed. (2) There was some shot without sound, so the text feature could not be computed. (3) Also, some image was pure black, we did not compute the local pattern feature and the color feature of them. That means, for a complete test data list, not all the data in it have all the features extracted. Address these problems, we changed some options in the algorithm shown in Figure 3:

Run1 : Set threshold = the mean value of the weight of all data. If e_i(in eq.(3)) $\geq$ the mean value, treat the data without feature h as misclassified, else treat it as correct classified.

Run2 : Set e_i = 1. (just treat the data without feature h as misclassified).

Table 1. 6 runs for HLF in TRECVID2008.

Runs	Description	infAP
Run1 UEC_fusion_ver6	Combine color, face, motion, text and BOF model of local pattern features. Boosting error measure setting : threshold=mean weight.	0.0297
Run 2 UEC_fusion_ver2	Combine color, face, motion, text and BOF model of local pattern features. Boosting error measure setting : misclassified.	0.0278
Run3 UEC_fusion_ver1	Combine color, face, motion, text and BOF model of local pattern features. Boosting error measure setting : mean value.	0.0187
Run4 UEC_fusion_ver5	Combine color, face, motion,text and BOF model of local pattern features. Boosting error measure setting : threshold=0	0.0314
Run5 UEC_fusion_c_bdd_ver6	Combine color, face and BOF. Boosting error measure setting : threshold=mean weight.	0.0342
Run6 UEC_fusion_c_bdd_ver2	Combine color, face and BOF. Boosting error measure setting : misclassified.	0.0299

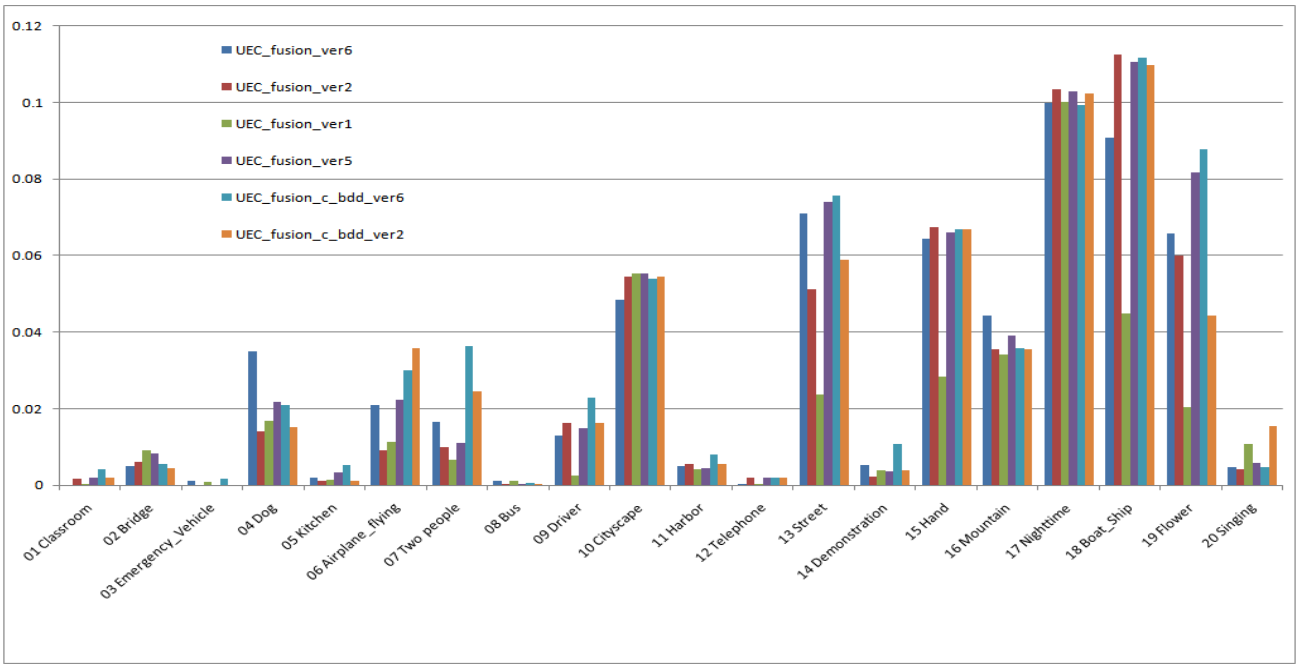

Figure 4. The comparison of 6 runs.

Run3 : Set e_i = 0.5. (set it as the mean value in [0,1])

Run4 : set threshold =0. If $e_i \geq 0$, treat the data without feature h as misclassified, else treat it as correct classified.

Since the number of data without motion and text feature was large, we also tried to take off these 2 features and only fuse color, face and local pattern features:

Run5 : the same setting as Run1.

Run6 : the same setting as Run2.

The 6 runs for each concept are compared as Figure 4. And our best result (run5) are compared with the best and mean result of participators as Figure 5.

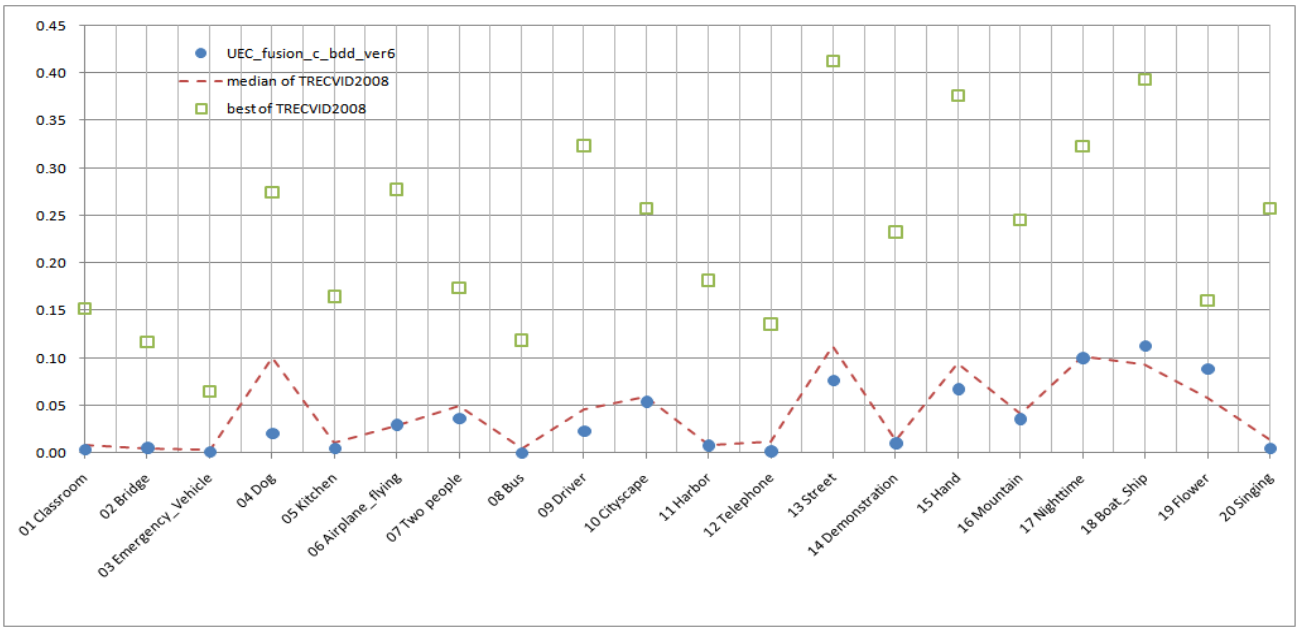

Figure 5. The comparison with the median and best results in TRECVID 2008.

From this result, we noticed that the motion feature and the text feature gave the opposite effect to the output actually, and it is enough to fuse only the static visual features to get an effective result.

4. Conclusions

In the high-level feature extraction task of TRECVID2008, we implemented a simple version of Adaboost as a late fusion to combine color, text, face, motion and local pattern features. This method can choose the suitable weight for every automatically no matter how many kinds of features there are. Since some learning data could not have the motion and text features extracted, they were hard to be handled in the algorithm. Finally, only the fusion of color, face and local pattern feature got the best result out of our 6 runs. For future work, we will try some early fusion methods such as multiple kernel learning SVM to improve our system.

References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *Proc. of ECCV Workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.

[2] W. Jiang, S. Chang, and A. Loui. Kernel sharing with joint boosting for multi-class concept detection. In *Proc. of CVPR Workshop on Semantic Learning Applications in Multimedia*, 2007.

[3] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[4] 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*, volume 3, 1981.

[5] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *Proc. of European Conference on Computer Vision*, 2006.

[6] R. Schapire, Y. Freund, and R. Schapire. Experiments with a New Boosting Algorithm. In *Proc. of International Conference on Machine Learning*, pages 148–156, 1996.

[7] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proc. of IEEE Computer Vision and Pattern Recognition*, volume 1, 2001.

[8] D. Wang, X. Liu, L. Luo, J. Li, and B. Zhang. Video diver: generic video indexing with diverse features. In *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 61–70, 2007.