Attribute-Based MED System with Word Histograms

Optical Research Laboratory, Nikon Corporation
Takeshi Matsuo and Shinichi Nakajima

November 7, 2011

1 Basic Concept

We used a similar system to our previous system 2010 [1], which we built for the last year competition. However, we need to cope with the notable difficulty imposed in the TRECVID 2011 MED task, i.e., the set of the *target* event classes and the set of the *training* event classes are completely disjoint. This means that no sample video belonging to any of the target event classes is available in the training phase. As a knowledge transfer method, we adopted the attribute-based classification (AC) approach [2]. The AC approach classifies a test sample into the set of the *target* classes by combining outputs from the classifiers learned for the *training* classes. This requires class similarities between the *target* classes and the *training* classes, in order to weight the classifier outputs. We used text words associated with each video as side information, and adopted the similarity between the word histograms as the class similarity [3].

2 Detailed Description of Our System

Our system consists of two training steps and a test step. In the first training step, we build classifiers for the *training* events. In the second training step, we calculate similarities between the *test* (*target*) events and the *training* events. In the test step, we classify the test samples, by weighting the classifier outputs, based on the similarities.

2.1 Training Step 1: Creating Classifiers

For this step, we basically adopted our previous system [1], built for TRECVID 2010. But we applied some minor changes for reducing computation time. In this subsection, we explain the differences from our previous system, which consists of the following steps;

1. Create a space-time (ST) image from a video,

2. Perform scenecut detection based on the ST image,
3. Extract keyframes from each scene,
4. Construct a bag-of-words (BoW) histogram from the set of keyframes,
5. Train the support vector machines (SVM) with the BoW histograms as input vectors.

2.1.1 Space-time Image Creation

In our 2010 system, each frame is resized into 40×30 pixels image. The space-time (ST) image is constructed by stacking all the vertical lines, so that the height of the ST image is 1200 pixels. The sampling frame rate was 2/FPS second.

On the other hand, our new system adopted "visual rhythm" [4], i.e., the ST image is constructed by stacking only two diagonal lines, so that the height of the ST image is 80 pixels. The sampling frame rate is 0.5 second. This modification substantially reduced the computation time.

2.1.2 Scenecut Detection

Our 2010 system used the Canny edge detector and the Hough voting for scenecut detection. We instead adopted the following procedure for our new system. First, a y-directional (21 pixels) median filter is applied to the ST image (an example pair of original ST image and filtered image are shown in Figure 1).

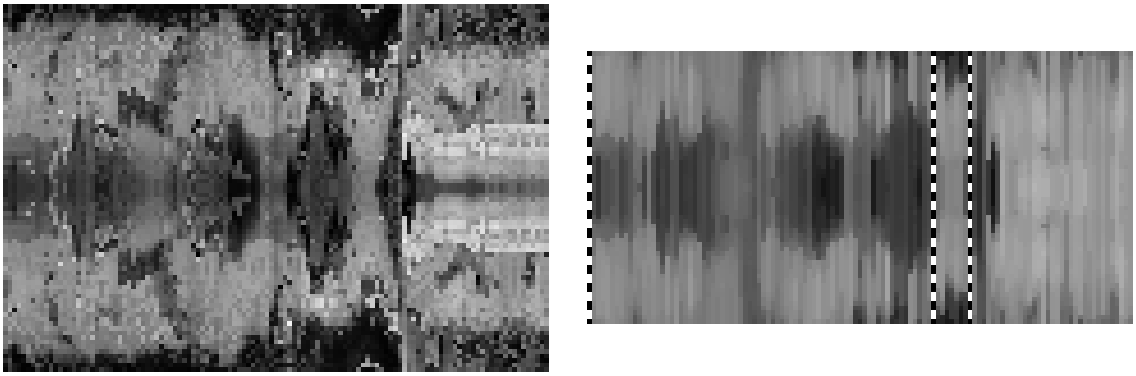

Fig. 1: A space-time (ST) image (left) and the median filtered image (right). The dashed lines in the right image indicate the detected scenecuts with $\theta = 0.18$.

Let $w = \lfloor \text{duration} \times 0.5 \rfloor$ be the width of the ST image, and $I_{\text{mid}}(x, y)$ be the normalized intensity of the median filtered image at the position (x, y). Then, our system calculates

$$d_x \equiv \frac{1}{60} \sum_{y=1}^{60} |u_{x-s,y} - u_{x,y}|, \qquad u_{x,y} \equiv \frac{1}{s} \sum_{k=0}^{s-1} I_{\text{mid}}(x + k, y) \tag{1}$$

for $x = s, s+1, s+2, \ldots, w-s-1$. If d_x is larger than a threshold θ and $x > x' + s$, where x' is the previous scenecut position, our system makes a scenecut at the time x. We set $s = 3$ and $\theta = 0.18$, based on our preliminary experiment.

2.1.3 Keyframe Extraction

No change has been done. Our new system extracts 2 frames at each 2 longest scenes for each video.

2.1.4 Bag-of-Words Histogram Construction

Our 2010 system created visual words based on the SIFT [5] descriptor. Our new system instead uses the SURF [6] and the color average features.

The color average feature is 9-dimensional and calculated as follows: First we throw away the pixels within 1/40 of the width (or height) to the edge. Then, each image is divided into 3 (lower, middle, and higher) regions, each of which has the same height. After that, in each region, each channel of the RGB vector is averaged over the pixels.

We construct two bags-of-words [7], based on the SURF and the color average features, respectively.

2.1.5 Classification with Support Vector Machine

Our 2010 system used LIBSVM [8] with χ^2 kernel. Our new system uses LIBSVM with the linear kernel for reducing computation time. The *cost* parameter (balancing the loss and the regularization terms) is optimized by grid search with 2-fold cross validation.

2.2 Training Step 2: Calculating Similarities for Event Knowledge Transfer

To calculate similarities between events, we used the text words given in the knowledge sources (`*_JudgementMD.csv`) and in the event definition files (`E001.txt,...,E015.txt`). We calculated word histograms of each event after excluding stop words and infrequent words. We say a word is infrequent if it appears less than or equal to n times over the whole data set.

Let $\boldsymbol{a}_i$ be the word histogram of the i-th event. We define as the event similarity between the i-th and the j-th events the correlation coefficient between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$:

$$c_{j,i} \equiv \frac{(\boldsymbol{a}_i, \boldsymbol{a}_j)}{|\boldsymbol{a}_i|\,|\boldsymbol{a}_j|} = \frac{\sum_{k=1}^{K} a_{i,k}\, a_{j,k}}{\sqrt{\left(\sum_{k=1}^{K} a_{i,k}^2\right)\left(\sum_{k=1}^{K} a_{j,k}^2\right)}}, \tag{2}$$

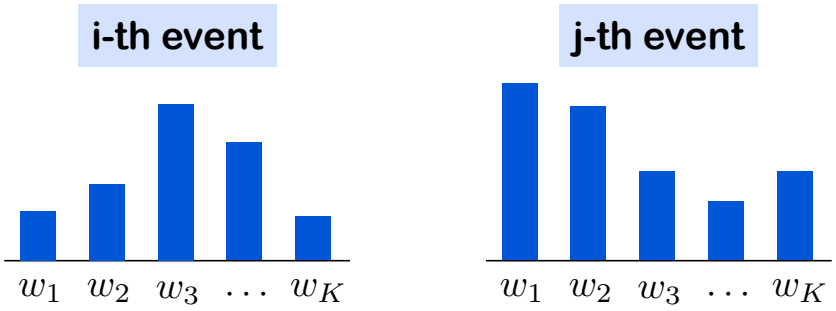

Fig. 2: Illustration of word histograms for the two events. Similarity between two events is defined as the correlation between the histograms.

where

$$\boldsymbol{a}_i \equiv \begin{pmatrix} a_{i,1} \\ a_{i,2} \\ \vdots \\ a_{i,K} \end{pmatrix}, \tag{3}$$

and K is the number of histogram bins.

2.3 Test Event Detection Step

We use the *probability output* [9] from SVM. Let v be a video clip in the test data, and $Y_i = (Y_{i,1}, Y_{i,2})$ be the output probability from the SVM for the i-th event. $Y_{i,1}$ and $Y_{i,2}$ are the probability outputs based on the SURF and the color average features, respectively. Then, we merge the outputs based on the two features by

$$q_i(v) \equiv (Y_{i,1} \times Y_{i,2})^{\frac{1}{2}}, \qquad i \in \{\textit{training} \text{ events}\}. \tag{4}$$

This is our output probability that the test sample belongs to the i-th event.

To convert the probabilities of the *training* events into the probability of a *test* event, we use the sigmoid function of the weighted sum of probabilities:

$$p_j(v) \equiv \tanh\left(\alpha \sum_{i \in \{\textit{training} \text{ events}\}} c_{j,i}\, q_i(v)\right), \qquad j \in \{\textit{test} \text{ events}\}. \tag{5}$$

Here, α is a parameter to adjust the slope of the probability increase, which we set to $\alpha = 2$.

2.4 Data Set

In Training Step 1, we trained 8 classifiers for the *training* events. (E001, ..., E005 in DEVT and EVENTS, and P001, ..., P003 in MED10EVAL and MED10TRN). In Training

Step 2, we calculated the similarities of the *test* events (E006, . . . , E015) to the 8 *training* events.

2.5 System Hardware and Runtime Computation

We used a dual 3.6GHz Intel Xeon CPU (we used 1 CPU), with a 3.2GB RAM and a 1TB HDD storage. The computation time is as follows:

- Training step 1:
  - Scenecut including creating ST images: 57 hours,
  - Keyframe extraction: 18 hours,
  - Training linear SVMs: 2 hours,
- Training step 2:
  - Calculating similarities for knowledge transfer: 7 hours,
- Test step: 264 hours (11 days).

3 TRECVID 2011 Evaluation

We submitted our output for the evaluation data. Table 1 and Figure 3 show the results for $n = 8$, provided by NIST.

4 Experimental Result

We also evaluated our system, using only the data included in the development kit. We used all the video clips of the *training* events for training, and all the video clips of the *test* events for test, since there is no overlap between these two sets of data. However, we need to divide the video clips of the *null* events into the negative samples in the *training* data and the negative samples in the *test* data. We used a half of the *null* event data for training, and the rest for test.

Table 2 and Figure 4 show the *minimum normalized detection costs* for $n = 8, 16$.

References

[1] Takeshi Matsuo and Shinich Nakajima. Nikon multimedia event detection system, 2010. `http://www-nlpir.nist.gov/projects/tvpubs/tv10.papers/nikon.pdf`.

[2] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *Computer Vision and Pattern Recognition*, 2009.

[3] Marcus Rohrbach, Michael Stark, György Szarvas, and Bernt Schiele. Combining language sources and robust semantic relatedness for attribute-based knowledge transfer. In *Parts and Attributes Workshop at ECCV 2010*, 9 2010. Code and supplemental material are provided at `http://www.d2.mpi-inf.mpg.de/nlp4vision`.

[4] Silvio Jamil Ferzoli Guimarães, Michel Couprie, Arnaldo de Albuquerque Araújo, and Neucimar Jerônimo Leite. Video segmentation based on 2D image analysis. *Pattern Recognition Letters*, 24(7):947–957, 2003.

[5] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. `http://www.cs.ubc.ca/~lowe/keypoints/`.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and LucṼan Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[7] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 59–74, 2004.

[8] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

Table. 1: *Minimum normalized detection cost* with evaluation data in NIST. n is the threshold such that infrequent words that appear less than or equal to n are ignored in event similarity calculation.

n	8
E006	1.000
E007	1.000
E008	1.000
E009	0.999
E010	1.000
E011	1.000
E012	1.000
E013	0.990
E014	1.000
E015	1.000
avg.	0.999

Table. 2: *Minimum normalized detection cost* with development kit in our experiment. n is the same as in Table 1.

n	8	16
E006	1.003	1.003
E007	1.003	1.003
E008	0.982	1.003
E009	1.003	1.003
E010	1.003	1.003
E011	0.993	1.003
E012	1.003	1.003
E013	**0.700**	0.723
E014	1.003	1.003
E015	0.945	0.960
avg.	**0.964**	0.971

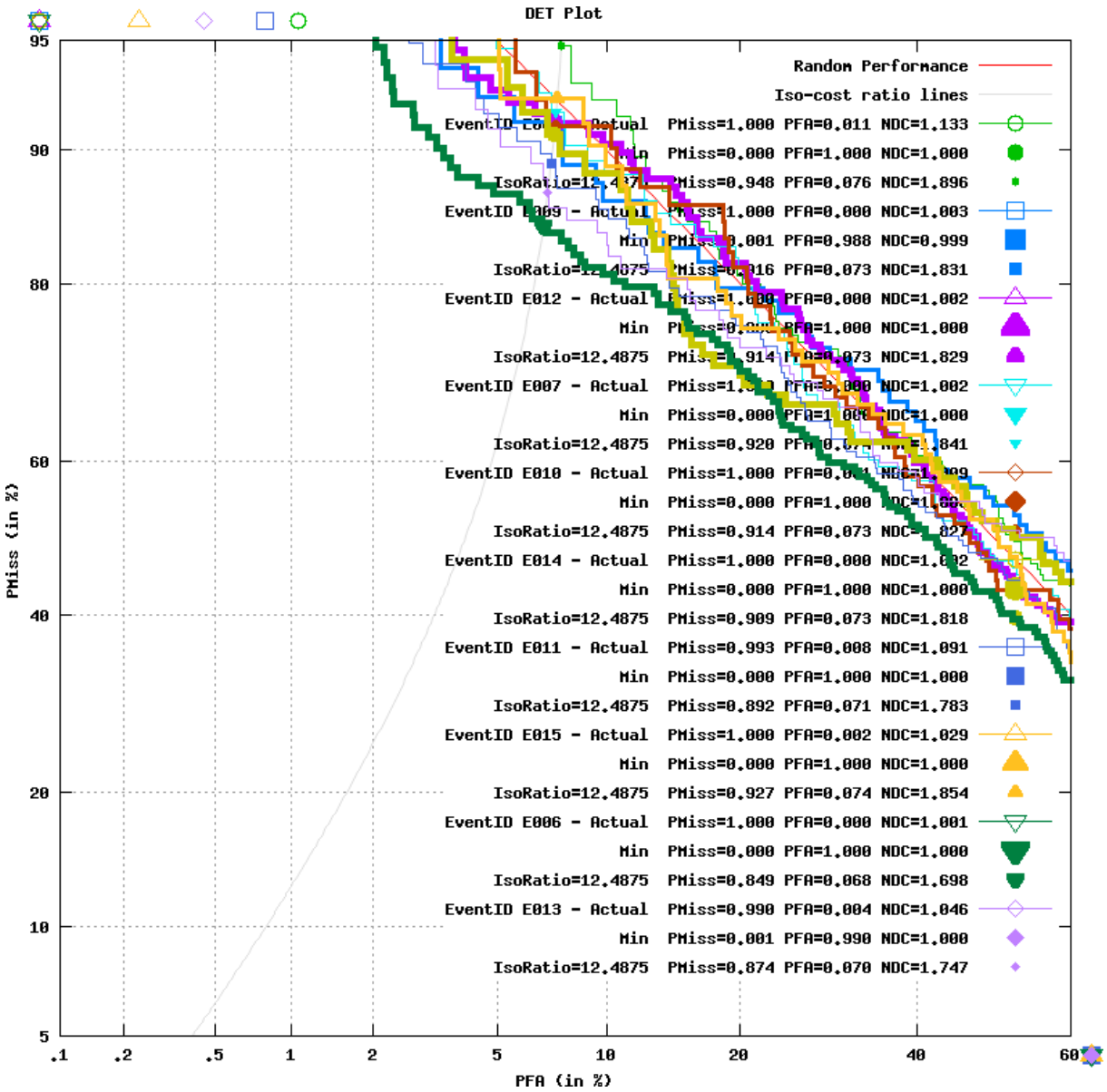

Fig. 3: The result of primary output $n = 8$ with evaluation data by NIST.

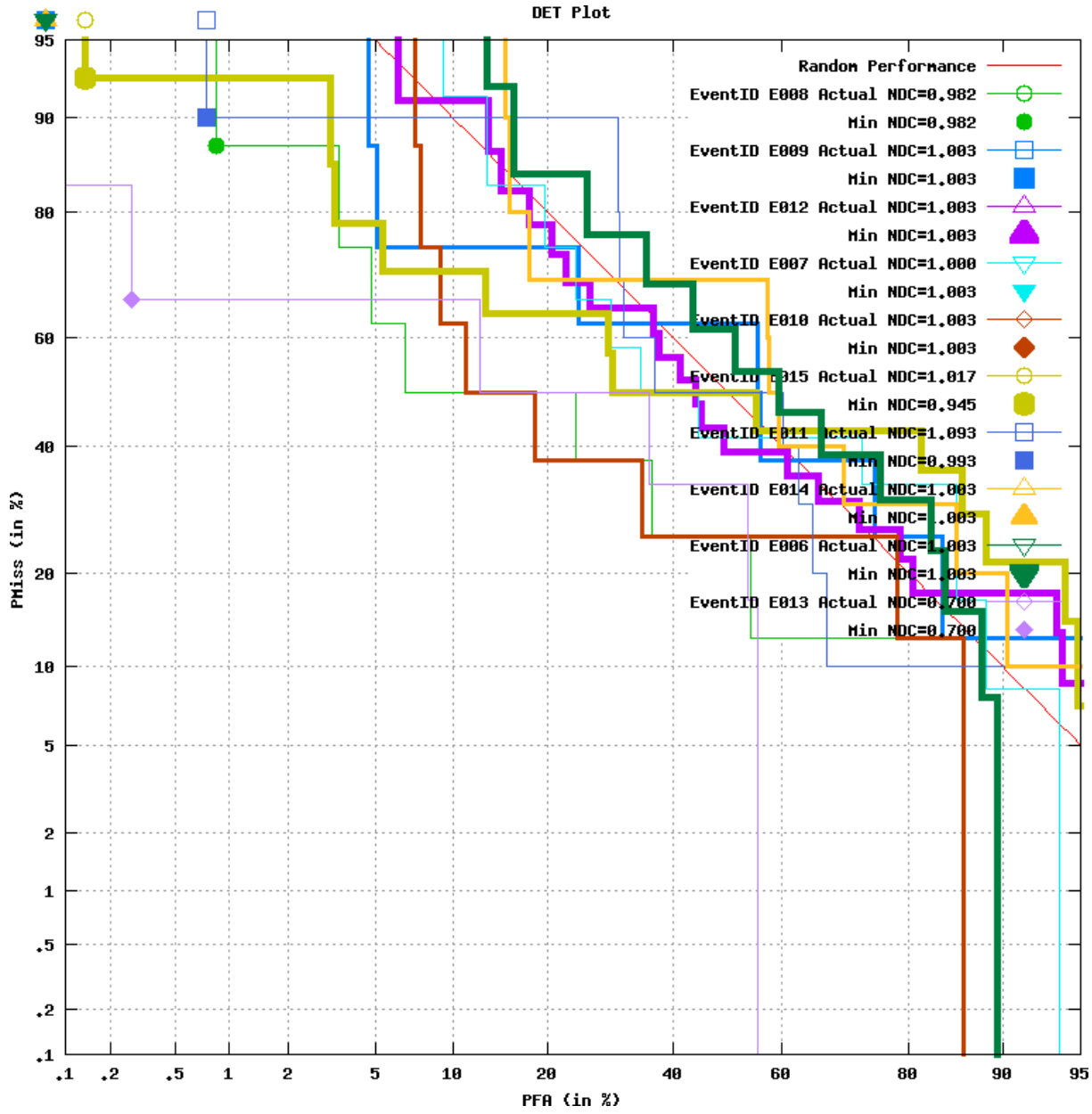

Fig. 4: Our experimental result when $n = 8$.