論文 / 著書情報
Article / Book Information

Title	TokyoTechCanon at TRECVID 2012
Authors	Nakamasa Inoue, Yusuke Kamishima, Kotaro Mori, Koichi Shinoda
Citation	TRECVID 2012, , ,
Pub. date	2012, 11

TokyoTechCanon at TRECVID 2012

Nakamasa Inoue Yusuke Kamishima Kotaro Mori Koichi Shinoda
Department of Computer Science, Tokyo Institute of Technology
{inoue,kamishi,mori}@ks.cs.titech.ac.jp {shinoda}@cs.titech.ac.jp

1 Semantic Indexing

We aim at developing a high-performance semantic indexing system using Gaussian-mixture-model (GMM) supervectors and tree-structured GMMs [1, 2, 3]. GMM supervectors corresponding to six types of audio and visual features are extracted from video shots. Tree-structured GMMs reduce the computational cost of maximum a posteriori (MAP) adaptation for estimating GMM parameters while keeping accuracy at high levels. This year, we introduce two new low-level features of HOG-Dense and LBP-Dense and video-clip scores. HOG-Dense and LBP-Dense are extracted from up to 100 frames per shot by using dense sampling. The video-clip score is defined as the maximum value of shot scores among all the shots in a video clip and is used for re-ranking video shots. Our best result was 32.10% in terms of Mean InfAP, which was ranked first over all semantic indexing runs in the full task.

1.1 Low-Level Feature Extraction

The following six types of visual and audio features are extracted from video data. This year, two new features are introduced: HOG-Dense (multi-frame) and LBP-Dense (multi-frame).

1. SIFT features with Harris-Affine detector (SIFT-Har)

   Scale Invariant Feature Transform (SIFT) proposed by Lowe [4] is a local feature extraction method that is widely used for object categorization since it is invariant to image scaling and changing illumination. The Harris-Affine detector [5], which is an extension of the Harris corner detector, improves the robustness against affine transform of local regions. SIFT features are extracted from every other frame, and principal component analysis (PCA) is applied to reduce their dimensions from 128 to 32.

2. SIFT features with Hessian-Affine detector (SIFT-Hes)

   SIFT features are extracted with the Hessian-Affine detector [5], which is complementary to the Harris-Affine detector. The combination of several different detectors can improve the robustness against noise. SIFT features are extracted from every other frame, and PCA is applied to reduce their dimensions from 128 to 32.

3. SIFT and hue histogram with dense sampling (SIFTH-Dense)

   SIFT features and 36-dimensional hue histograms [6] are combined to capture color information. SIFT+Hue features are extracted from key-frames by using dense sampling (100x100 grid with 3 scales). PCA is applied to reduce dimensions from 164 to 32.

4. HOG with dense sampling (HOG-Dense)

   32-dimensional histogram of oriented gradients (HOG) are extracted from up to 100 frames per shot by using dense sampling with 2x2 blocks. PCA is applied but dimensions of the HOG features are kept to 32.

5. LBP with dense sampling (LBP-Dense)

   Local Binary Patterns (LBPs) [7] are extracted from up to 100 frames per shot by using dense sampling with 2x2 blocks to capture texture information. We follow the procedure in [7] to extract LBP features. PCA is applied to reduce dimensions from 228 to 32.

6. MFCC audio features (MFCC)

   Mel-frequency cepstral coefficients (MFCCs), which describe the short-time spectral shape of audio frames, are extracted to capture audio information. MFCCs are widely used not only for speech recognition but also for generic audio classification. Δ MFCCs, $\Delta\Delta$ MFCCs, Δ log-power and $\Delta\Delta$ log-power are extracted in addition to the MFCCs. Here, "Δ" means the derivation of the feature. The dimension of the audio feature is 38, including 12-dimensional MFCCs.

1.2 Gaussian Mixture Models

Gaussian mixture models (GMMs), whose probability density function (pdf) is given by

$$p(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{1}$$

are used to model video shots. Here, x is a local feature, $\theta = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is a set of GMM parameters, K is the number of Gaussian components (vocabulary size), w_k is a mixture coefficient, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a Gaussian pdf with a mean vector μ_k and a covariance matrix Σ_k.

The GMM parameters are estimated for each shot under the maximum a posteriori (MAP) criterion. The MAP solution for GMM means, namely MAP adaptation, is given by

$$\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(\mathrm{U})} + \sum_{i=1}^{n} c_{ik} x_i}{\tau + C_k}, \quad c_{ik} = \frac{w_k g_k(x_i)}{\sum_{k=1}^{K} w_k g_k(x_i)}, \quad C_k = \sum_{i=1}^{n} c_{ik}, \quad g_k(x) = \mathcal{N}(x|\hat{\mu}_k^{(\mathrm{U})}, \hat{\Sigma}_k^{(\mathrm{U})}), \tag{2}$$

where $X_{\mathrm{F}} = \{x_i\}_{i=1}^{n}$ ($\mathrm{F} \in \{\text{SIFT-Har, SIFT-Hes, SIFTH-Dense, HOG-Dense, LBP-Dense, MFCC}\}$) is a set of feature vectors extracted from a shot, τ is a predefined hyper-parameter, and $\hat{\theta}^{(\mathrm{U})}$ is the parameter for a universal background model (UBM). The UBM presents how the features are distributed in the general case: therefore, the parameter $\hat{\theta}^{(\mathrm{U})}$ is estimated by using all features in the training set.

1.3 Fast MAP Adaptation

The fast MAP adaptation technique [1, 2] reduces computational costs for calculating posterior probabilities c_{ik} in Eq. (9) by constructing a tree-structed GMM. The basic idea of the tree-structured GMM is to cluster Gaussian components and approximate them with a single Gaussian. Each leaf node corresponds to a Gaussian component of the UBM, and each non-leaf node has a single Gaussian that approximates its descendant Gaussian components.

The tree-structured GMM $\mathcal{T}$ is defined as

$$\mathcal{T} = (V, E, G_{\mathrm{TREE}}), \tag{3}$$

where V is a set of nodes, E is a set of edges, and G_{TREE} is a set of Gaussian components for nodes in V. Here, $g^{(v)} \in G_{\mathrm{TREE}}$ denotes a Gaussian component for $v \in V$. The Gaussian components for the UBM, $G_{\mathrm{UBM}} = \{g_k\}_{k=1}^{K}$, are assigned to leaf nodes, i.e., for each leaf $\ell \in V$, there exists $g_k \in G_{\mathrm{UBM}}$ that satisfies $g^{(\ell)} = g_k$. Gaussian components for non-leaf nodes and their children are determined by applying k-means clustering to G_{UBM} (see [1, 2] for details).

With the tree-structured GMM, posterior probabilities c_{ik} in Eq. (9) are calculated only for *active* Gaussian components. The following algorithm finds active leafs by expanding active nodes V_A from the root node to output c_{ik} quickly.

1. Set $V_A \leftarrow \{r\}$, where r is the root node.
2. Expand active nodes by making child nodes of the active nodes active:

$$V_A \leftarrow \bigcup_{v \in V_A} C(v), \tag{4}$$

   where $C(v)$ is a set of child nodes of the node v. Here, $C(\ell) = \{\ell\}$ is used for leaf nodes ℓ to keep the leaf nodes active.
3. Calculate posterior probabilities $c_i^{(v)}$ for an active GMM given by

$$p(x|V_A) = \sum_{v \in V_A} \tilde{w}^{(v)} g^{(v)}(x), \quad \tilde{w}^{(v)} = \frac{w^{(v)}}{\sum_{v \in V_A} w^{(v)}}, \tag{5}$$

   i.e., calculate

$$c_i^{(v)} = \frac{\tilde{w}^{(v)} g^{(v)}(x_i)}{\sum_{v \in V_A} \tilde{w}^{(v)} g^{(v)}(x_i)} = \frac{w^{(v)} g^{(v)}(x_i)}{\sum_{v \in V_A} w^{(v)} g^{(v)}(x_i)}. \tag{6}$$

4. Keep a node v active if $c_i^{(v)}$ is larger than the predetermined threshold c_{TH}, i.e.

$$V_A \leftarrow \{v \in V_A \mid c_i^{(v)} > c_{\text{TH}}\}. \tag{7}$$

5. If all nodes in V_A are leaf nodes, output

$$\hat{c}_{ik} = \begin{cases} c_i^{(\ell)} & (\ell \in V_A,\ g^{(\ell)} = g_k) \\ 0 & (\text{otherwise}) \end{cases}. \tag{8}$$

Otherwise, return to Step 2.

Finally, MAP adaptation is given by

$$\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(\text{U})} + \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik} x_i}{\tau + \hat{C}_k}, \quad \hat{C}_k = \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik}. \tag{9}$$

1.4 GMM Supervector SVM

After video shots are represented by GMMs, GMM supervectors are extracted by combining normalized mean vectors as

$$\phi(X_{\text{F}}) = \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \\ \vdots \\ \tilde{\mu}_K \end{pmatrix}, \quad \tilde{\mu}_k = \sqrt{w_k^{(U)}} (\Sigma_k^{(U)})^{-\frac{1}{2}} \hat{\mu}_k. \tag{10}$$

Support vector machines (SVMs) with the following RBF-kernel are used to train discriminative models for each semantic concepts.

$$k(X_{\text{F}}, X'_{\text{F}}) = \exp\left(-\gamma \|\phi(X_{\text{F}}) - \phi(X'_{\text{F}})\|_2^2\right), \quad \gamma = \frac{1}{\tilde{d}}, \tag{11}$$

where $\tilde{d}$ is the average distance between two GMM supervectors. Finally, trained discriminative functions are linearly combined as

$$f(X) = \sum_{\text{F} \in \mathcal{F}} \alpha_{\text{F}} f_{\text{F}}(X_{\text{F}}), \quad 0 \leq \alpha_{\text{F}} \leq 1, \quad \sum_{\text{F}} \alpha_{\text{F}} = 1. \tag{12}$$

where $\mathcal{F} = \{\text{SIFT-Har}, \text{SIFT-Hes}, \text{SIFTH-Dense}, \text{HOG-Dense}, \text{LBP-Dense}, \text{MFCC}\}$. Combination coefficients α_{F} are optimized on a validation set.

1.5 Video-Clip Scores

The relationship between shots are useful for detecting semantic concepts. For example, Safadi et al. [8] proposes a re-ranking method to re-evaluate scores of video shots by using shot-score distributions. In our re-ranking method, we define a video-clip score as the maximum value of shot scores among all the shots in a video clip:

$$s_{\max} = \max_i s_i \tag{13}$$

where $s_i (i = 1, 2, \cdots, n)$ are shot scores for a video-clip that consists of n shots. Our final score for ranking shots is given by

$$s'_i = (1 - p) s_i + p s_{\max} \tag{14}$$

where p is a probability of appearance of a semantic concept in a video clip given by

$$p = \left\langle \frac{\#(\text{positive shots in a video clip})}{\#(\text{shots in a video clip})} \right\rangle. \tag{15}$$

The final score s'_i gets closer to $s_{\max}$ as the concept appear more often (e.g. an anchorperson in a news video). It gets closer to the original shot score s_i for concepts appear in few times (e.g. a bus in a street video).

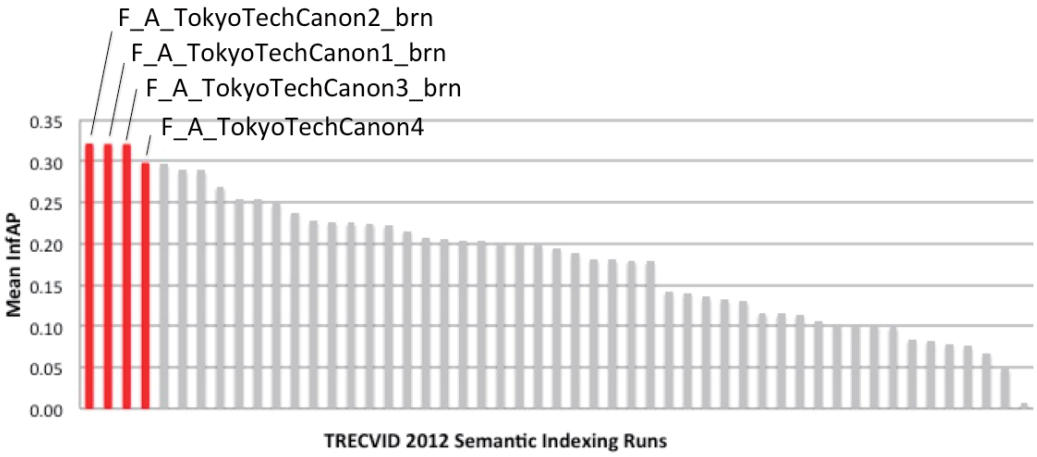

Figure 1: Overview of results of the semantic indexing task in TRECIVD 2012. Our best result was Mean InfAP of 32.10%.

1.6 Experimental Conditions

Mikolajczyk's implementation [5] was used to extract SIFT-Har and SIFT-Hes features. SIFT++ [9] was used to extract SIFTH-Dense features. HTK [10] (speech recognition toolkit) was used to extract MFCC. The sum of calculation time (for PCA projection, MAP adaptation, and SVM prediction) was reduced from 2.47 sec to 1.00 sec (59.5%) by using the fast MAP adaptation technique. The calculation time was measured by using a Intel Xeon 2.93 GHz CPU.

The following four runs are submitted to the TRECVID 2012 semantic indexing.

F_A_TokyoTech_Canon_4 (baseline)

This run used GMM supervector SVMs with the six types of features described in Section 1.1. The number of Gaussian components was $K = 512$ for the visual features, and $K = 256$ for the audio feature. The UBMs were trained using 1,000,000 samples. Optimal tree structures for the UBMs were selected as to minimize computational costs for MAP adaptation (see [1, 2]). The hyper-parameter τ for MAP adaptation was set to 20.0. Weights α_{F} for fusion are optimized on IACC_1_B dataset.

F_A_TokyoTech_Canon_3_brn

This run used the same features and parameters as F_A_TokyoTech_Canon_4 with additonal annotations provided by a team of Brno University [11].

F_A_TokyoTech_Canon_2_brn

This run is the same as F_A_TokyoTech_Canon_3_brn but weights α_{F} for fusion are optimized on IACC_1_A and IACC_1_B dataset.

F_A_TokyoTech_Canon_1_brn

This run used PCA for GMM-supervectors, which reduce thier dimension to 400, in addition to F_A_TokyoTech_Canon_3_brn.

1.7 Results

Figure 1 shows the overview of results of the semantic indexing task. Our best result by the run of F_A_TokyoTech_Canon_2_brn was 32.10 % in terms of Mean InfAP, which is ranked first among 51 runs for the full submissions. The runs of F_A_TokyoTech_Canon_1_brn and F_A_TokyoTech_Canon_3_brn, which achieved Mean InfAPs of 32.06% and 32.05% respectively, performed very similar to F_A_TokyoTech_Canon_2_brn as shown in Figure 2. The run of F_A_TokyoTech_Canon_4 achieved Mean InfAP of 29.78% but we found a bug in the result of "Male_Person" for this run. The bug fixed version of this run achieved Mean

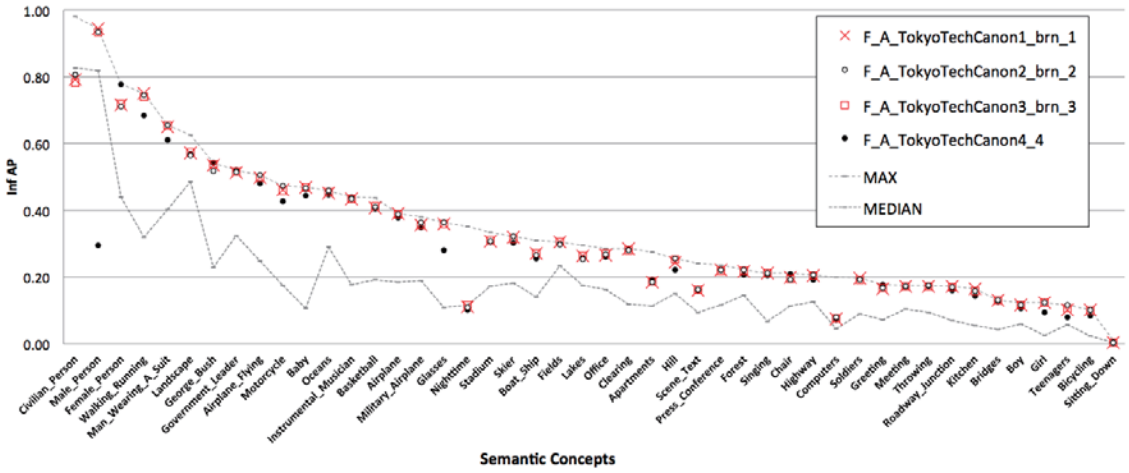

Figure 2: InfAP by semantic concept.

InfAP of 31.17%. This means that the additional annotations provided by a team of Brno University [11] improved the performance by 0.88% (from 31.17% to 32.05%).

1.8 Conclusion

We proposed a high-performance semantic indexing system using Gaussian mixture model (GMM) supervectors with the six audio and visual features. Our best result was 32.10 % in terms of Mean InfAP, which was ranked first over all semantic indexing runs in the full task. Our future work will focus on the localization of semantic concepts and the temporal analysis for video indexing.

2 Multimedia Event Detection

In this section, we will present the details of our system in Multimedia Event Detection (MED) task.

2.1 Introduction

As in the previous year 2011, we applied the combination of GMM supervectors and SVMs in our Semantic Indexing method to this MED. This year, we tried to improve the detection performance by introducing the camera motion canceled features to get the true foreground motions. We also introduced two other low-level features, spatial pyramids, and semantic score vectors as high-level features.

We submitted four runs, all of which were applied to all the testing events. In terms of mean Actual NDC, the best result of our runs ranked 7th of the 49 submissions, 3rd among the 17 teams in Pre-Specified task, and 14th of the 18 submissions, 10th among the 13 teams in Ad Hoc task.

2.2 Features

We introduce camera motion canceled features (CC-DSTIP), to get the true motions of foreground objects. We also use the five low-level features (SIFT-Har, SIFT-Hes, MFCC, STIP, and HOG) used in the previous year, new two low-level features (SURF and RGB-SIFT), spatial pyramids for low-level features, and a high-level feature; semantic score vector (HOG-SCV).

2.2.1 Low-level features

1. SIFT features with Harris-Affine detector (SIFT-Har)

   Scale-Invariant Feature Transform (SIFT) [4] has been effective in many researches of image analysis and video analysis including our semantic indexing method [2, 3]. We extract SIFT features, 128 dimensional vectors with key point detector; Harris-Affine detector[5]. The detected regions are robust for affine transformation. They are often used for corner detection. Since extracting SIFT features from all the clips and all of the frames is computationally too expensive, we use one frame in every two seconds for extraction. After extraction, we apply PCA (Principal Components Analysis) to reduce the dimension for saving computational costs in training and detection step. As a result, we compute 32-dimensional vectors.

2. SIFT features with Hessian-Affine detector (SIFT-Hes)

   In addition to Harris-Affine detector, we also use Hessian-Affine detector [5] for extraction of SIFT. Hessian-Affine detector is often used for blob detection and extracted SIFT features are expected to be complement to SIFT-Har. The frame rate for extraction and PCA dimensions are same as SIFT-Har.

3. MFCC features (MFCC)

   Since audio is one of the important clues to analyze the video contents in this MED task, we use MFCC (Mel Frequency Cepstral Coefficient) features, which is often used in speech recognition. We also use ΔMFCC, $\Delta\Delta$MFCC, Δpower, and $\Delta\Delta$power in addition to MFCC. The dimensions are 38 and PCA is applied keeping the dimension.

4. HOG features with dense sampling (HOG)

   We also use 32-dimensional histogram of oriented gradients (HOG) [16] features sampled densely from image space. Differently from features based on keypoints such as SIFT-Har and SIFT-Hes, dense sampling can give us a fixed number of features although they may include some noise. In the same way as SIFT, we sample the features from one frame every two seconds. PCA is applied keeping the dimension.

5. STIP features with Harris 3D detector (STIP)

   Motion is also one of the important clues for event detection. We use space-time interest points (STIP) [17] to get motion information. STIP From the regions detected as STIP [17], which have the significant spatial-temporal changes, we extract 72-dimensional HOG features and 90-dimensional histograms of optical flow (HOF) features, and then combine these two vectors. The 162-dimensional vectors are converted into 64-dimensional vectors by PCA.

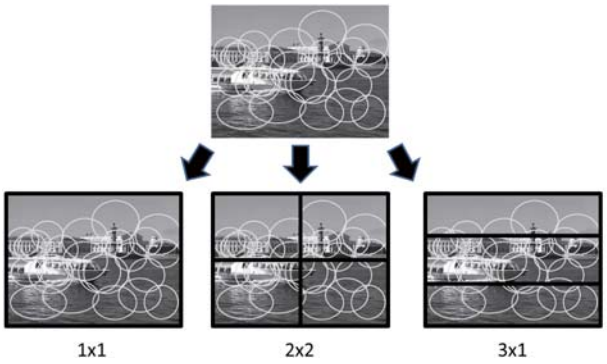

Figure 3: Spatial pyramids.

6. Camera motion canceled STIP features with dense sampling (CC-DSTIP)

   Although motion analysis is necessary for event detection, many clips are with camera motions, ant thus the motions of the foreground objects in such clips are different from their true motions. We try to remove camera motions from video clips and get true motions of the foreground objects. We estimate camera motion for each frame by estimating optical flows. Since the center region of a frame tends to include foreground objects, we only use the optical flows in the peripheral region. We move each frame image to the same direction with the same length as the camera motion. We then extract dense STIP features (CC-DSTIP), convert 162-dimensional vectors into 64-dimensional vectors by applying PCA, and construct the GMM supervectors from the feature vectors. In preliminary experiments, this CC-DSTIP outperformed dense STIP features without camera cancellation (DSTIP) (See 2.4.).

7. SURF features (SURF)

   Speeded Up Robust Features (SURF) [18], which are several times faster to extract than SIFT, are also used in our system. Feature vectors are extracted using sums of 2D Haar Wavelet response. In the same way as SIFT, we use one frame every two seconds and reduce the dimensions from 64 to 32 by using PCA.

8. RGB-SIFT features with dense sampling (RGB-SIFT)

   Since color information is often helpful for video analysis, we introduce color information to our system by using RGB-SIFT features. RGB-SIFT features are the concatenated SIFT features extracted from each of RGB channels of a image. Accordingly, the dimensions are 384 ($= 128 \times 3$). We extract RGB-SIFT features from one frame every six seconds and reduce the dimensions to 64 by using PCA. We sample the features per every six pixels with different two scales.

We introduce spatial pyramids [19] for SIFT-Har, SIFT-Hes, HOG, SURF and RGB-SIFT. This technique enables us to use the location information of feature vectors. We apply the spatial pyramid technique with three level (1x1, 2x2 and 3x1) like Figure 3. We construct a GMM supervector for each of eight regions surrounded by black lines and concatenate the GMM supervectors for all of the regions into one vector.

2.2.2 Semantic score vectors of SIN concepts using HOG features (HOG-SCV)

To detect an event, information about concepts related to it would be useful. We try to make concepts useful for event detection by using a semantic score vector [20]. A semantic score vector consists of the SVM prediction scores for the 346 concepts in Semantic Indexing task [21]. We make 346-dimensional semantic score vectors as the input to an SVM for each event, and fuse its SVM score with those of GS-SVMs (2.3) for other features.

2.3 Event detection using GMM supervectors and SVMs (GS-SVM)

We apply the combination of GMM supervectors and SVMs (GS-SVM), which is used in our Semantic Indexing method [2, 21], to this MED [22]. We construct a GMM supervector per one low-level feature

Run ID	Task	System ID	Mean ANDC	Mean Pfa	Mean Pmiss
Run 1	Pre-Specified	p-GSSVM7PyramidCcScv-r1	0.5328	0.0143	0.3537
Run 2	Pre-Specified	c-GSSVM7PyramidCc-r2	0.5296	0.0145	0.3489
Run 3	Pre-Specified	c-GSSVM7Pyramid-r3	0.5344	0.0144	0.3554
Run 4	Pre-Specified	c-GSSVM5-r4	0.5502	0.0135	0.3813
Run 5	Ad Hoc	p-GSSVM7PyramidCcScv-r5	1.7490	0.1204	0.2448
Run 6	Ad Hoc	c-GSSVM5-r6	2.5351	0.1665	0.4556

Table 1: Our runs for TRECVID2012 MED task.

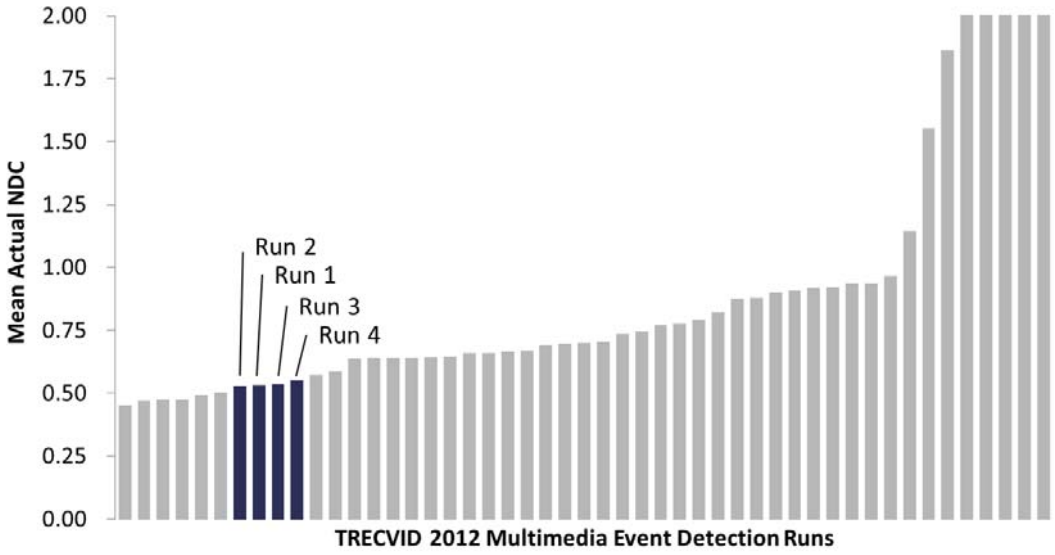

Figure 4: The overview of TRECVID 2012 MED Pre-Specified task.

(2.2.1) for each clip. Then, for each type of features, we model the distribution of the feature vectors extracted from a video clip by using a Gaussian mixture model (GMM) and construct a GMM supervector from the GMM parameters. From GMM supervectors, we train event models using support vector machines (SVMs) with RBF-kernel. Finally, we fuse multiple features by using the weighted average of their SVM scores and convert them into $[0, 1]$ domain with sigmoid fitting to get the detection scores.

2.4 Runs and results in Pre-Specified task

We submitted the four runs for Pre-Specified task. Table 1 shows the Run ID, task, system ID, mean Actual NDC, mean false alarm rate (Pfa), and mean missed detection rate (Pmiss) for each run . In all the runs for Pre-Specified task, the detection thresholds and the fusion weights were determined by 2-fold cross validation. The number of Gaussians in GMM for each feature was determined from our past experiments.

Figure 4 shows the overview of TRECVID 2012 MED Pre-Specified task. In terms of mean Actual NDC, our best run (Run 2) is ranked 7th of the 49 submissions and 3rd among the 17 teams. The following is the results and analysis for each run.

Run 4

The method used in Run 4 was the same as in our best run in last year. We used five types of features; SIFT-Har, SIFT-Hes, MFCC, HOG, and STIP without spatial pyramids. It ranked 10th in all the submissions. We reconfirmed the effectiveness of the GMM supervectors and SVMs with the five types of features.

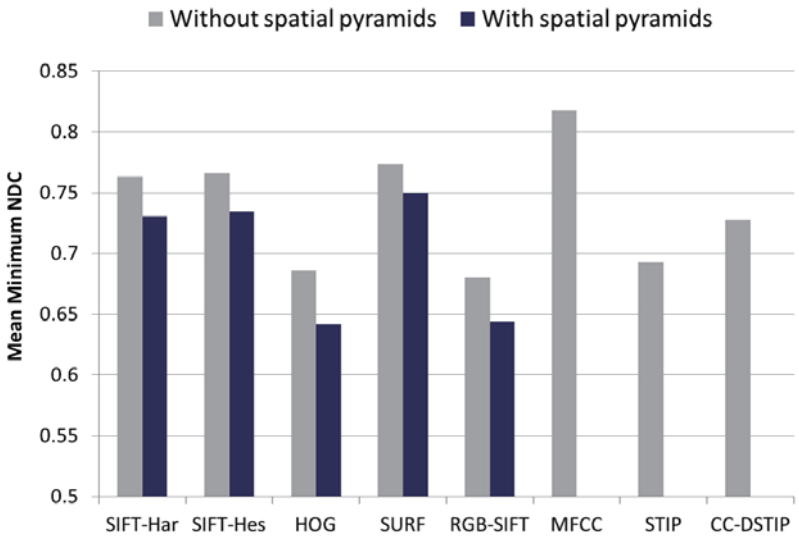

Figure 5: The mean minimum NDC for each single feature in our 2-fold cross validation using MED 2012 training data.

Feature	Mean MNDC
STIP	0.6771
CC-STIP	0.6936
DSTIP	0.7064
STIP+CC-DSTIP	**0.6346**
STIP+DSTIP	0.6634
CC-DSTIP+DSTIP	0.6694
STIP+CC-DSTIP+STIP	0.6362

Table 2: The mean Minimum NDCs of each and the combinations of the three types of motion features in MED11 dataset.

Run 3 : RUN4 + SURF + RGBSIFT + spatial pyramids

In Run 3, we added SURF, RGB-SIFT, and spatial pyramids to Run 2. Figure 5 shows the performance of each single low-level feature in our 2-fold cross validation using MED12 training dataset. We can see that RGB-SIFT, which we introduced this year, performed best in all of the single features. Figure 5 also shows that the performance of all the image features were improved by using spatial pyramids. In particular, the dense features were more improved (HOG : 6.5%, RGB-SIFT : 5.2%), compared to the features with sparse sampling (SIFT-Har : 4.3%, SIFT-Hes : 4.1%, SURF : 3.0%). It may be because in dense sampling, every regions in spatial pyramids include the uniform numbers of features.

Run 2 : RUN3 + CC-DSTIP

Run 2 is the combination of Run 3 and CC-DSTIP. We had the improvement of 0.0048 in terms of mean Actual NDC when combining CC-DSTIP with other seven low-level features. The more detailed evaluation results of CC-DSTIP in our preliminary experiments using MED11 dataset are on Table 2. It shows the effectiveness of each and the combinations of the three types of motion features; STIP, CC-DSTIP, and DSTIP. The combination of STIP and CC-DSTIP outperformed others, even the combination of all of three. It means CC-DSTIP provided us the complementary information to STIP and the information of DSTIP was slight complementary to STIP. This difference should be the effectiveness of camera motion cancellation.

Run 1 : Run 2 + HOG-SCV

Run 1 is our primary run, which is the combination of Run 2 and HOG-SCV. From the comparison of Run 1 and Run 2, HOG-SCV was not effective in this year's testing. One cause may be the smallness of

the training data. We have the 346 concepts in SIN task. However, the dimensions of score vectors might be too small for the detection in the large dataset. We may need more concepts, more data useful for event detection. It is also expected to use MED dataset to model the semantic concepts since different dataset have different characteristics such as video length, amount, and type (e.g. home videos and TV shows).

2.5 Runs and results in Ad Hoc task

The IDs and Actual NDCs of the two runs for Ad Hoc task are on Table 1. In Run 5 and Run 6, we used the same features in Run 1 and Run 4 for Pre-Specified task for each. However, we didn't optimize the detection threshold and the fusion weights for each event in the two runs for Ad Hoc task. As the detection thresholds for Ad Hoc events, we used the average of the detection thresholds for 20 Pre-Specified events. The fusion weights were also determined by the same way.

From the comparison to the results in Pre-Specified task, our detection results in Ad Hoc task were not derived appropriately. The conceivable causes are : (1) the detection thresholds (2) the fusion weights (3) bugs, human errors, or computer troubles. Since our preliminary experiments showed the averaged detection thresholds and fusion weights worked effectively. The mean NDC was 0.321 and just 9.5 % higher than that with the optimal thresholds and fusion weights. Thus, the unexpected results in Ad Hoc task might be possibly due to (3) bugs, human errors, or computer troubles.

2.6 Conclusion

In this year's MED task, we used event detection method using GMM supervectors and SVMs as in the previous year, and newly introduced camera motion cancellation to get the true motions of foregrounds, two low-level features, and spatial pyramids. They showed the effectiveness for MED Pre-Specified task. Although we also tried to make concepts useful for event detection by using semantic score vector, it wasn't effective in this year's testing. We may need more concepts, more data for this method. Our future work will be extraction of effective high-level features for event detection and improvement of the fusion method of multiple features.

References

[1] N. Inoue, and K. Shinoda. A Fast and Accurate Video Semantic-Indexing System Using Fast MAP Adaptation and GMM Supervectors. In IEEE trans. on Multimedia, vol.14, no.4, pages 1196–1205, 2012.

[2] N. Inoue, and K. Shinoda. A Fast MAP Adaptation Technique for GMM-supervector-based Video Semantic Indexing Systems. In Proc. of *ACM Multimedia* (short paper), 2011.

[3] N. Inoue, and et al. High-Level Feature Extraction using SIFT GMMs and Audio Models. In Proc. of *ICPR*, 2010.

[4] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *In IJCV*, 2004.

[5] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. In *IJCV*, 60(1):63–86, 2004.

[6] J. van de Weijer and C. Schmid. Coloring local feature extraction. In Proc. of *ECCV*, vol.2, pages 334–348, 2006.

[7] X. Wang, T. X. Han, and S.Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In Proc. of *ICCV*, pages 32–39, 2009.

[8] B. Safadi and G. Qunot. Re-ranking by Local Re-scoring for Video Indexing and Retrieval. In Proc. of *CIKM*, 2011.

[9] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, http://www.vlfeat.org/, 2008,

[10] S. J. Young, G. Evermann, M. J. F. Gales, D. Kershaw, G. Moore, J. J. Odell, D. G. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. The htk book, version 3.4, 2006.

[11] M. Hradis, M. Kolar, J. Kral, A. Lanik, P. Zemcik, and P. Smrz. Annotating Images with Suggestions - User Study of a Tagging System. In Proc. of *ACIVS*, 2012.

[12] T.Ojala, M.Pietikainen and D.Harwood. Acomparative study of texture measures with classification based on feature distribution. *Pattern Recognition*, vol. 29, no. 1, pp. 51-59 1996.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, vol. 1, 2001.

[14] M. Isard and A. Blake. Condensation: Unifying low-level and high-level tracking in stochastic framework. *Proc. ECCV*, 1998

[15] B. K. P. Horn and B. G. Schunck, Determining optical flow. *Artifical Intelligence*, vol. 17, no. 1, pp. 185-203, 1981.

[16] N. Dalal and B. Triggs.B. Histograms of oriented gradients for human detection. In Proc. of *CVPR*, 2005.

[17] I. Laptev. On space-time interest points. In *IJCV*, vol.64(2), pp.107-127, 2005.

[18] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, Surf: Speeded up robust features. *CVIU*, vol. 110, no. 3, pp. 346-359, 2008.

[19] S. Lazebnik, C. Schmid and J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In Proc. of *CVPR*, 2006.

[20] L. Zhang, L. Jiang, Y. Li, L. Bao and S. Takahashi, Informedia@TRECVID 2011: Surveillance Event Detection. TREC Video Retrieval Evaluation (TRECVID) workshop, Dec.5, 2011.

[21] N. Inoue, Y. Kamishima, T. Wada, K. Shinoda and S. Sato, TokyoTech+Canon at TRECVID 2011. TREC Video Retrieval Evaluation (TRECVID) workshop, Dec.5, 2011.

[22] Y. Kamishima, N. Inoue, K. Shinoda and S. Sato, MULTIMEDIA EVENT DETECTION USING GMM SUPERVECTORS AND SVMS. In Proc. of *ICIP*, 2012.

[23] C. Snoek , M. Worring , and A. Smeulders, Early versus late fusion in semantic video analysis. In Proc. of *ACM Multimedia*, 2005.