MSRA AT TRECVID 2008:
HIGH-LEVEL FEATURE EXTRACTION AND AUTOMATIC SEARCH

Tao Mei, Zheng-Jun Zha, Yuan Liu, Meng Wang
Guo-Jun Qi, Xinmei Tian, Jingdong Wang, Linjun Yang, Xian-Sheng Hua

Microsoft Research Asia
{tmei, xshua}@microsoft.com

ABSTRACT

This paper describes the MSRA experiments for TRECVID 2008. We performed the experiments in high-level feature extraction and automatic search tasks. For high-level feature extraction, we representatively investigated the benefit of global and local low-level features by a variety of learning-based methods, including supervised and semi-supervised learning algorithms. For automatic search, we focused on text and visual baseline, query-independent learning, and various reranking methods.

***Index Terms*—** support vector machines, bag of words, semi-supervised learning, optimal multi-graph learning, transductive multi-label learning, video annotation, video search, query-independent learning, search reranking.

1. INTRODUCTION

MSRA took part in two tasks and submitted six runs for high-level feature extraction and five runs for automatic search task in TRECVID 2008.

In high-level feature extraction, we focused on exploring the benefit of both the global and local low-level features. We leveraged nine types of global feature and two kinds of local feature. We also studied the effectiveness of various learning-based methods, such as semi-supervised learning methods: optimal multi-graph semi-supervised learning (OMGSSL) [1] and transductive multi-label learning (TML) [2], and supervised learning methods: bag of words (BoW) [3] and the baseline method support vector machine (SVM) [4]. As a result, there are four classifiers adopted in high-level extraction. For each classifier, we trained different models on a variety of low-level visual modalities (i.e., global and local visual modalities), as well as different data splitting manners. In total, we trained 58 different models. Then, based on different fusion strategies (i.e., linear, max, and average fusion) of these models, we had 10 methods. Finally, we fused the models and methods according to different fusion strategies and submitted the following six runs. The pipeline of high-level feature extraction is shown in Figure 1, including data preparation, modalities (i.e., low-level features), classifiers, models, methods, and runs.

Table 1. The performances of six runs for feature extraction.

RUN ID	MAP
A_MSRA_HLF_1	0.1103
A_MSRA_HLF_2	0.1101
A_MSRA_HLF_3	0.1089
A_MSRA_HLF_4	0.1008
A_MSRA_HLF_5	0.1021
A_MSRA_HLF_6	0.1101

- **A_MSRA_HLF_1:** linear weighted fusion of all the 10 methods and 58 models.
- **A_MSRA_HLF_2:** linear weighted fusion of all the 15 methods.
- **A_MSRA_HLF_3:** linear weighted fusion of SVM related runs.
- **A_MSRA_HLF_4:** linear weighted fusion of the top 5 methods for each concept. The principle for selecting methods is based on the performance over selection set.
- **A_MSRA_HLF_5:** linear weighted fusion of all the 58 models.
- **A_MSRA_HLF_6:** re-ranked results of A_MSRA_HLF_1 [1].

The corresponding performances of high-level feature extraction are listed in Table 1, in which we found that A_MSRA_HLF_1 achieved the best MAP among the submitted six runs.

In automatic search, we focused on text and visual baseline, query-independent learning and various reranking methods. The pipeline of automatic search is shown in Figure 2. Finally, we submitted the following five runs:

[1] We exploit GRF-based [5] reranking over the results of A_MSRA_HLF_1.

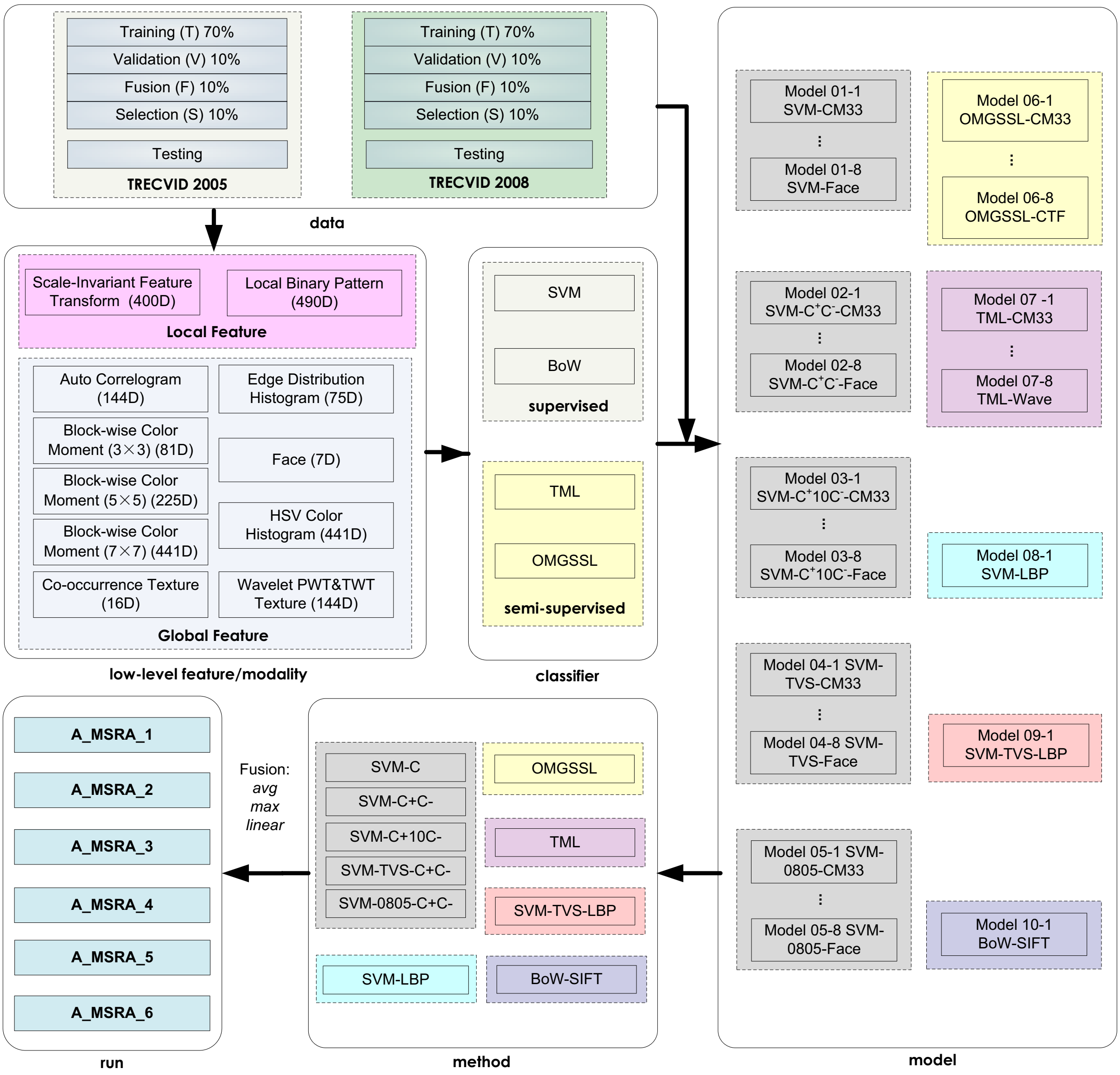


Fig. 1. The MSRA TRECVID 2008 high-level feature extraction pipeline. 2008—2008 data set, 2005—2005 data set, T—training set, V—validation set, F—fusion set, S—selection set, C—only tuning parameter C in SVM, C^+ and C^-—tuning both of two parameters for SVM.

Table 2. The performances of five runs for automatic search.

RUN ID	MAP
F_A_2_MSRA.TV8_1 (SEARCH_1)	0.0407
F_A_2_MSRA.TV8_2 (SEARCH_2)	0.0407
F_A_2_MSRA.TV8_3 (SEARCH_3)	0.0330
F_A_1_MSRA.TV8_4 (SEARCH_4)	0.0180
F_A_1_MSRA.TV8_5 (SEARCH_5)	0.0338

- **F_A_2_MSRA.TV8_1:** linear weighted fusion of text baseline and visual baseline, then temporal expansion over the fusion results.
- **F_A_2_MSRA.TV8_2:** linear weighted fusion of text baseline and visual baseline.
- **F_A_2_MSRA.TV8_3:** averaging fusion of Bayesian reranking and MIIL reranking results over the text baseline; then temporal expansion over the fusion results.
- **F_A_1_MSRA.TV8_4:** text baseline. Averaging fusion of results based on Okapi BM25 ranking function and vector space models.
- **F_A_1_MSRA.TV8_5:** visual baseline. Averaging fusion of results based on query-dependent learning and query-independent learning.

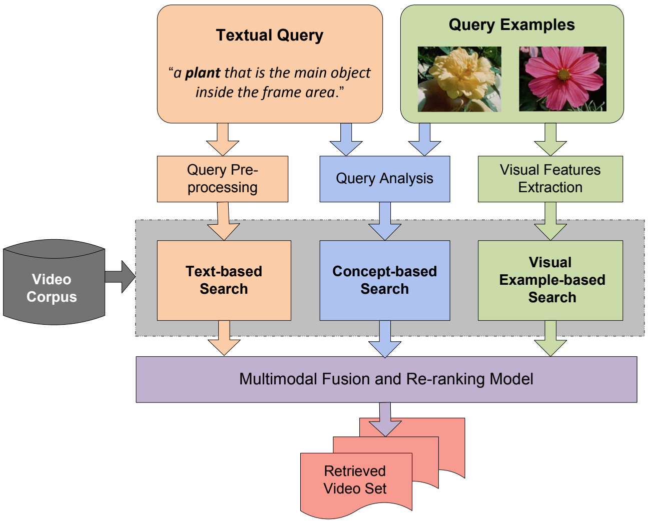

Textual Query
"a plant that is the main object inside the frame area."
Query Examples
Query Pre-processing
Query Analysis
Visual Features Extraction
Video Corpus
Text-based Search
Concept-based Search
Visual Example-based Search
Multimodal Fusion and Re-ranking Model
Retrieved Video Set

Fig. 2. MSRA_SEARCH automatic search pipeline.

The corresponding performances of automatic search are listed in Table 2.

2. HIGH-LEVEL FEATURE EXTRACTION

2.1. Modalities (Low-level features)

For each key-frame, we extracted 11 types of low-level features (referred to as "modality" in this notepaper), including nine types of global feature and two kinds of local feature.

2.1.1. Global Feature

We extracted nine types of global feature, which characterize the different properties of the key-frames, such as color, texture, and shape. Table 3 lists the detailed information of these modalities.

2.1.2. Local Feature

We extracted two kinds of local feature, including Local Binary Patterns (LBP) [7] and Scale-Invariant Feature Transform (SIFT) [8] feature. For LBP feature, we separated each key-frame to 49 non-overlapped rectangles of size around 50×40. For each rectangle, we extracted a 10-bin histogram of LBP feature. These histograms were then concatenated to form the feature vector. As a result, 490-D LBP features were obtained for each sample. For SIFT feature, local patches are extracted densely from the key-frames and translated to SIFT descriptors. We extracted the patches centering on a regular grid with spacing 10 pixels and calculated the descriptor of each local patch. As a result, each key-frame is translated to a set of local features. Since the main orientation of the sampled local patches is unknown, for each sampled patch, we used two orthogonal orientations as the main orientation and concatenated the calculated descriptors, resulting in a vector of 256 dimensions. Then the Principle Component Analysis (PCA) was applied to transform the features to 80 dimensions to reduce the computation and storage cost.

2.2. Classifiers on Global Feature

2.2.1. Support Vector Machine (SVM)

Implementation: SVM [4] was adopted as the baseline. Here we adopt the late fusion strategy. Specifically, SVM was trained in each of the eight global features described in Table 3 except for "co-occurrence texture" modality. Then we fused these eight models by linear weights. As a result, we applied the following five methods: (1) SVM-C, (2) SVM-C^+C^-, (3) SVM-C^+10C^-, (4) SVM-TVS-C^+C^-, and (5) SVM-0805-C^+C^-.

We separated both TRECVID 2005 and 2008 development set into four partitions including "Training," "Validation," "Fusion," and "Selection" set. The detailed information of data splitting, models and methods is described in Figure 8. We used RBF kernels in SVM, which have two primary parameters: C (the cost parameter in soft-margin SVMs) and γ (the width of the RBF function). The effectiveness of SVM classifiers is highly subject to the selection of model parameters. To address the unbalance problem, we set different cost parameters for positive and negative samples, respectively. Therefore, we considered three model parameters: C^+ (the cost parameter for the positive examples), C^- (the cost parameter for the negative examples), and γ. In our method, we assign the ratio $\frac{C^+}{C^-} = \frac{N^-}{N^+}$ or $\frac{C^+}{C^-} = 10 \times \frac{N^-}{N^+}$, where N^+ and N^- are the numbers of positive and negative training examples, respectively. Based on the "Validation" set shown in Figure 1, we selected the best choice of these parameters.

2.2.2. Optimizing Multi-Graph Learning (OMGSSL)

Formulation: OMGSSL is a semi-supervised method to learn from multiple graphs [1]. Suppose we have G graphs $\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_G$, the regularization framework is formulated as

$$Q(f,\alpha)=\sum_{g=1}^{G}\sum_{i,j}\alpha_g^r\left(W_{g,ij}\left|\frac{f_i}{\sqrt{D_{g,ii}}}-\frac{f_j}{\sqrt{D_{g,jj}}}\right|^2+\mu\sum_i|f_i-Y_i|^2\right)$$

$$[f,\alpha] = \arg\min{}_{f,\alpha} Q(f,\alpha), s.t. \sum_{g=1}^{G} \alpha_g = 1 \qquad (1)$$

Note that we have proposed to adopt multiple distance metrics in [1]. However, in TRECVID 2008 experiments, we have only used L_1 for simplicity. We generate M graphs from M modalities and generate a graph to indicate temporal consistency, i.e., $G = M + 1$.

Implementation: Eq. (1) can be solved in an EM-style iterative way. But in [1], we have mentioned that when l is not extremely small, we can derive an approximate solution

Table 3. Low-level feature (modalities)

Level	Feature	Dim	Description
Global	Auto-correlogram	144	36-bin color histogram based on 4 different distance k, i.e., k = 1,3,5,7.
	ColorMoment3-by-3	81	Based on 3 by 3 division of images in Lab space
	ColorMoment5-by-5	225	Based on 5 by 5 division of images in Lab space
	ColorMoment7-by-7	441	Based on 7 by 7 division of images in Lab space
	Co-occurrence Texture	16	The same feature as in [6]
	Edge Distribution Histogram	75	The same feature as in [6]
	Face	7	Face number, face area ratio, the position of the largest face
	HSV Color Histogram	64	The same feature as in [6]
	Wavelet PWT&TWT Texture	128	The same feature as in [6]
Local	Local Binary Pattern	490	The feature proposed in [7]
	Scale-Invariant Feature Transform	80	The feature proposed in [8]

which can reduce computational costs. Specifically, we first compute α_g

$$\alpha_g = \frac{\left(\frac{1}{Y^T L_g Y}\right)^{\frac{1}{r-1}}}{\sum\limits_{g=1}^{G}\left(\frac{1}{Y^T L_g Y}\right)^{\frac{1}{r-1}}} \tag{2}$$

Then we compute f

$$f = \left(\mathbf{I} + \frac{1}{\sum_{g=1}^{G} \alpha_g^r \mu_g} \frac{\sum\limits_{g=1}^{G} \alpha_g^r \mathbf{L}_g}{\sum\limits_{g=1}^{G} \alpha_g^r} \right)^{-1} Y \tag{3}$$

Analogous to that described in manifold-ranking, Eq. (3) can be solved in an iterative way as follows.

```
1: Initialize $f^{(t)}$ where $t = 0$.
2: Update $f$ by
   $f^{(t+1)} = \frac{1}{1 + \sum_{g=1}^{G} \alpha_g^r \mu_g} \Big( \mathbf{I} - \frac{\sum\limits_{g=1}^{G} \alpha_g^r \mathbf{L}_g}{\sum\limits_{g=1}^{G} \alpha_g^r} \Big) f^{(t)} + \frac{\mu}{1+\mu} Y$
3: Let $t = t + 1$, and then jump to step 2 until convergence.
```

Similar to the experiments described in [6], we constructed experiments TRECVID 2008 data set. There are four parameters should be tuned in this algorithm, including σ_g, μ_g, γ, and A. The optimal parameter configuration was selected based on "Validation" subset.

2.2.3. *Transductive Multi-Label Learning (TML)*

Formulation: TML [2] is a semi-supervised multi-label classification approach based on the discrete hidden Markov random field model, which simultaneously models the labeling consistency between the visually similar videos and the multi-label interdependence for each video in an integrated framework. It aims to find a labeling such that the multi-label interdependence over the unlabeled data points is coherent with that over the labeled ones. TML formulates the multi-label interdependence as a pairwise Markov random field model, in which all the combinations of relations, including the co-positive, co-negative, and cross-positive relations, are explored. In [2], the potential function over all the edges on the dHMRF is defined as:

$$\begin{aligned} \Pr(\mathbf{Y}) &= \tfrac{1}{Z} \Pr_0(\mathbf{Y}_l) \Pr_s^{\lambda}(\mathbf{Y}) \Pr_d^{1-\lambda}(\mathbf{Y}_u) \\ &= \tfrac{1}{Z} \textstyle\prod_{i \in L, c \in C} \Pr(y_{ic}) \prod_{c \in C, (i,j) \in \varepsilon_s^c} \varphi^{\lambda}(y_{ic}, y_{jc}) \\ &\quad \textstyle\prod_{i \in U, \alpha, \beta \in C} \phi^{1-\lambda}(y_{i\alpha}, y_{i\beta}) \end{aligned} \tag{4}$$

Such potential faction simultaneously captures the compatibility with pre-labeling, consistency over local labeling, and independence among multiple labels.

Implementation: The solution to transductive multi-label classification is found as the joint maximum,

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} \Pr(\mathbf{Y}) \tag{5}$$

The optimization of Eq. (5) is performed by combining two optimization methods for discrete hidden Markov random field: tree-reweighted message passing and graph cuts.

2.3. Classifiers on Local Feature

We learned SVM classifiers based on the LBP features according to the experiment setting described in Sectionr̃efsec:svm For SIFT feature, we adopted the Bag of Words method [3]. For each key-frame, we transformed the bag of sift features to a 400-bin histogram. These histograms were regarded as the visual features and were fed into SVM to learn the classifiers. The algorithmic parameters were tuned on the "Validation" subset.

2.4. Experimental Results

Figure 3 shows the AP performance of MSRA_HLF six runs submitted to TRECVID 2008 for each concept. Figure 4 shows the overview of top 70 high-level feature extraction runs submitted to TRECVID 2008, ranked according to MAP. The black bars correspond to the performances of MSRA_HLF.

3. AUTOMATIC SEARCH

MSRA team continued its effort on automatic search and submitted five automatic runs, including text baseline (only using the text information), visual baseline (using no text information) and three reranking or fusion runs. The automatic search system again consists of several main components, including query pre-processing and query analysis, uni-modal search, multimodal fusion and re-ranking. The framework of the search system is shown as Figure 2. By analyzing and pre-processing the query, the multimodal query (i.e., text, key-frames and shot) are input to individual search models, such as text-based and visual example-based model. Moreover, the related concepts to the given query are detected and the concept-based model can be built. Then several search re-ranking and fusion approaches are applied.

All runs were done at the shot level based on the master shot boundary reference [9]. For the text baseline (SEARCH_4), we only use the common ASR/MT without any reranking methods. For the visual baseline (SEARCH_5), we use query examples to build both query-dependent and query-independent models [10], and combine the results predicted by the two types of models in averaging way. We got the MAP score of 0.0338 and 0.0180 for the text baseline and visual baseline, respectively. After fusing the text baseline and visual baseline in weighted averaging way, we achieved the MAP score of 0.0407 (SEARCH_2). Furthermore, we tried several reranking methods over the text baseline, including the temporal expansion [6], Bayesian reranking [11] and MIIL reranking [12]. We simply averaging of these reranking results and got the MAP score of 0.0330 (SEARCH_3), which improves the text baseline 23.7%. We also apply the temporal expansion on the fusion results of text and visual baseline (SEARCH_2), the final MAP score is 0.0407 (SEARCH_1) - the same as the initial MAP score.

3.1. Ranking/ reranking for video search

This year we emphasized various video search and reranking methods, including query-dependent learning, query-independent learning [10], Bayesian reranking [11] and MIIL reranking [12].

3.1.1. Query-dependent learning

Formulation: The learning-based video search aims to use machine learning techniques to explicitly model the query semantics. The search is posed as a classification problem, targeting identifying a video shot relevant (i.e., "positive") or irrelevant (i.e., "negative") to the given query. The query-dependent learning aims to learn a model for each query. The features used in the learning often represent the visual or textual attributes of video shots. The upside of Figure 5 illustrates the diagram of the query-dependent learning.

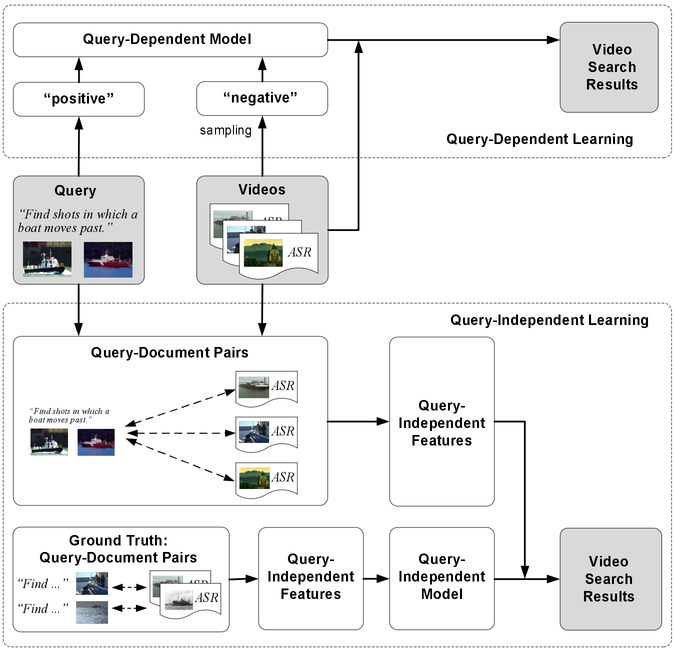


Fig. 5. The diagrams of query-dependent and query-independent learning.

Implementation: We used the query examples as pseudo positives and randomly sampled the shots as pseudo negative, and then built 10 models for each query and then average the predicted scores as the final results. We use SVM to train the query models with RBF kernels. Due to the low number of training samples, we had no validation dataset to choose the optimal kernel parameters, i.e., C (the cost parameter in soft-margin SVM) and γ (the width of the RBF function); thus we selected global kernel parameters. The low-level features we used are: color moments on a 5-by-5 fixed grid (225-D), wavelet based textures (128-D) and edge distribution histogram (75-D).

3.1.2. Query-independent learning

Formulation: Query-independent learning [10] takes each query-shot pair as a sample, as shown in the downside of Figure 5. The aim is to learn the relevance relation (relevant or irrelevant) from these pairs. The features used in query-independent learning should indicate the relevance relation between query-shot pairs. In other word, the features measuring such relevance relation are extracted from each query-

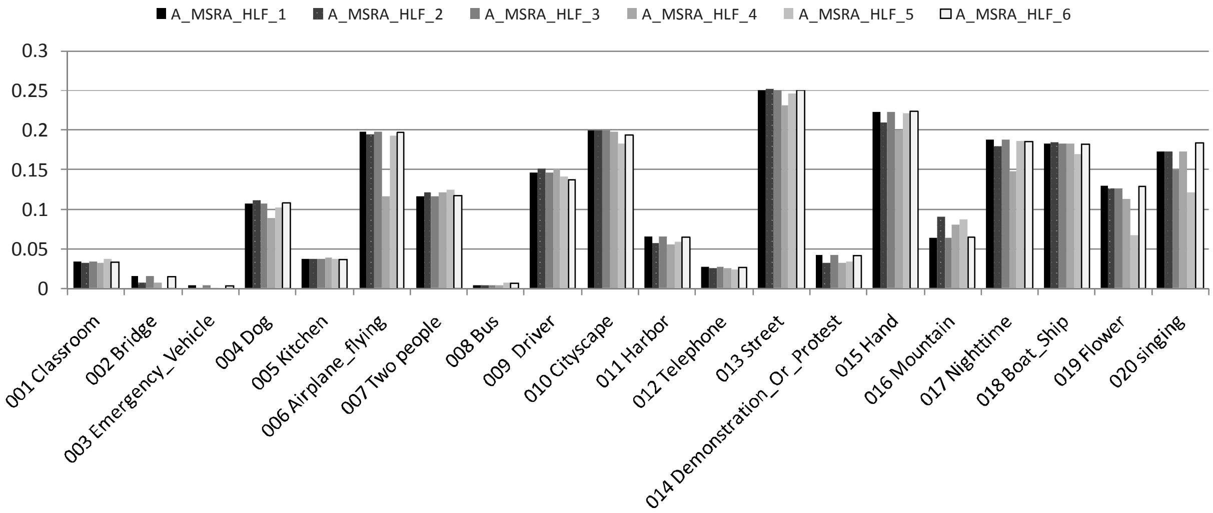


Fig. 3. The AP performance of MSRA_HLF six runs submitted to TRECVID 2008.

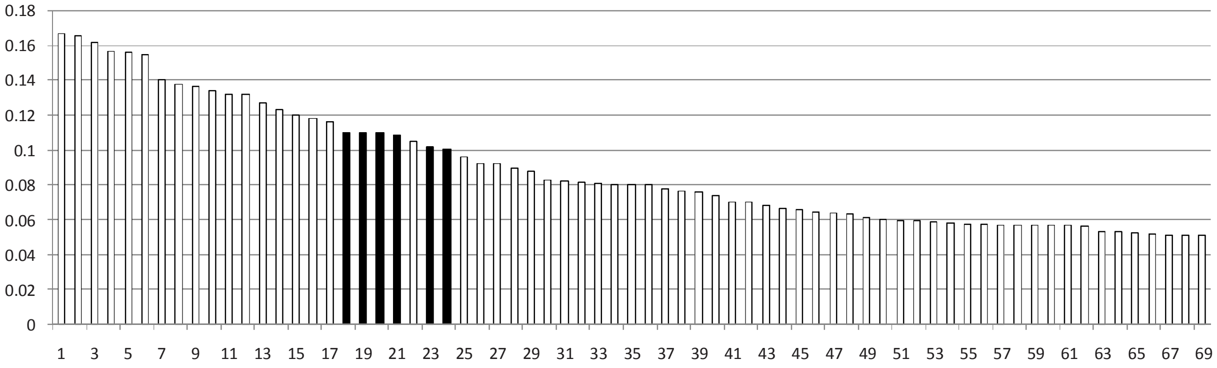


Fig. 4. Overview of Top 70 high-level feature extraction runs submitted to TRECVID 2008, ranked according to MAP. The black bars correspond to the performances of MSRA_HLF.

shot pair, rather than from each query or shot. The common relational features are as follows:

a)*Visual term frequency (VTF)*: The more often query terms occur in a shot; the more likely the shot is to be relevant to the query. Visual term frequency is such a measure to relate query and shot, it is given by:

$$VTF(i,j) = \sum\nolimits_{k=1}^{H_j} vtf(i,j,k) \tag{6}$$

where $vtf(i,j,k)$ denotes the score of visual term t_j^k in shot d_i. If the visual term is not the related concept of the shot, the score is zero.

To produce other more effective features, we introduce *VIDF* (Inverse visual document frequency). It was developed based on the intuition that a query term which occurs in many shot, like *people* is not a good discriminator.

$$\begin{aligned} VIDF(j) &= \sum\nolimits_{k=1}^{H_j} vidf(j,k) \\ &= \sum\nolimits_{k=1}^{H_j} \log \frac{M+0.5}{m(j,k)+0.5} \end{aligned} \tag{7}$$

where $m(j,k)$ is the number of shots which visual term t_j^k occurs in, 0.5 is a smoothing correction used to avoid denominator equaling to 0. Note that *VIDF* is a constant value to the same query.

b)*Visual TFIDF, BM25 (VTFIDF,VBM25)*: Some features obtained by combining *VTF* and *VIDF*, as well as the visual document length $dl(i)$ (i.e., the sum scores of all the related concepts in shot d_i). For example, *VTFIDF* and *VBM25* are such kind of features, which are defined as:

$$VTFIDF(i,j) = \sum\nolimits_{k=1}^{H_j} vtf(i,j,k) \times vidf(j,k) \tag{8}$$

$$VBM25(i,j) = \sum\nolimits_{k=1}^{H_j} \frac{vtf(i,j,k) \times vidf(j,k) \times (a+1)}{a \times (1-b+b \times ndl(i)) + vtf(i,j,k)} \tag{9}$$

where $ndl(i)$ is normalized visual document length, defined with the average length of visual documents $\bar{dl}$: $ndl(i) = dl(i)/\bar{dl}$. a and b are tuning constants.

c)*Visual query term distribution (VQD)*: For a query contains multiple visual terms, the more different terms occur in a shot, the more likely it is relevant to the query. *VQD* is defined as the number of different query terms occurrences in a shot.

$$VQD(i,j) = \sum\nolimits_{k=1}^{H_j} q(i,j,k) \tag{10}$$

where $q(i,j,k)$ is a binary function to indicate whether visual term t_j^k occurs in visual document d_i.

Implementation: We took $log(x + 1.0)$ function on all the relational feature values in order to reduce the effects of large numbers, where x is the original feature values. Then, we also exploited SVM with RBF kernels to train the models. TRECVID 2007 test set were used for validation, while TRECVID 2005 and TRECVID 2006 were used for training. The models were built upon training set while the parameters (C and γ) were tuned in validation set. Thus we had two results, and then fused them in average.

3.1.3. Bayesian reranking

Formulation: In Bayesian reranking method [11], with the visual consistency assumption, reranking is explicitly formulated into a global optimization problem. For a query q, when given its top-N relevant samples (video shots)$\{x_1, x_1, \ldots, x_N\}$ returned by the search engine and their initial text search result represented by the ranking score list $\bar{\mathbf{r}} = [\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N]^T$ where $\bar{r}_i$ is the ranking score of sample $x_i, 1 \leq i \leq N$, the optimal reranked score list $\mathbf{r}^*$ is obtained by minimizing the following energy function:

$$E(\mathbf{r}) = \mathrm{Re}g(\mathbf{G}, \mathbf{r}) + c * Dist(\mathbf{r}, \bar{\mathbf{r}}) \tag{11}$$

The first term in Eq.(11) is the regularization term which penalizes the ranking score inconsistency between the visually similar samples while the second term is the ranking distance term which penalizes the derivation of the reranked result from the initial ranking. c is a trade-off parameter which balances the influence of the two terms.

For the regularization term, a local kernel regularizer is adopted to model multiple-wise visual consistency, $\mathrm{Re}g(\mathbf{G}, \mathbf{r}) = \mathbf{r}^T R_L \mathbf{r}$, R_L is the local kernel regularization matrix.

In local kernel regularizer, for each sample x_i, a local learner $o_i(\cdot)$ is trained locally with the data $\{(x_j, r_j)\}_{x_j \in Nei(x_i)}$, where $Nei(x_i)$ denotes the set of x_i's neighboring samples. x_i's ranking score can be predicted by this local learner. Then the regularization term can be modeled by aggregating the local learner's prediction loss on each sample:

$$\mathrm{Re}g(\mathbf{G}, \mathbf{r}) = \sum_i (r_i - o_i(x_i))^2 \tag{12}$$

The kernel ridge regression is adopted to model the dependencies between $Nei(x_i)$ and its score vector $\mathbf{r}_i = [r_j]^T_{x_j \in Neg(x_i)}$, i.e., $o_i(x) = \mathbf{w}^T \phi(x)$. By solving this problem, we can get $\mathbf{w} = \Phi_i(\Phi_i^T \Phi_i + \lambda \mathbf{I})^{-1} \mathbf{r}_i$ where Φ_i denotes matrix $[\phi(x_j)]^T$ for $x_j \in Neg(x_i)$. Then, for x_j, the score predicted by its local learner $o_i(\cdot)$ is: $o_i(x_i) = \mathbf{w}^T \phi(x_i) = \mathbf{k}^T (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{r}_i = \beta_i^T \mathbf{r}_i$. $\mathbf{k}$ is a vector with $k_j = \phi(x_i)^T \phi(x_{t_j}) = k(x_i, x_{t_j})$ and $\mathbf{K}$ is a matrix with $k_{mn} = \phi(x_{t_m})^T \phi(x_{t_n}) = k(x_{t_m}, x_{t_n})$, $j, m, n = 1, 2, \ldots, |Nei(x_i)|$ and $x_{t_j}, x_{t_m}, x_{t_n} \in Nei(x_i)$ with t_p is the subscript of the p-th sample in $Nei(x_i)$. Then, the local kernel learning regularizer is formulated as:

$$\begin{aligned} \mathrm{Re}g(\mathbf{G}, \mathbf{r}) &= \sum_i (r_i - o_i(x_i))^2 \\ &= \sum_i \left(r_i - \beta_i^T \mathbf{r}_i\right)^2 = \mathbf{r}^T R_L \mathbf{r} \end{aligned} \tag{13}$$

where $\beta_i^T = \mathbf{k}^T (\lambda \mathbf{I} + \mathbf{K})^{-1}$, $R_L = (\mathbf{I} - \mathbf{B})^T (\mathbf{I} - \mathbf{B})$ and $\mathbf{B} = [b_{ij}]_{N \times N}$, b_{ij} equals the corresponding element of β_i if $x_j \in Nei(x_i)$, otherwise $b_{ij} = 0$.

For the distance term, a point-wise distance is adopted for the ranking distance $Dist(\mathbf{r}, \bar{\mathbf{r}}) = (\mathbf{r} - \bar{\mathbf{r}})^T (\mathbf{r} - \bar{\mathbf{r}})$.

With these two terms, a closed-form solution for $\mathbf{r}$ is derived as:

$$\mathbf{r} = (\mathbf{R}_L + \mathbf{C})^{-1} \mathbf{C} \bar{\mathbf{r}} \tag{14}$$

where $\mathbf{C} = diag(c_1, c_2, \ldots, C_N)$ with $c_i = c$ for all samples.

Implementation: For each query, we only reranked the top 1,400 video shots returned by the text search engines. The initial text search ranking score list $\bar{\mathbf{r}}$ is set according to the Normalized Rank strategy given in [11]. Then, the reranked score list $\mathbf{r}$ can be calculated as $\mathbf{r} = (\mathbf{R}_L + \mathbf{C})^{-1} \mathbf{C} \bar{\mathbf{r}}$. The final ranking list is obtained by sorting $\mathbf{r}$ with descending order.

3.1.4. MIIL reranking

Formulation: Conventional supervised reranking approaches empirically convert the reranking as a classification problem in which each document is determined relevant or not, followed by reordering the documents according to the confidence scores of classification. In fact, reranking can be also viewed as an optimization problem in which the ranked list is globally optimal if any two arbitrary documents from the list are correctly ranked in terms of relevance.

MIIL reranking [12] directly optimize video search reranking from a viewpoint of information theory, that is, to identify an optimal set of correctly-ranked document pairs which maximally preserves the relevant information Y^+ and simultaneously carries the irrelevant information Y^- as little as possible. Let t be an element of the pair set $\tilde{T}$, the mutual information between t and Y (Y^+ or Y^-), $I(t, Y)$ is defined as:

$$\begin{aligned} I(t, Y) &= p(t, Y) \log \tfrac{p(t,Y)}{p(t) \times p(Y)} \\ &= \sum\nolimits_{y \in Y} p(t, y) \log \tfrac{p(t,y)}{p(t) \times p(y)} \\ &= p(t) \sum\nolimits_{y \in Y} p(y|t) \log \tfrac{p(y|t)}{p(y)} \end{aligned} \tag{15}$$

The dual optimization task can be approached by maximizing the weighted difference:

$$\begin{aligned} L(\tilde{T}) &= \sum\nolimits_{t \in \tilde{T}} L(t) \\ &= \sum\nolimits_{t \in \tilde{T}} I(t, Y^+) - \lambda I(t, Y^-) \end{aligned} \tag{16}$$

where $L(t) = I(t, Y^+) - \lambda I(t, Y^-)$, λ determines the tradeoff between preservation of the relevant information Y^+ and loss of the irrelevant information Y^-. Thus the reranking criterion is given by:

$$\tilde{T}^* = \arg\max_{\tilde{T} \subset T} L(\tilde{T}) \tag{17}$$

When the optimal pair set $\tilde{T}^*$ is obtained, a round robin criterion is introduced to obtain the final reranked list $Z = F(T^*)$. Specifically, if t_{ij} (i.e., $x_i \succ x_j$) is an element of the optimal pair set $\tilde{T}^*$, we assign a vote to x_i. Conversely, the prediction $x_j \succ x_i$ would be considered as a vote for x_j. Then add all the votes assigned to each sample, and the samples are finally ranked in descending order of the sum of the votes they are assigned.

To solve the above optimization problem defined in Equation (17), we represent the relevant/irrelevant information as a set of concept detections. The prior distribution of concept y, $p(y)$, is estimated by the distribution of training data of concept detection, and the posterior probability of pair samples $p(y|t_{ij})$ and the prior distribution of pair samples $p(t_{ij})$ are defined as:

$$p(y|t_{ij}) = \frac{1}{1 + e^{-m \times [f(x_i, y) - f(x_j, y)]}} \tag{18}$$

$$p(t_{ij}) = \frac{1}{1 + e^{-n \times [g(x_i) - g(x_j)]}} \tag{19}$$

where m and n determine the confidence of information learned from query examples and the confidence of initial search results, respectively.

The mapping function $f : X, Y \rightarrow \Re$ and the initial model $g : X \rightarrow \Re$ are defined as follows:

$$f(x, y) = detection\ score\ of\ sample\ x\ for\ concept\ y \tag{20}$$

$$g(x) = order\ of\ sample\ x\ in\ initial\ ranked\ list \tag{21}$$

Implementation: Inspired by the lossy information compression theory, we can view reranking as "denoising" problem, such that "noise" is defined as the incompressible part in the data while the compressible part defines the meaningful information bearing signal [13]. Thus we select "the best possible pair" at each round, and "the best possible pair" is viewed as the compressed data which preserves the most relevant information while excludes the most irrelevant information. Let $t^{(i)}$ be the selected "the best possible pair" at the i^{th} round, $T^{(i)}$ be the current pair sample set and $T^{(i+1)}$ denote the new pair sample set after the minus of several pairs in $T^{(i)}$. The reranking can be formulated as:

$$t^{(i+1)} = \arg\max_{t \in T^{(i)}} \{I(t, Y^+) - \lambda I(t, Y^-)\} \tag{22}$$

After pair selecting at each round, we map the selected pair into the new ranked list. At the i^{th} round, the two samples of the selected pair are located at the rank i and rank $N^{(0)} - i + 1$ in the new ranked list Z.

The criterion presented in Equation (22) is equivalent to finding the pair which has least information loss:

$$t^{(i+1)} = \arg\min_{t \in T^{(i)}} \{I(T^{(i)}, Y) - [I(t, Y^+) - \lambda I(t, Y^-)]\} \tag{23}$$

where $I(T^{(i)}, Y) = I(T^{(i)}, Y^+) - \lambda I(T^{(i)}, Y^-)$, it is a constant for each $t \in T^{(i)}$. Obviously, the information loss is incremental with rounds increasing. Thus the reranking is formulated as finding most confidential pairs via *minimum incremental information loss* (MIIL).

3.2. Fusion method

The fusion methods used in this year include simple averaging and weighted averaging fusion. For weighted averaging fusion, we simply classified all the queries to four classes, i.e., "object," "people-object," "object-event" and "people-event," and then the weight for each class is validated using TRECVID 2007 test dataset and queries. The four classes are defined as follows:

Object: queries for a certain type of objects, such as "0225: a bridge" and "0250: an airplane exterior."

People-object: queries for finding both objects and persons, such as "0223: one or more people with one or more horses" and "0229: one or more people where a body of water can be seen."

Object-event: queries for finding a scene, which contains an event with one or more objects, such as "0236: waves breaking onto rocks" and "0244: a vehicle approaching the camera."

People-event: queries for finding a scene, which contains an event with one or more persons, such as "0221: a person opening a door" and "0252: one or more people, each riding a bicycle."

We performed POS (Part-of-speech tagging) on the queries with Tree-tagger [14][6]. POS represents the syntactic property of a term, e.g., noun, verb, adjective, and so on. On the other hand, we built a person-related terms lexicon, which contains "people," "person," "man," "woman," "baby" and so on.

By labeling each term of a query with POS tags and person-related tag, we classified the queries only containing noun tags but no person-related tags into "object," queries containing noun and person-related tags into "people-object," queries containing noun and verb tags into "object-event," queries containing person-related and verb tags into "people-event."

3.3. Experiments and Results

We submitted five automatic type A runs for search task. Figure 6 shows the performances of five submitted runs for automatic search. Our two baseline runs had the MAP scores

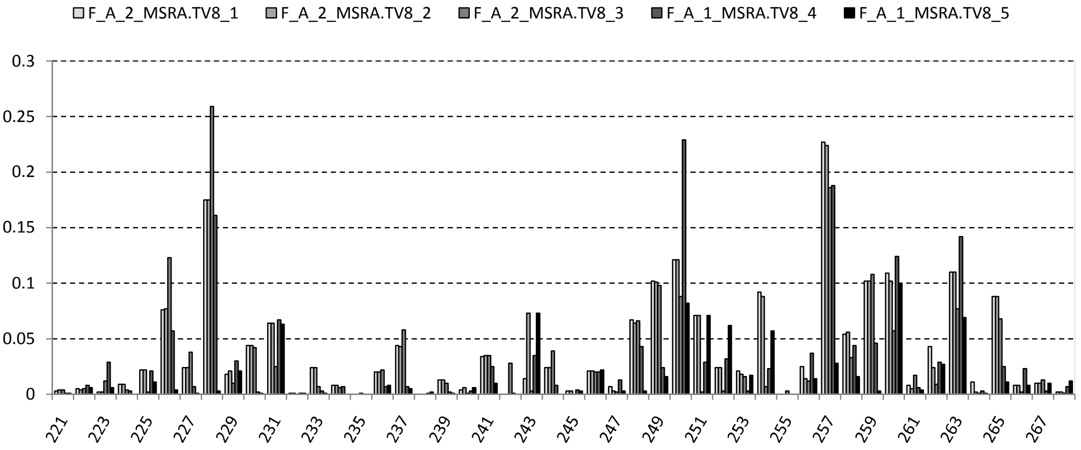


Fig. 6. The AP performance of MSRA_SEARCH five runs submitted to TRECVID 2008.

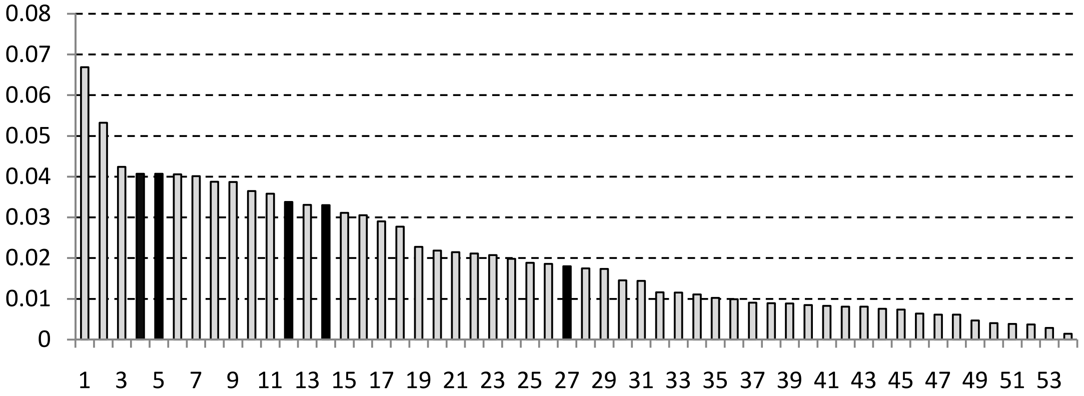


Fig. 7. Overview of top 54 automatic runs of type A submitted to TRECVID 2008, ranked according to MAP. The black bars correspond to the performances of MSRA_SEARCH.

of 0.0338 and 0.0180, ranked in the top 12 and 27 among all the type A automatic runs as shown in Figure 7, respectively. When we introduced some reranking methods over the text baseline (MAP:0.0180), we gain the MAP score of 0.0330. When we fused text and visual baseline, we gain the MAP score of 0.0407. Then we exploited temporal expansion over the fused results, we found some topics have improved AP and some topics have lower AP, the MAP score had no improvement.

4. CONCLUSIONS

We participated high-level feature extraction and automatic search tasks in TRECVID 2008. In this paper, we have presented preliminary results and methods for these two tasks.

5. REFERENCES

[1] M. Wang, X.-S. Hua, X. Yuan, Y. Song, and L.-R. Dai, "Optimizing multi-graph learning: towards a unified video annotation scheme," in *Proceedings of ACM Multimedia*, 2007.

[2] J. Wang, Y. Zhao, and X. S. Hua, "Transductive multi-label learning for video concept detection," in *Proceedings of ACM International Conference on Multimedia Information Retrieval*, Vancouver, Canada, October 2008.

[3] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[4] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[5] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

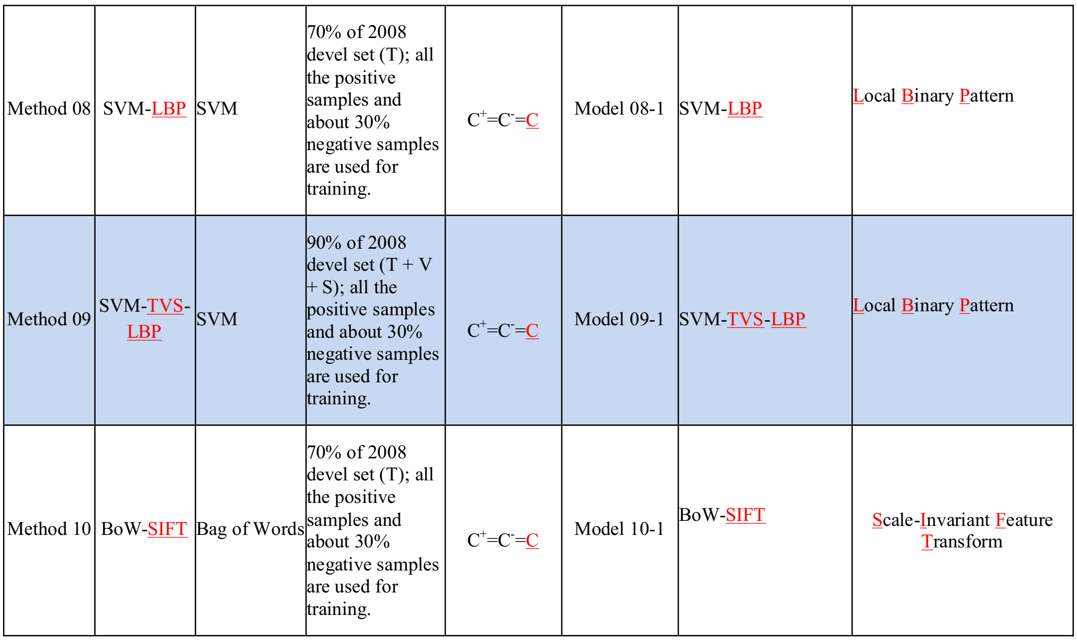
[6] T. Mei, X.-S. Hua, W. Lai, L. Yang, Z.-J. Zha, Y. Liu, Z. Gu, G.-J.Qi, M. Wang, J.Tang, X. Yuan, Z. Lu, and J.Liu, "MSRA-USTC-SJTU AT TRECVID 2007: HIGH-LEVEL FEATURE EXTRACTION AND SEARCH," in *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

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

Method ID	Method Name	Classifier	Training Data	Weights	Model ID	Model Name	Modality (Low-level feature)
Method 01	SVM-C	SVM	70% of 2008 devel set (T); all the positive samples and about 30% negative samples are used for training.	$C^+=C^-=C$	Model 01-1	SVM-CM33	Color Moment 3-by-3
					Model 01-2	SVM-CM55	Color Moment 5-by-5
					Model 01-3	SVM-CM77	Color Moment 7-by-7
					Model 01-4	SVM-Auto	Auto-correlogram
					Model 01-5	SVM-HSV	HSV Color Histogram
					Model 01-6	SVM-EDH	Edge Distribution
					Model 01-7	SVM-Wave	Wavelet PWT-TWT
					Model 01-8	SVM-Face	Face
Method 02	SVM-C^+C^-	SVM	70% of 2008 devel set (T)	$C^+/C^- = N^-/N^+$	Model 02-1	SVM-C^+C^--CM33	Color Moment 3-by-3
					Model 02-2	SVM-C^+C^--CM55	Color Moment 5-by-5
					Model 02-3	SVM-C^+C^--CM77	Color Moment 7-by-7
					Model 02-4	SVM-C^+C^--Auto	Auto-correlogram
					Model 02-5	SVM-C^+C^--HSV	HSV Color Histogram
					Model 02-6	SVM-C^+C^--EDH	Edge Distribution
					Model 02-7	SVM-C^+C^--Wave	Wavelet PWT-TWT
					Model 02-8	SVM-C^+C^--Face	Face
Method 03	SVM-C^+10C^-	SVM	70% of 2008 devel set (T)	$C^+/C^- = 10*N^- /N^+$	Model 03-1	SVM-C^+10C^--CM33	Color Moment 3-by-3
					Model 03-2	SVM- C^+10C^--CM55	Color Moment 5-by-5
					Model 03-3	SVM- C^+10C^--CM77	Color Moment 7-by-7
					Model 03-4	SVM- C^+10C^--Auto	Auto-correlogram
					Model 03-5	SVM- C^+10C^--HSV	HSV Color Histogram
					Model 03-6	SVM- C^+10C^--EDH	Edge Distribution
					Model 03-7	SVM- C^+10C^--Wave	Wavelet PWT-TWT
					Model 03-8	SVM- C^+10C^--Face	Face
Method 04	SVM-TVS-C^+C^-	SVM	90% of 2008 devel set (T+V+S)	$C^+/C^- = N^-/N^+$	Model 04-1	SVM-TVS -CM33	Color Moment 3-by-3
					Model 04-2	SVM-TVS -CM55	Color Moment 5-by-5
					Model 04-3	SVM-TVS -CM77	Color Moment 7-by-7
					Model 04-4	SVM-TVS -Auto	Auto-correlogram
					Model 04-5	SVM-TVS -HSV	HSV Color Histogram
					Model 04-6	SVM-TVS -EDH	Edge Distribution
					Model 04-7	SVM-TVS -Wave	Wavelet PWT-TWT
					Model 04-8	SVM-TVS -Face	Face
Method 05	SVM-0805-C^+C^-	SVM	70% of 2008 devel set (T) and the positive samples over 2005 devel set.	$C^+/C^- = N^-/N^+$	Model 05-1	SVM-0805-CM33	Color Moment 3-by-3
					Model 05-2	SVM-0805-CM55	Color Moment 5-by-5
					Model 05-3	SVM-0805-CM77	Color Moment 7-by-7
					Model 05-4	SVM-0805-Auto	Auto-correlogram
					Model 05-5	SVM-0805-HSV	HSV Color Histogram
					Model 05-6	SVM-0805-EDH	Edge Distribution
					Model 05-7	SVM-0805-Wave	Wavelet PWT-TWT
					Model 05-8	SVM-0805-Face	Face
Method 06	OMGSSL	Optimal Multi-Graph Semi-Surprised Learning (OMGSSL)	70% of 2008 devel set (T)	$C^+/C^- = N^-/N^+$	Model 06-1	OMGSSL-CM33	Color Moment 3-by-3
					Model 06-2	OMGSSL-CM55	Color Moment 5-by-5
					Model 06-3	OMGSSL-CM77	Color Moment 7-by-7
					Model 06-4	OMGSSL-Auto	Auto-correlogram
					Model 06-5	OMGSSL-HSV	HSV Color Histogram
					Model 06-6	OMGSSL-EDH	Edge Distribution
					Model 06-7	OMGSSL-Wave	Wavelet PWT-TWT
					Model 06-8	OMGSSL-CTF	Concatenated feature, including Co-occurrence Texture and Face
Method 07	TML	Transductive Multi-Label Learning	70% of 2008 devel set (T)	$C^+=C^-=C$	Model 07-1	TML-CM33	Color Moment 3-by-3
					Model 07-2	TML -CM55	Color Moment 5-by-5
					Model 07-3	TML -CM77	Color Moment 7-by-7
					Model 07-4	TML -Auto	Auto-correlogram
					Model 07-5	TML -HSV	HSV Color Histogram
					Model 07-6	TML -EDH	Edge Distribution
					Model 07-7	TML -Wave	Wavelet PWT-TWT

[8] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[9] C. Petersohn, “Fraunhofer hhi at trecvid 2004: Shot boundary detection system,” in *TREC Video Retrieval Evaluation Online Proceedings*, URL: www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf, 2004.



Method 08	SVM-LBP	SVM	70% of 2008 devel set (T); all the positive samples and about 30% negative samples are used for training.	C$^+$=C$^-$=C	Model 08-1	SVM-LBP	Local Binary Pattern
Method 09	SVM-TVS-LBP	SVM	90% of 2008 devel set (T + V + S); all the positive samples and about 30% negative samples are used for training.	C$^+$=C$^-$=C	Model 09-1	SVM-TVS-LBP	Local Binary Pattern
Method 10	BoW-SIFT	Bag of Words	70% of 2008 devel set (T); all the positive samples and about 30% negative samples are used for training.	C$^+$=C$^-$=C	Model 10-1	BoW-SIFT	Scale-Invariant Feature Transform

Fig. 8. The description of all the methods and models for high-level feature extraction.

[10] Y. Liu, T. Mei, G.-J. Qi, X. Wu, and X.-S. Hua, "Query-independent learning for video search," in *Proceedings of IEEE International Conference in Multimedia & Expo*, 2008.

[11] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua, "Bayesian video search reranking," in *Proceedings of ACM International Conference on Multimedia*, Vancouver, Canada, October 2008.

[12] Y. Liu, T. Mei, X. Wu, and X.-S. Hua, "Optimizing video search reranking via minimum incremental information loss," in *Proceedings of ACM Conference on Mutltimedia Information Retrieval*, 2008.

[13] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, 2000.

[14] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.