# Florida International University and University of Miami TRECVID 2009 - High Level Feature Extraction

Lin Lin, Chao Chen, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33124, USA
l.lin2@umiami.edu, c.chen15@umiami.edu, shyu@miami.edu

Fausto Fleites, Shu-Ching Chen
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
fflei001@cs.fiu.edu, chens@cs.fiu.edu

## Abstract

*In this paper, the details about FIU-UM group TRECVID2009 high-level feature extraction task submission are presented. Six runs were conducted using different feature sets, data pruning approaches, classification algorithms, and ranking methods. A proportion of TRECVID2009 development data were randomly sampled from the whole development data archives (all TRECVID2007 development data and test data), which include all positive data instances (target-high-level feature data) and partial negative data instances (around one-third non-target-high-level feature data) for each high-level feature. Two strategies dealing with the skipping/not-sure shots were also introduced. First four runs treated the skipping/not-sure data instances as positive instances in the training data (ALL), and the last two runs disregarded these skipping/not-sure data instances from the training data (PURE).*

- FIU-UM-1*: KF+ALL+CB+MCA+RANK, training on partial TRECVID2009 development data with all positive set (ALL) and using key-frame based low-level features (KF), correlation-based pruning (CB), MCA-based classifier (MCA), and ranking method (RANK). The RANK method uses the Euclidean distances of two selected features between each testing data instance and the positive training set as additional scores integrated with the scores from MCA-based classifier to obtain the final ranking scores.*

- FIU-UM-2*: KF+ALL+CB+MCA, training on partial TRECVID2009 development data with all positive set (ALL) and using key-frame based low-level features (KF), correlation-based pruning (CB), MCA-based classifier (MCA), and a ranking process used MCA-based scores from the classifier.*

- FIU-UM-3*: SF+ALL+DB+SB, training on partial TRECVID2009 development data with all positive sets (ALL) and using shot-based low-level features (SF), distance-based pruning (DB), subspace-based classifier (SB), and a ranking process used subspace-based scores from the classifier.*

- FIU-UM-4*: SF+ALL+DB+SB+SVMC, training on partial TRECVID2009 development data with all positive set (ALL) and using shot-based low-level features (SF), distance-based pruning (DB), subspace-based classifier (SB), and SVMC ranking method. The SVMC method brings the retrieval results from SVM with chi-square kernel (SVMC) and considers these results as additional scores which are later combined with subspace-based scores to form the final ranking scores.*

- FIU-UM-5*: KF+PURE+CB+MCA+RANK, training on partial TRECVID2009 development data with pure positive set (PURE) and using key-frame based low-level features (KF), correlation-based pruning (CB), MCA-based classifier (MCA), and ranking method (RANK).*

- FIU-UM-6*: SF+PURE+DB+SB, training on partial TRECVID2009 development data with pure positive set (PURE) and using shot-based low-level features (SF), distance-based pruning (DB), subspace-based classifier (SB), and a ranking process used subspace-based scores from the classifier.*

*In the TRECVID2009 high-level feature extraction task submission, we are able to improve the framework in several ways. First, more key-frame based visual features (*$513$*) were extracted in addition to the* $28$ *old shot-based features, and different normalization methods were applied. Second, all development data (*$219$ *videos) and testing data (*$619$ *videos) were processed. Third, a key-frame detection algorithm was implemented to extract the key-frames from testing videos, which are not provided by TRECVID. Fourth, different data pruning methods were proposed to solve the data imbalance issue, and from other experimental results, our proposed methods performs well on removing noisy data and selecting the typical positive and negative data instances. Fifth, two new classifiers were proposed in our framework rather than using the existing classifiers like Support Vector Machine, Decision Tree, etc. Finally, in addition to concept detection, we are able to extend our framework to the area of video retrieval. In other words, we proposed several scoring methods to rank the retrieved results.*

*However, we are still facing a lot of challenges. First, as can be seen from the description of each run, three runs by utilizing the CB+MCA model were trained by the key-frame based low/mid-level visual features. By adding some low-level audio features, the extraction performance for some high-level features would be improved, such as person-playing-a-musical-instrument, people-dancing, and singing. Similarly, more visual features would help the runs trained only by the shot-based feature data. Therefore, how to integrate the audio features with the key-frame based features and add more visual features with shot-based features need to be done. Second, to solve the data imbalance problem, the negative data instances were first randomly sampled. This is very risky since by doing this, the difference of the distribution of the training set and testing set could be enlarged. Then even the training performance is pretty good as in our experiments, the testing results may not be as good as expected. Therefore, more investigations on data sampling and data pruning should be considered. Third, from the results we could see that the ranking methods are not good enough. More research on ranking the retrieved results should be studied.*

# 1 Introduction

With the rapid development of the Internet and the decrease of the storage cost, the demand of accessing and collecting multimedia data is rapidly increasing. The high-level feature extraction task in the TRECVID project [9] addresses one of the biggest challenges in concept-based multimedia retrieval. To automatically detect high-level features (concepts) using low-level features extracted from video, audio, and text has become a popular research topic [8][10], facing the issues of semantic gap between the low-level features and high-level concepts, data imbalance between positive and negative instances, etc.

In our previous studies, we have been able to demonstrate the effectiveness of MCA (Multiple correspondence analysis) in learning the correlation between low-level features and high-level concepts. In [3], MCA was used to learn the correlation between feature-value pairs and classes in order to generate classification rules for different concepts from TRECVID2007 high-level feature extraction task. We used our work described in [3] with several modifications to produce the results submitted to the TRECVID2008 high-level feature extraction task with two runs. Later, we were able to extend the 1-feature-value pair rules to 2-feature-value pair rules in [6] and to $n$-feature-value pair rules in [4]. The evaluation used the concepts from TRECVID2007 and TRECVID2008 high-level feature extraction tasks and the performance was compared to several well-known classification algorithms. Our results were promising and demonstrated superior performance in both balanced and imbalanced data sets. For TRECVID2009 task submission, MCA was utilized for data pruning and classification. Moreover, the correlation information and percentage information combined for classification can be further utilized to rank the retrieved results.

In our previous work, subspace-based (SB-based) modeling on the principle component space has been utilized for multi-class classification [1]. The SB-based classification algorithm in [1] was robust to outliers and was effective even for imbalanced data sets. Compared with the other learning algorithms, such as SVM and Decision Tree, the SB-based classification algorithm showed its merits not only on its good performance, but also on its ability of dimension reduction. In TRECVID2009, our proposed SB-based framework is able to build a reasonable model by making a balance between anomalous data and boundary data. Moreover, it can provide a ranking score based on its distance toward the concept model.

This paper is organized as follows. In Section 2, the proposed framework and the detailed discussion on each component are presented. Section 4 discusses our experiments as well as our observations. This paper is concluded in Section 5.

# 2 The Proposed Video Semantic Concept Detection Framework

The proposed multimedia content-based concept detection framework that produced the results submitted to the TRECVID2009 high-level feature extraction task consists of 4 stages (shown in Figure 1), namely data preparation, data pruning, model training and classification, and ranking. The details of each component used in different approaches are introduced in the following sections.

## 2.1 Data Preparation

Based on the provided shot boundary and key-frame information, both shot-based and key-frame based features are captured. Note that for the testing videos, the key-frame information is not provided.


**Figure 1. The Proposed Framework.**

Since segmentation is beyond the scope of this task, the key-frame detection algorithm is not introduced in this paper. In shot-based feature set, there are 16 audio features, 11 visual features, and 1 meta feature. The meta feature is the length of the shots. The low-level audio features are exploited in the time-frequency domain, which are divided into volume-related, energy-related, and spectrum-flux-related features. Furthermore, the average zero crossing rate is added. The low-level shot-based visual features are pixel change, pixel histogram change, background mean, background variance, dominant color (RGB) values, grass ratio, and features to estimate the motion intensity of the video shot like center to corner ratio, etc. In the key-frame based feature set, 16-dimensional features representing color dominant in RGB space, 51-d features for color histogram in HSV space, 108-d local features for color moment in YCbCr space, 47-d features for edge histogram, 36-d features for texture co-occurrence, 219-d features for texture wavelet, 3-d features for Tamura texture, 24-d features for Gabor texture, and 1 feature representing the local binary patterns are extracted. In addition, 8 more mid-level features from face detection are captured, such as the number of faces, among others. All the 28 shot-based features are used, and 20 key-frame based features are selected from 513 features for each high-level feature using the chi-squared attribute evaluation algorithm in WEKA [11].

Once all the audio and visual features are extracted, each feature set for each video is normalized to reduce the effects caused by the fact that different videos were broadcasted differently. Two different normalization methods were applied. One is min-max method which is to subtract the minimum value and divide by the distance between the maximum and the minimum values for each feature. The other is z-score normalization, which gives the range between the raw feature and the population mean in units of the standard deviation. The aforementioned feature extraction process generates a set of continuous numerical features. Since MCA takes only nominal data, all the extracted features are discretized when the MCA-based methods are utilized. The method of discretization described in [2] uses the information gain as the disparity measure. If only one partition yields, the mean value of the feature is used as the disparity measure to make sure that at least two partitions for each feature are provided after discretization. For discretizing the testing data set, the same partitions obtained from the training data are used. The partitions created in the discretization process are called feature-value pairs.

### 2.2 CB Pruning + MCA Classifier + Rankings

Correspondence Analysis (CA) refers to a technique designed to analyze simple two-way tables which contain some measure of correspondence between the rows and columns. Multiple correspondence analysis (MCA) is an extension of the standard CA to more than two variables [7]. MCA analyzes a set of observations described by a set of nominal variables. Each of these variables comprises several levels which are coded by MCA, and each level is coded to a binary column. For each nominal variable, only one of the columns (levels) can get a value of 1. CA analyzes this indicator matrix but MCA analyzes the product of such coded matrix, which results in the generation of the Burt table. The functionality of MCA motivated us to explore its utilization to analyze labeled instances described by a set of low-level features to capture the correspondence between the feature-value pairs and the investigated concept classes. The chi-square distance among tabulations of the Burt table is applied, and then principle components are captured by using singular value decomposition (SVD). This allows us to project our multimedia data set into a new space by using the first and second principle components in the 2-d space. The similarity of each feature-value pair and every concept class can be represented by the angle between them. Then the higher correlation is, the smaller the angle between the feature-value pair and the class. From the calculation of the inner product of a feature-value pair ($A_i^j$) and class ($C$: $C_p$ or $C_n$), the angle ($angle_i^j \in [0, 180]$) could be captured, where $i$ is from $1$ to $I$ total number of features, $j$ varies for each feature from $1$ to the number of partitions for each feature, $C_p$ is the target concept class, and $C_n$ is the non-concept class. $angle_i^j$ between feature-value pair $A_i^j$ and class label $C_p$ being smaller than $90$ degrees shows that the feature-value pair has a higher correlation relationship with the positive class; while $angle_i^j$ between $A_i^j$ and $C_n$ being smaller than $90$ degrees demonstrates that the feature-value pair indicates a higher correlation relationship with the negative class. If $angle_i^j$ is equal to $90$ degrees, it means that the feature-value pair has equal correlation relationships with both positive and negative classes.

Here, a score for each data instance is calculated by MCA-based correlation. The score is defined as $score_k$ (shown in Equation (1)) and calculated by the sum of its feature-value pair weights $weight_i^j$ (shown in Equation (2)), where the weights are converted from the angle values (shown in Equation (3)). $k$ is from $1$ to the total number of data instances, $i$ is from $1$ to $I$ (the total number of features), $j$ can be any value between $1$ and the number of partitions for the $i^{th}$ feature, and $C$ can be $C_p$ or $C_n$. In addition, if $angle_i^j$ between $A_i^j$ and $C_p$ is less than a certain threshold, then the positive sign is set; while if $angle_i^j$ between $A_i^j$ and $C_n$ is less than a certain threshold, then the negative sign is set. Otherwise, $weight_i^j$ is set to be $0$. Note that due to our definition of the score, when the absolute values of the corresponding score values are large, the data instances are considered as typical/pure positive or negative data instances. If a data instance score is larger than or equal to this threshold value, the data instance is considered to be estimated as a positive data instance. On the other hand, if a data instance score is smaller than this threshold value, the data instance is considered to be estimated as a negative data instance. If the class label of the data instance is the same as the estimation, the data instance is consider as a typical instance; otherwise, the data instance should not be used for training the models. For the threshold determination, different threshold values are applied to the training data set and the ones with the highest $F1$-score are selected as the thresholds.

$$score_k = \sum_{i=1}^{I} weight_i^j; \tag{1}$$

$$weight_i^j = \pm(180 - angle_i^j)/90; \tag{2}$$

$$angle_i^j = \arccos\left(\frac{A_i^j \cdot C}{|A_i^j||C|}\right). \tag{3}$$

The classification model is trained using the pruned training data set. The model is introduced in [5] and weighted association rules are generated by using the correlation information and percentage information. Performing classification by using these weighted association rules is that each testing data instance in the testing data set is checked to see if it includes any of the feature-value pairs. For those feature-value pairs that exist in the testing data instance, the sum of the weights of the matched feature-value pair rules is considered as a score to determine the class label assigned to the testing data instance. Moreover, we extended the scores to rank the retrieved results. Another method for ranking is simply selecting two low-level features for each high-level concept (feature), considering the positive data instances in the training set as the clustering center, and computing the Euclidean distances of these two features between each data instance in the testing set and the clustering center. These distances are used as additional scores integrated with the MCA-based ranking scores as the final scores for ranking.

# 3 DB Pruning + SB Classifier + Rankings

In order to build a representative training model, a necessary step is to select typical training instances which are able to show the characteristics of the whole training data. Intuitively, outliers should not participate in building the model since their existence usually compromises the performance of the training model. In the training stage of our framework, both model $A$ for typical positive instances and model $B$ for typical negative instances are built, respectively. The idea is to remove a small proportion of the training data instances step by step based on their Mahalanobis distance until the training model reaches a local optimization point. As for model $A$, the Mahalanobis instance of each positive data instance is first calculated. After sorting the explored Mahalanobis distances, a pruning step is employed to remove a proportion of positive data instances with large Mahalanobis distances.

Next, a score distance to measure the dissimilarity of one instance towards both model $A$ and model $B$ is constructed. As for positive data instances, z-score standardization is first applied to prevent them from being dominated by a few large-scale attributes. The result of the normalization is that the columns of the data satisfy standard normal distribution. Although these columns of data show strong statistical characteristics, the dependency hiding among them prevents them from being able to be analyzed directly by adopting singular value decomposition (SVD). The projection of the original columns on the subspace which is spanned by the eigenvectors from the process of the decomposition results in columns of uncorrelated data with zero mean. Another important property is that the variance of one column $var(y_j) = \lambda_j$. Therefore, $y_j^2/\lambda_j$ satisfies chi-square distribution with the freedom of one. Then a combination of principal components (PCs) is selected, which can yield the best performance for the training model. Hence, we construct a distance measure **D** which is denoted in Equation (4), where **y** is the projected data on the subspace spanned by the selected eigenvectors for SVD, $\lambda_l$ is the $l^{th}$ selected eigenvalue from SVD, and **D** can be $Dpos$ for model $A$ or $Dneg$ for model $B$.

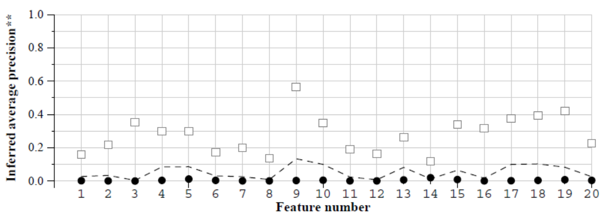
$$\mathbf{D} = \sum_{l} \frac{(\mathbf{y}_l)^2}{\lambda_l}. \tag{4}$$

It is worth mentioning that not all eigenvectors are considered in the calculation of **D**. Intuitively, some eigenvectors are so important that they must be included in this distance calculation, while others may be trivial or even may worsen the performance. Therefore, it requires an eigenvector selection process before calculating **D**. In our framework, a forward selection hill climbing searching algorithm is used to select the best combination of eigenvectors to construct the subspace. There are two merits resulting from this selection: (1) fewer eigenvectors are retained, which means the dimension of the training data is reduced; and (2) trivial features are removed and more precise model is expected. **D** renders a way to rank the data instances according to the training model. Since **D** satisfies chi-square distribution for the concept instances, while negative instances may not hold this statistical characteristics. Therefore, a data instance with a large **D** value implies that the data instance does not fit the model well. For model $A$, a data instance with a large **D** value is probably negative. The same process will be applied to model $B$.

The classification rule built on the constructed distance measure is as follows. For data instance $k$, let $Dpos_k$ be the Distance toward the concept model and $Dneg_k$ be the Distance toward the negative (non-concept) model. If $Dpos_k - Dneg_k \geq$ threshold $T$, then we assign the positive label to data instance $k$. Otherwise, we assign a negative label to data instance $k$. The generated classification rule is later used in the testing stage to judge the class label of each data instance based on its distance toward the concept model and non-concept model. Our ranking strategy makes use of the difference distance value $score_k = Dpos_k - Dneg_k$ calculated from the previous classification step. For the model for the positive data instances, the one with the smallest score value ranks as the first, while the one with the largest value ranks as the last. Furthermore, to refine our ranking methods, the classification result using SVM chi-square kernel is integrated. Each positive data instances recognized by the concept model will deduct a few distances if it also recognized by the concept model trained using SVM chi-square kernel. It is hoped that by combining both classification results from different learning algorithms, the concept instances that show an obvious pattern belonging to the corresponding concept will be emphasized and therefore ranked higher than using the SB-based score only.

## 4 Experiments and Results

Our concept detection framework was trained using the partial TRECVID2009 development data, which is randomly sampled from all TRECVID2009 development data consists of all TRECVID2007 development data and all TRECVID2007 test data. The sampled data set includes all positive data sets (target-high-level feature data) and partial negative data set (around one-third non-target-high-level feature data) for each high-level feature. The six runs we have submitted were produced using all of the TRECVID2009 test data including all TRECVID2008 testing data plus additional TRECVID2009 testing data. Due to the fact that some concepts had a very low number of positive data instances, we used the skipping/not-sure shots as the positive instances in some runs. Otherwise, the skipping/not-sure shots are ignored and only the instances labeled as positive and negative are used for training. Our system was trained and tested as introduced in Section 2.



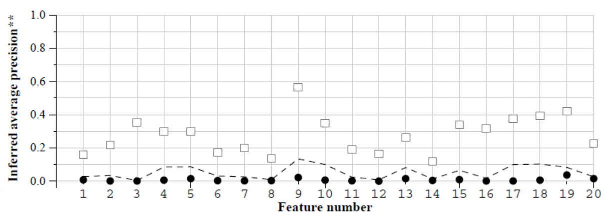




**Figure 2. Run scores (dot) versus median (—) versus best (box) for *FIU-UM-1* and *FIU-UM-2*.**

The TRECVID evaluations for the six runs can be seen in Figure 2, Figure 3, and Figure 4, respectively. *FIU-UM-1* is generated by KF+ALL+CB+MCA+RANK, training on partial TRECVID2009 development data with all positive sets and using key-frame based low-level features, correlation-based

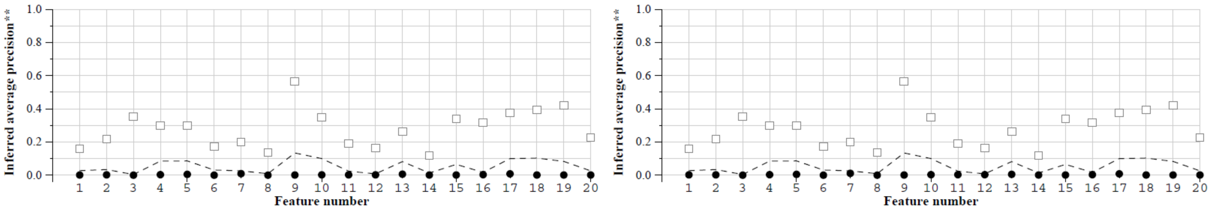



**Figure 3. Run scores (dot) versus median (—) versus best (box) for *FIU-UM-3* and *FIU-UM-4*.**

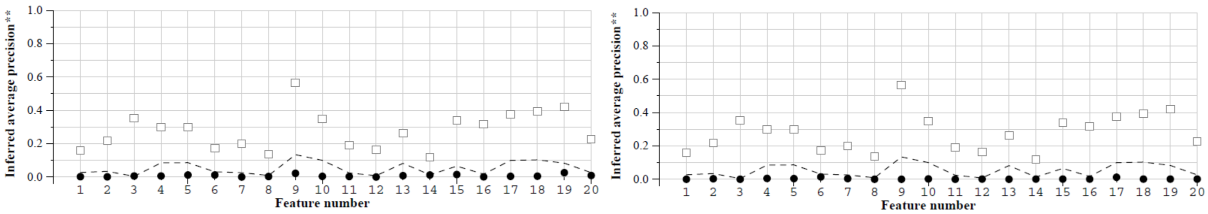



**Figure 4. Run scores (dot) versus median (—) versus best (box) for *FIU-UM-5* and *FIU-UM-6*.**

pruning, MCA-based classifier, and RANK ranking method. *FIU-UM-2* is KF+ALL+CB+MCA, training on partial TRECVID2009 development data with all positive sets, using key-frame based low-level features, correlation-based pruning, MCA-based classifier, and a ranking process used MCA-based scores from the classifier. *FIU-UM-3* is SF+ALL+DB+SB, training on partial TRECVID2009 development data with all positive sets and using shot-based low-level features, distance-based pruning, SB-based classifier, and a ranking process used subspace-based scores from the classifier. *FIU-UM-4* is SF+ALL+DB+SB+SVMC, training on partial TRECVID2009 development data with all positive sets, using shot-based low-level features, distance-based pruning, SB-based classifier, and SVMC ranking method. *FIU-UM-5* is KF+PURE+CB+MCA+RANK, training on partial TRECVID2009 development data with pure positive set and using key-frame based low-level features, correlation-based pruning, MCA-based classifier, and RANK ranking method. *FIU-UM-6* is SF+PURE+DB+SB, training on partial TRECVID2009 development data with pure positive set and using shot-based low-level features, distance-based pruning, SB-based classifier, and a ranking process used subspace-based scores from the classifier.

Based on the evaluation information returned to us from TRECVID, our second run *FIU-UM-2* performed the best and followed by *FIU-UM-5*. For all $20$ high-level features, out of $7036$ true positive shots, *FIU-UM-2* returned $656$ positive shots with the mean inferred precision of $0.008$, while *FIU-UM-5* returned $638$ positive shots with the mean inferred precision of $0.007$. For more observations, the average precision values at $n$ shots for six runs are listed in Table 1. By comparing the results between *FIU-UM-1* and *FIU-UM-5* runs, and between *FIU-UM-3* and *FIU-UM-6* runs, the results showed that using PURE positive set gives better results than using ALL positive set, although we expected that for a quite imbalance data set, more positive instances would help improve the accuracy as in the *FIU-UM-2* run. In other words, even though there are more positive data instances for ALL than PURE in the training set, the skipping/not-sure data instances may confuse the classifiers and thus reduce the accuracy. Based on the submitted results and our results produced in some of our previous studies, we believe that the MCA process has the capability to learn the correlation between low-level features and high-level features and to narrow the semantic gap between them. SB-based classifier is a new proposed method

**Table 1. The average precision at $n$ shots for six runs**

| runs | 5 | 10 | 15 | 20 | 30 | 100 | 200 | 500 | 1000 | 2000 |
|---|---|---|---|---|---|---|---|---|---|---|
| *FIU-UM-1* | 14.0% | 13.0% | 8.7% | 8.5% | 7.3% | 5.4% | 4.1% | 3.3% | 2.9% | 2.5% |
| *FIU-UM-2* | 20.0% | 13.0% | 13.3% | 11.0% | 9.7% | 8.1% | 5.7% | 4.8% | 3.9% | 3.3% |
| *FIU-UM-3* | 4.0% | 6.0% | 5.3% | 4.5% | 5.0% | 4.9% | 3.9% | 2.9% | 2.3% | 1.8% |
| *FIU-UM-4* | 8.0% | 8.0% | 7.3% | 7.5% | 7.0% | 5.3% | 3.9% | 3.0% | 2.3% | 1.8% |
| *FIU-UM-5* | 16.0% | 11.0% | 8.7% | 8.5% | 7.0% | 5.1% | 4.5% | 4.2% | 3.8% | 3.2% |
| *FIU-UM-6* | 4.0% | 7.0% | 5.3% | 4.5% | 4.0% | 4.9% | 3.6% | 2.6% | 2.0% | 1.5% |

this year, so more investigations are needed in the future work. It also can be seen from the results in the *FIU-UM-1* and *FIU-UM-2* runs, the RANK method did not help much. However, comparing the results in the *FIU-UM-3* and *FIU-UM-4* runs, the SVMC ranking method helped a little bit to the whole framework.

# 5 Conclusion

In this paper, a content-based video concept detection framework is introduced, which we have used to generate six runs submitted to the TRECVID2009 high-level feature extraction task. The details of different low-level and mid-level, audio and visual, shot-based and key-frame based extracted features, strategies dealing with the positive instances, algorithms of correlation-based and distance-based data pruning, MCA-based and SB-based classifiers, methods of ranking are given. The evaluation results of our framework are provided and discussed. In TRECVID2009 high-level feature extraction task submission, all development and testing data ($819$ videos) were processed. Moreover, $513$ new key-frame based visual features were extracted in addition to the $28$ old shot-based features, and different normalization methods were adopted. More features brought new aspects to investigate, such as feature set integration and feature selection. Two different data pruning methods were proposed to solve the data imbalance issue by removing some noisy data and selecting the typical data set for training. From the results and other experimental results, data pruning has shown its importance in the whole processing, and our proposed correlation-based and distance-based methods have shown to perform well. For the two new proposed classifiers and ranking methods, we are more interested in building new models rather than using the existing classifiers and ranking methods. A good classifier is important since it could retrieve more accurate results; while a good ranking method is also very important in the framework since it can list the most relevant retrieved results to the users. Therefore, in addition to high-level feature extraction and classifier investigation, we would like to extend our framework to provide ranking scores to rank the retrieved results. Though we are still facing a lot of challenges as we discussed earlier, we believe that after more investigations, we should be able to improve the efficiency of our concept detection and video retrieval framework.

# References

[1] C. Chen, M.-L. Shyu, and S.-C. Chen. Supervised multi-class classification with adaptive and automatic parameter tuning. In *IEEE International Conference on Information Reuse and Integration (IRI09)*, pages 433–434, August 2009.

[2] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, January 1992.

[3] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen. Correlation-based video semantic concept detection using multiple correspondence analysis. In *IEEE International Symposium on Multimedia (ISM08)*, pages 316–321, December 2008.

[4] L. Lin and M.-L. Shyu. Mining high-level features from video using associations and correlations. In *IEEE International Conference on Semantic Computing (ICSC09)*, pages 137–144, September 2009.

[5] L. Lin and M.-L. Shyu. Weighted association rule mining for video semantic detection. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, page accepted for publication, 2009.

[6] L. Lin, M.-L. Shyu, G. Ravitz, and S.-C. Chen. Video semantic concept detection via associative classification. In *IEEE International Conference on Multimedia and Expo (ICME09)*, pages 418–421, July 2009.

[7] N. J. Salkind, editor. *Encyclopedia of Measurement and Statistics*. SAGE Publications, Inc, 2007.

[8] M.-L. Shyu, S.-C. Chen, Q. Sun, and H. Yu. Overview and future trends of multimedia research of content access and distribution. *International Journal of Semantic Computing*, 1(1):29–66, March 2007.

[9] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. pages 151–174, 2009.

[10] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2008.

[11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, second edition, June 2005.