# Florida International University and University of Miami TRECVID 2012

Qiusha Zhu, Dianting Liu, Tao Meng, Chao Chen, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33146, USA
{q.zhu2, d.liu4, t.meng, c.chen15}@umiami.edu, shyu@miami.edu

Yimin Yang, HsinYu Ha, Fausto Fleites, Shu-Ching Chen
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
{yyang010, hha001, fflei001, chens}@cs.fiu.edu

## Abstract

*This paper presents the framework and results of the team "Florida International University - University of Miami (FIU-UM)" in TRECVID 2012 Semantic Indexing (SIN) task [3] [13]. Four runs of the SIN results were submitted, and the summary of the four runs is as follows:*

- F_A_FIU-UM-1-brn_1*: Fusion of the results generated from three models, corresponding to the rest of the three runs.*
- F_A_FIU-UM-2_2*: SMR+KF+CAN - Subspace Modeling and Ranking (SMR) using the Key Frame-based low-level features (KF). The Concept Association Network (CAN) is applied to the ranking results to improve some poor detected concepts according to their relationship to other concepts.*
- F_A_FIU-UM-3-brn_3*: MCA+KF+SF+CAN - Multiple Correspondence Analysis (MCA) based ranking using KF features and shot-based features(SF). CAN is applied to the ranking results of this round as well.*
- F_A_FIU-UM-4_4*: LR+KF+CAN - Logistic Regression (LR) using KF features. CAN is applied to the ranking results of this round as well.*

In Runs 2, 3, and 4, each of them uses a different learning algorithm to train the model and predict testing instances. KF features are used in all these runs, but SF features are used only in Run 3. Additional training labels provided by NIST are also used in Run 3 (called "brn" in the name) as a trial. The Concept Association Network (CAN) is applied to all these runs to utilize the correlation between the concepts to improve the concepts with poor performance by the concepts with good performance. Finally, the results of these three runs are fused together to generate Run 1 as the best run. From the submission results, Run 1 does perform the best among all the four runs.

## 1 Introduction

The semantic indexing (SIN) task [14] in TRECVID 2012 project [11] aims to identify the semantic concept contained within a video shot, with the attempt to address the challenges like semantic gap, data imbalance, scalability, etc. The automatic annotation of semantic concepts within video shots can be a fundamental technology for categorization, retrieval, and other video exploitation. Research directions of semantic concept retrieval include developing robust learning methods that adapt to the increasing size and diversity of the videos, detecting low-level and mid-level features that have a high discrimination capability and fusing the information from other sources such as audio and text.

Compared to last year's SIN task, the same $346$ high-level semantic concepts are kept and used this year. However, the size of the training video collection this year is $1/3$ more than that of last year's training video collection. The participants are allowed to submit a maximum of $2,000$ possible shots for each of the $346$ semantic concepts, and the submission result is evaluated using a measure called mean extended inferred average precision (mean xinfAP) [17].

This paper is organized as follows. Section 2 describes our proposed framework and the specific methods used in each run. Section 3 shows the submission results in details. Section 4 concludes this paper and proposes some future directions.

## 2 Semantic Indexing (SIN)

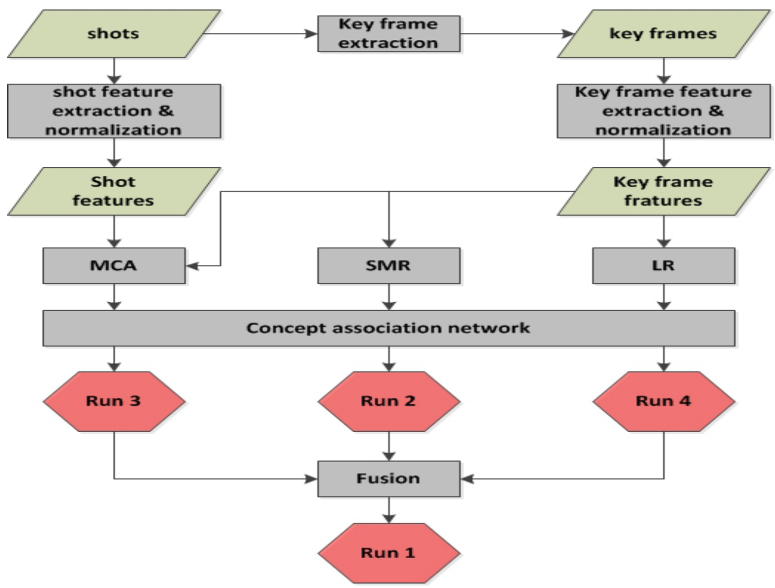



**Figure 1. The whole framework for semantic indexing**

Our overall framework of TRECVID 2012 SIN task is shown in Figure 1. As can be seen from this figure, both the shot level features (SF) and key frame level features (KF) are extracted and normalized. Runs $2$ and $4$ are generated using KF alone from SMR and LR, respectively; while Run $3$ is generated from MCA-based ranking using both KF and SF. Results from the $3$ models are fused to get Run $1$ based on the evaluation using xinfAP. Specifically, we select the ranking results for each concept based on the Top-2000 xinfAP values for that specific concept from Runs $2$ to $4$. The xinfAP values are calculated from the models trained on TRECVID 2011 training data and evaluated on TRECVID 2011 testing data.

### 2.1 Data Pre-processing and Feature Extraction

In both training and testing videos, one key frame per shot is provided to SIN task participants. The key frame delivers certain information of the shot, but obviously not sufficient to present the whole content of the shot. Considering the limited information in the key frame, supplemental frames are extracted as complementary information from the shot for the training purposes. The number of supplemental frames extracted from each training shot depends on the length of the shot. The maximum number of supplemental frames extracted from a shot is four, while if the length of the video is smaller than one second, no supplemental frame is extracted. In the testing phase, supplemental frames are not used.

Ten KF features are extracted from each extracted frame in the training and testing frames, including color histogram in the HSV space, color moment in the YCbCr space [15], canny edge histogram, sobel edge histogram, texture co-occurrence, color and edge directivity descriptor (CEDD) [4], histogram of oriented gradients (HOG) [6], haar wavelets [16], Gabor wavelets [7], and local binary patterns (LBP) [10]. Histogram equalization is employed to adjust the contrast of frames before extracting the features.

Besides KF features, we also consider motion features based on optical flow analysis to improve the detection performance of motion-related concepts, such as "Airplane Flying" and "Running". Optical flow is characterized by its global descriptive capability and is usually used for modeling the temporal dynamics of moving objects. First, five keyframes are extracted around the original key frame at an interval of $0.2$ second. Next, the $30$-bin Histogram of Oriented Optical Flow (HOOF) [5] features are calculated for each consecutive pair of keyframes. Finally, a $30 \times 4$ dimensional motion feature vector is constructed for each shot.

### 2.2 Subspace Modeling and Ranking

In Run $2$, subspace modeling and ranking (SMR) proposed in [12] is utilized to train and rank the testing data. For each concept, the training data set is first split into a positive subset and a negative subset. The positive subset is made up of positive instances; whereas the negative subset consists of negative instances. Two subspace models are built from the two subsets separately. First, the z-scores normalization (as shown in Equations (1) and (2)) is applied to the positive subset and the negative subset, respectively.

$$PosX = \frac{X^{pos} - \mu^{pos}}{\sigma^{pos}}; \quad (1)$$

$$NegX = \frac{X^{neg} - \mu^{neg}}{\sigma^{neg}}, \quad (2)$$

where $\mu^{pos}$ and $\sigma^{pos}$ values are the sample mean value and standard deviation of the positive training instances and $\mu^{neg}$ and $\sigma^{neg}$ values are the sample mean value and standard deviation of the negative training instances. Then, Singular Value Decomposition (SVD) is used to derive the Principal Components (PCs) and eigenvalues of the normalized positive instances (denoted by *PosX*) and those of the normalized negative instances (denoted by *NegX*) from their covariance matrix *CovPosX* (see Equation 3) and *CovNegX* (see Equation 4), respectively.

$$CovPosX = \frac{1}{m_{pos}} PosX^T \cdot PosX; \quad (3)$$

$$CovNegX = \frac{1}{m_{neg}} NegX^T \cdot NegX, \quad (4)$$

where $m_{pos}$ and $m_{neg}$ are the numbers of positive instances and negative instances, and $PosX^T$ and $NegX^T$ are the transpose of $PosX$ and $NegX$, respectively. Equation (5) shows how SVD is applied to *CovPosX* with the eigenvalues $\lambda_1^{pos} \geq \lambda_2^{pos} \geq \cdots$.

$$CovPosX = U_{pos} \Sigma_{pos} V_{pos}^*. \quad (5)$$

Here, $U_{pos}$=$\{PC_1^{pos}, PC_2^{pos}, ...\}$ and the diagonal value of $\Sigma_{pos}$ is $\{\lambda_1^{pos}, \lambda_2^{pos}, ...\}$. $U_{neg}$ and $\Sigma_{neg}$ can be derived in the same manner. Those PCs attached to zero eigenvalues are discarded since they contain no extra information.

A subspace spanned by $U_{pos}$ is built for the positive training instances and likewise a subspace spanned by $U_{neg}$ is built for the negative training instances. The two subspaces as well as those related eigenvalues are used in the testing phase for each testing instance.

In the testing phase, each testing instance $X_i$ goes through the normalization step using the pairs ($\mu^{pos}$, $\sigma^{pos}$) and ($\mu^{neg}$, $\sigma^{neg}$) (see Equation (1) and Equation (2)) to get $PosX_i$ and $NegX_i$. Then, $PosX_i$ is projected to the subspace spanned by $U_{pos}$, and $NegX_i$ is projected to the one spanned by $U_{neg}$, as shown in Equation (6) and Equation (7).

$$Y_s^{pos} = PosX_i \cdot PC_s^{pos}; s \in [1, \text{\# of PCs in } U_{pos}] \tag{6}$$

$$Y_t^{neg} = NegX_i \cdot PC_t^{neg}; t \in [1, \text{\# of PCs in } U_{neg}] \tag{7}$$

The distance measures shown in Equation (8) and Equation (9) are used to calculate the distances of the projected data from Equation (6) and Equation (7) to the positive and negative models.

$$DisX_i^{pos} = \sum_s \frac{Y_s^{pos} \cdot Y_s^{pos}}{\lambda_s^{pos}}; s \in [1, \text{\# of PCs in } U_{pos}] \tag{8}$$

$$DisX_i^{neg} = \sum_t \frac{Y_t^{neg} \cdot Y_t^{neg}}{\lambda_t^{neg}}; t \in [1, \text{\# of PCs in } U_{neg}] \tag{9}$$

The idea behind these distance measures is that an instance fits to a model if the distance calculated from the model is small. Based on this idea, a ranking strategy is proposed in Equation (10).

$$SCORE_i = \frac{DisX_i^{neg} - DisX_i^{pos}}{DisX_i^{neg} + DisX_i^{pos}}. \tag{10}$$

This ranking strategy implies that an instance closer to the positive learning model than to the negative learning model must have a larger possibility to belong to the positive class. Therefore, for a testing instance, the higher it holds a $SCORE$ value, the closer it is towards the positive model. Therefore, it should get a higher rank.

### 2.3 MCA-based Ranking

In Run 3, both KF features and SF features are early fused together to train the ranker since the MCA-based ranking algorithm we developed processes one feature at a time. Therefore, increasing the dimension of the feature space only increases the complexity of the learning algorithm linearly. [8] introduced the main MCA technique; whereas the MCA-based ranking algorithm is extended from it. After generating the angles between each feature and the class, Equation (12) is adopted to calculate the weight between an interval $j$ of a feature $F$ and the positive class $C^1$, where $a_F^j$ is the angle between interval $j$ of feature $F$ and the positive class $C^1$ in the projected space. Here, $a_F^j$ indicates the correlation between the interval and the positive class, and a smaller angle indicates a larger correlation. The squared cosine value of the angle is usually used to measure the quality of the correlation, which serves as the weight of a feature interval to the positive class in our framework. The ranking score of an instance $SCORE_i$ can be calculated by summing all the weights of its feature intervals. If $|F|$ is the total number of features, and $m$ is the total number of instances, $SCORE_i$ is calculated by Equation (12), where $weight_F^j$ is the weight of the interval into which instance $X_i$ of feature $F$ falls.

$$SCORE_i = \sum_{F=1}^{F=|F|} weight_F^j; \tag{11}$$

$$where \quad weight_F^j = (\cos(a_F^j))^2. \tag{12}$$

### 2.4 Logistic Regression Model

Run 4 is executed using the logistic regression model. The general idea of this model is to maximize the logarithm likelihood of the training data instances. Specifically, assume that the features of instance $X_i$ ($1 \leq F \leq |F|$, where $|F|$ is the total number of features) form a vector $\boldsymbol{s}^{(i)}$ and the weight vector is $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{|F|+1}]^T$. The likelihood that instance $X_i$ is positive is given by Equation (13). In the training data, if the positive instance is assigned the label $1$ and the negative instance is assigned the label $0$, a cost function could be defined in Equation (14). The weight vector $\boldsymbol{\theta}$ could be learned by minimizing the cost function using the gradient descent algorithm. The updating rule is given by Equation (15).

$$
\begin{aligned}
g_{\boldsymbol{\theta}}(\boldsymbol{s}^{(i)}) &= \frac{1}{1+\exp(-h_{\boldsymbol{\theta}}(\boldsymbol{s}^{(i)}))}; \quad (13)\\
where \quad h_{\boldsymbol{\theta}}(\boldsymbol{s}^{(i)}) &= \boldsymbol{\theta}^T \boldsymbol{s}^{(i)}\\
J(\boldsymbol{\theta}) &= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log\left(g_{\boldsymbol{\theta}}(\boldsymbol{s}^{(i)})\right)\\
&+ (1-y^{(i)}) \log(1-g_{\boldsymbol{\theta}}(\boldsymbol{s}^{(i)}))] \quad (14)\\
\boldsymbol{\theta}_q &\leftarrow \boldsymbol{\theta}_q - \delta \frac{\partial}{\partial \boldsymbol{\theta}_q} J(\boldsymbol{\theta}). \quad (15)
\end{aligned}
$$

Here, $\delta$ is the learning rate, $m$ is the total number of training data instances, $\boldsymbol{\theta}_q$ is the $q^{th}$ element of the vector $\boldsymbol{\theta}$ and $y^{(i)}$ is either 1 or 0 indicating the data instance is positive or negative. For a testing instance, the likelihood that the instance is positive is computed using Equation (13) by plugging in the feature values and the trained parameter $\boldsymbol{\theta}$.

### 2.5 Concept Association Network

Since the concepts do not occur independently in the TRECVID 2012 data sets and they have some correlations, the concept association network [9] is utilized in this project to model the correlations among different concepts. The proposed framework is shown in Figure 2(a) (training phase) and Figure 2(b) (testing phase). The training phase consists of the *Concept Based Classifiers Training Component* and the *Concept Association Network Training Component*. The former is the architecture of the concept detection framework proposed in our work. For example, in a training data set, there are $m$ instances and $n$ high-level ($n = 346$) concepts to detect. The training instances are preprocessed and a set of features are extracted. Afterwards, $n$ binary content-based classifiers such as the subspace-based models or the MCA-based classifiers are trained for $n$ concepts, so that each model $k$ ($1 \leq k \leq n$) outputs $m$ scores for the $k^{th}$ concept, represented by $C_k$ in the figure. The *Concept Association Network Training Component* receives the scores from the *Concept Based Classifiers Training Component* and discovers the frequent itemsets in the label matrix to build a Concept Association Network (CAN). The detailed steps of building the CAN are introduced as follows.

First, all the labels of the training instances are organized into a label matrix. Specifically, the labels of all the $m$ instances for the $n$ high-level concepts are organized into a label matrix $\boldsymbol{L} = \{l_{ik}\}$, $i = 1, 2, ..., m$ and $k = 1, 2, ..., n$, where $l_{ik} = C_k^1$ or $l_{ik} = C_k^0$ indicates the $i^{th}$ instance is labeled as positive or negative for the $k^{th}$ concept. Table 1 shows an example of a label matrix.

Next, the association links among different concepts are generated by mining the significant rules from the label matrix. The Apriori algorithm [1] is applied to the label matrix to discover the association rules. The specific algorithm to generate all the 2-item rules works as follows. First, all 1-$itemsets$ are generated for $\boldsymbol{L}$. Only the 1-$itemsets$ $\{C_k^1\}$ consisting of positive concept-class pairs are retained. Second, all the candidate 2-$itemsets$ are generated by combining the 1-$itemsets$ with a minimum support of one. Afterwards, the candidate 2-$itemsets$

**Table 1. Label matrix**

| Instance | $C_1$ | $C_2$ | ... | $C_k$ | ... | $C_n$ |
|---|---|---|---|---|---|---|
| Instance 1 | $C_1^0$ | $C_2^1$ | ... | $C_k^1$ | ... | $C_n^0$ |
| Instance 2 | $C_1^0$ | $C_2^1$ | ... | $C_k^0$ | ... | $C_n^1$ |
| ... | ... | ... | ... | ... | ... | ... |
| Instance $i$ | $C_1^1$ | $C_2^0$ | ... | $C_k^0$ | ... | $C_n^1$ |
| ... | ... | ... | ... | ... | ... | ... |
| Instance $m$ | $C_1^0$ | $C_2^1$ | ... | $C_k^1$ | ... | $C_n^0$ |

which contain the concept of interest are organized together. Based on these $2$-$itemsets$, a set of candidate rules which draw the conclusion that the concept of interest is positive are generated. In order to select the most significant rules, two rule pruning modules are incorporated into the framework.

Currently, only binary co-occurrences relationships between concepts are considered. Therefore, only the 2-item rules are generated. In order to retain the most significant rules, the support ratio and the interest ratio are used to select rules. Formally, let $C_t$ represent the target concept which is the concept of interest and $C_r$ represent the reference concept which is the concept used to help the detection of the target concept; and let $sup(X)$ represent the support value of the itemset $X$. The support ratio ($R_s$) and interest ratio ($R_i$) are defined in Equation (16) and Equation (17), respectively. Intuitively, these criteria represent rule selection from the target concept point of view and the reference concept point of view. The theoretical justification of these rules is in [9]. In addition, the thresholds for these two ratios are determined using the cross validation process.

$$R_s = \frac{sup(\{C_t^1, C_r^1\})}{sup(\{C_t^1\})}. \tag{16}$$

$$R_i = \frac{sup(\{C_t^1, C_r^1\})}{sup(\{C_t^1\}) \times sup(\{C_r^1\})}. \tag{17}$$

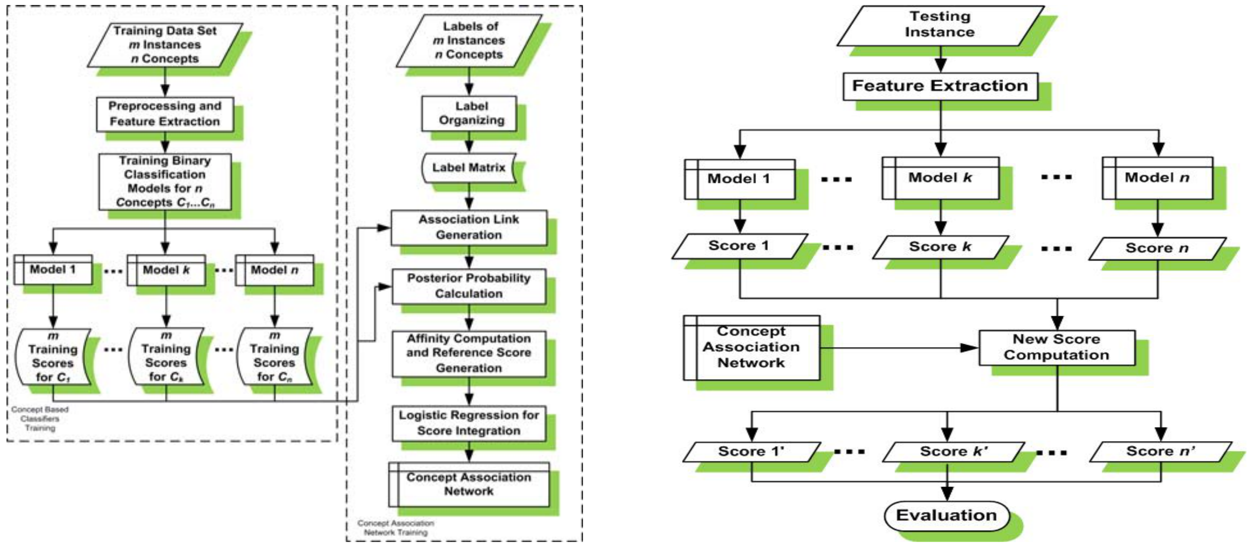
From the network point of view, if all the relationships among concepts are modeled in a network $G$=$\{V, A, W\}$, where $V$ is a set of nodes with each node representing a concept, $A$ represents a set of links, and each link has a weight in set $W$ to model the relationship between two nodes. The selected significant rules could be viewed as the significant links from the reference concepts to the target concept. These links are defined as the association links and form the core of the concept association network.

Because the output scores of different models could fall into different ranges, the raw scores are preprocessed to feed into the concept association network. In addition, the information of the credibility of the score is not included in the raw score. Therefore, the raw scores are converted to the probability-based scores using the Bayes Rule. Assuming for an instance $i$, the detection score of $C_k$ is $O(k, i)$. The output score $O'(k, i)$ for $C_k$ which encompasses the information of the credibility of the model is given in Equation (18).

$$O'(k,i) = \frac{p(O(k,i)|C_k = 1) \times p(C_k = 1)}{\sum_{z=0}^{1} p(O(j,i)|C_k = z) \times p(C_k = z)}, \tag{18}$$

where $p(C_k = 1)$ is the prior probability of $C_k$ appeared in a data instance and is estimated by dividing $sup(\{C_k^1\})$ by the total number of training instances. $p(C_k = 0)$ is one minus $p(C_k = 1)$ because there are only two possible cases. $p(O(k,i)|C_k = 1)$ is the conditional probability density function (pdf) $f_P(x) = p(x|C_k = 1)$ evaluated at $x = O(k,i)$, and $p(O(k,i)|C_k = 0)$ is the conditional pdf $f_N(x) = p(x|C_k = 0)$ evaluated at $x = O(k,i)$. To estimate $f_P(x)$ and $f_N(x)$, the Parzen-Window approach [2] is employed here.

The last step is to integrate the posterior probability scores from the reference concepts and the target concept properly to generate the final score. This process is the fusion process which is very important for the overall performance of the framework. In this study, the logistic regression model is utilized to fuse the outputs together.




(a) Training phase of the proposed framework

(b) Testing phase of the proposed framework

**Figure 2. CAN framework**

**Table 2. The MAP values at first $n$ shots for all four runs**

| n | 10 | 100 | 1000 | 2000 |
|---|---|---|---|---|
| *F_A_FIU-UM-1_1* | 45.7% | 30.6% | 17.2% | 14.2% |
| *F_A_FIU-UM-2_2* | 45.4% | 28.9% | 16.4% | 13.8% |
| *F_A_FIU-UM-3_3* | 37.8% | 25.4% | 15.2% | 12.9% |
| *F_A_FIU-UM-4_4* | 35.9% | 27.5% | 17.3% | 14.8% |

The details could be found in [9] and the weights for the links are learned using the cross validation procedure. After this step, the concept association network with all the learned weights is built using the training instances.

In the testing phase, the same set of features as in the training phase is first extracted. For each testing instance, it receives one score from each content-based classifier. By leveraging the concept association network, for a target concept, a new score which integrates the information from reference concepts is generated as the final output score.

## 3 Experimental Results

The whole framework of TRECVID 2012 SIN task contains three stages:

1. Model training: use TRECVID 2011 training videos as training data.

2. Model evaluation: use TRECVID 2011 testing videos as testing data to evaluate the framework and tune the parameters of the models.

3. Model testing: use TRECVID 2011 training + TRECVID 2011 testing videos as TRECVID 2012 training data, and TRECVID 2012 testing videos as testing data to generate the ranking results for submission.

Figure 3 to Figure 6 show the performance of our semantic indexing results. More clearly, Table 2 shows the mean average precision (MAP) values of the first 10, 100, 1000 and 2000 shots. The inferred true shots and mean xinfAP are shown in Table 3.

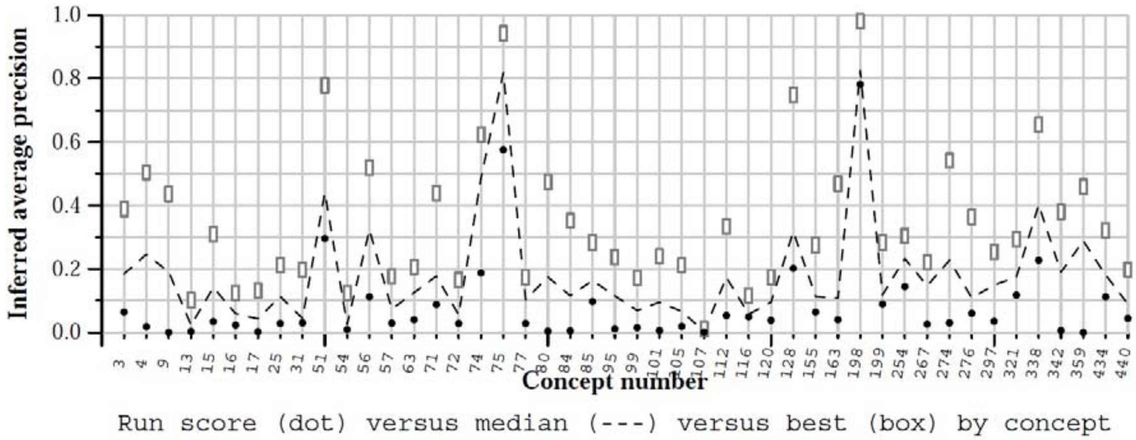



**Figure 3. Run scores (dot) versus median (—) versus best (box) for** *F_A_FIU-UM-1-brn_1*

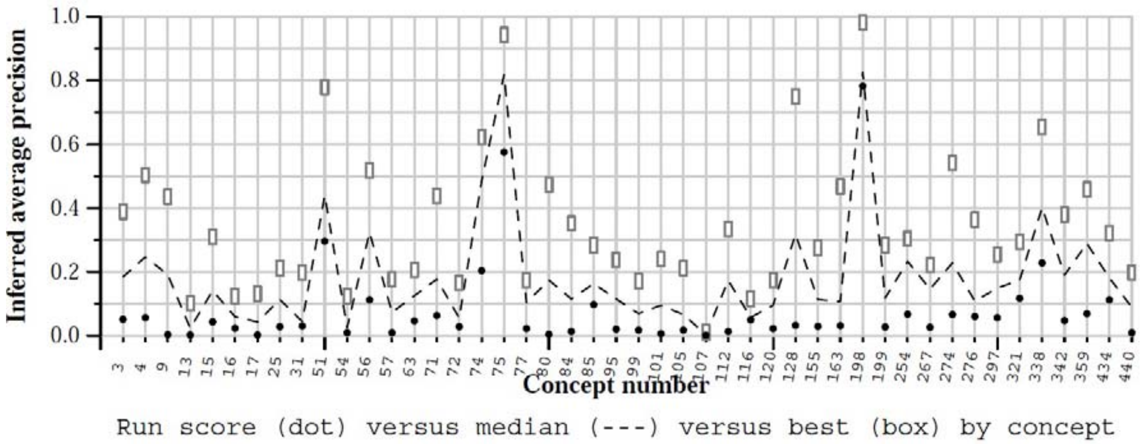



**Figure 4. Run scores (dot) versus median (—) versus best (box) for** *F_A_FIU-UM-2_2*

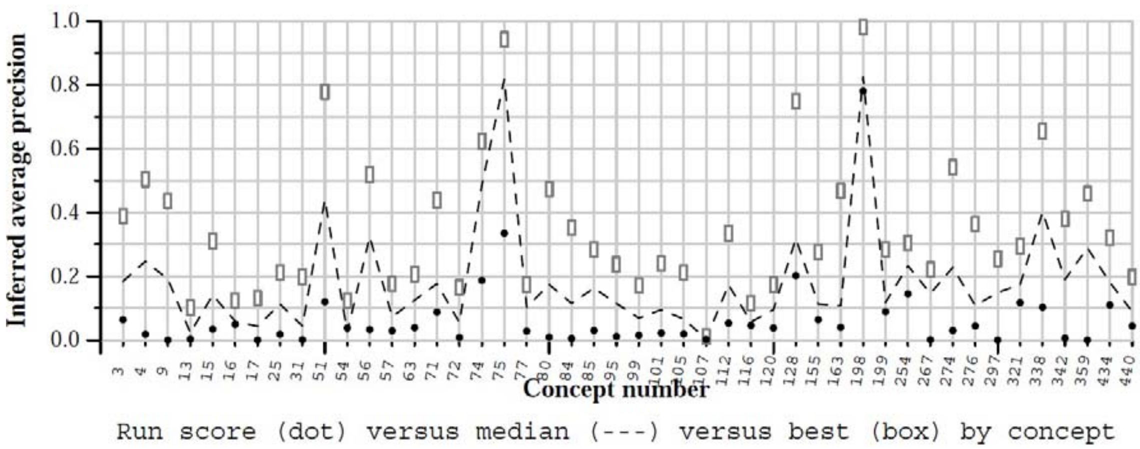



**Figure 5. Run scores (dot) versus median (—) versus best (box) for** *F_A_FIU-UM-3-brn_3*

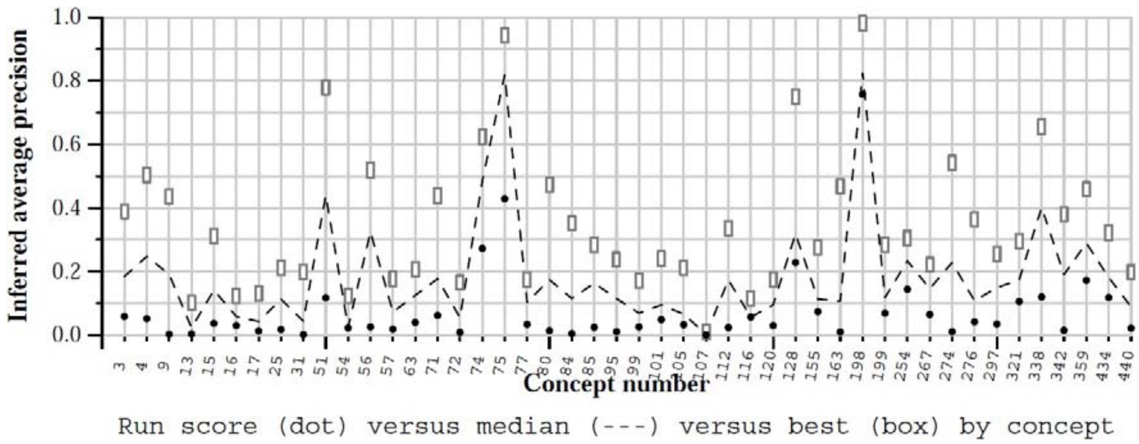



**Figure 6. Run scores (dot) versus median (—) versus best (box) for** *F_A_FIU-UM-4_4*

**Table 3. Inferred true shots and mean xinfAP**

| | Inferred true shots | Mean xinfAP |
|---|---|---|
| *F_A_FIU-UM-1_1* | 13049 | 0.084 |
| *F_A_FIU-UM-2_2* | 12721 | 0.079 |
| *F_A_FIU-UM-3_3* | 11897 | 0.068 |
| *F_A_FIU-UM-4_4* | 13597 | 0.076 |

Evaluation results show that similar results from all four runs, but Run 1 which fuses the results from the rest of three runs performs slightly better than the rest under different criteria. Based on the whole process of the task, we have the following insights:

- The model performance is proportional to the number of features, but it increases slower and slower after a certain number of features.

- Increasing the number of positive instances such as extracting more frames from positive shots can noticeably improve the model performance since for many concepts, the positive to negative ratio is very low, which is also referred as the data imbalance issue.

## 4 Conclusion and Future Work

In this notebook paper, the framework and results of team FIU-UM in TRECVID 2012 SIN task is summarized. From the results, we can see there are still a lot of improvements to be done. Some important directions need to be investigated:

- The current features in our framework are global features, so the object-level and mid-level features need to be explored.

- From the experiment, we have seen that by adding more positive training instances, the model performance has noticeably improved. Thus, other methods will be introduced to solve the data imbalance issue in order to improve the model performance.

- Although three different learning algorithms are adopted in our framework, further improvements need to be made to the current learning algorithms or other algorithms such as support vector machine (SVM) should be investigated.

## References


[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, September 1994.

[2] C. Archambeau, M. Valle, A. Assenza, and M. Verleysen. Assessment of probability density estimation methods: Parzen window and finite Gaussian mixtures. In *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, pages 3245–3248, May 2006.

[3] S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In *European Conference on Information Retrieval (ECIR)*, pages 187–198, Glasgow, Scotland, mar 2008.

[4] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Proceedings of the 6th international conference on Computer vision systems*, ICVS'08, pages 312–322, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, Oct. 1996.

[8] 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, Dec. 2008.

[9] T. Meng and M.-L. Shyu. Leveraging concept association network for multimedia rare concept mining and retrieval. In *IEEE International Conference on Multimedia and Expo (ICME12)*, pages 860–865, July 2012.

[10] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[11] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[12] M.-L. Shu, C. Chen, and S.-C. Chen. Multi-class classification via subspace modeling. *International Journal of Semantic Computing*, 5(1):55–78, 2011.

[13] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[14] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In A. Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[15] S. Sural, G. Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *in International Conference on Image Processing (ICIP). 2002: p. 589-592. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin*, pages 589–592, 2002.

[16] D. Verma and V. Maru. An efficient approach for color image retrieval using haar wavelet. In *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, pages 1–5. IEEE, 2009.

[17] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 603–610, New York, NY, USA, 2008. ACM.