University of Central Florida at TRECVID 2007 Semantic Video Classification and Automatic Search

Jingen Liu, Yusuf Aytar, Bilal Orhan
Jenny Han, Mubarak Shah

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida 32816, USA

ABSTRACT

In this paper, we describe our approaches and experiments in semantic video classification (high-level features extraction) and fully automatic topic search tasks of TRECVID 2007. We designed a unified high-level features extraction framework. Two types of discriminative low level features, Spatial Pyramid Edge/Color Histograms and Bag of Visterms, are extracted from the key-frames of the shots. Then the SVM classifiers with RBF kernel are used for classification. The final classification results are produced by fusing and combining these classifiers. The experiment results show that the combined classifiers substantially improved the performance over the individual feature based classifier. In fully automatic topic search task, we mostly focus on the video retrieval using the visual content through the high level features detectors. The main challenge in this task is mapping queries to the high level features. A novel earth mover's distance (EMD) based relevance procedure that finds the similarity between queries and videos through the high level features and semantic word similarity measures.

1. INTRODUCTION

This year, the Computer Vision Lab team at University of Central Florida participated in the high-level features extraction and fully automatic topic search tasks. We submitted six runs for high-level features extraction and six runs for automatic topic search. The returned evaluation results show that our approaches achieved reasonable results. Some of the runs had much better performance to the median results of the whole evaluation pool.

1.1. High-Level Feature Extraction

In the highe-level feature extraction task, we extracted two types of discriminative low-level features (spatial pyramid edge/color histogram and bag of visterms) and text feature extracted from ASR information using co-clustering. Then we trained the SVM classifiers on the features separately. On testing phase, unlike last year's TRECVID, we used multiple keyframes from each video shot. As we observed, the uniform sampling is enough to capture the variance of frames in one shot. The classified result on each keyframe is produced by fusing the classifiers trained on different low-level features. Finally, the classification score for one shot is computed from the average or maximization scores of all the keyframes. For every high-level feature our main steps are as follows:

- Extract low-level features;
- Train a classifier using different low-level feature independently;
- Combine the classifiers using training-based and non-training based approaches.

We submitted the following six runs in the high-level features extraction task:

- **A_UCF.W.PROD.ASR**: "weighted" product fusion of the classifiers using low-level features and text features (the weights for different low-level features are trained on training data set). The final score of one shot is the average of scores of the keyframes;

- **A_UCF.W.PROD.MEAN**: this run only use visual features compared to run **A_UCF.W.PROD.ASR**;
- **A_UCF.W.PROD.MAX**: "weighted" product fusion of the classifiers using the low-level features, and the final score of each shot is the max score of the keyframes;
- **A_UCF.W.AVERAGE**: "weighted" average fusion of the classifiers using the low-level features. Final score of each shot is the mean value of the scores on the keyframes;
- **A_UCF.PROD**: direct product fusion of the output of the classifiers using low-level features. Final shot score is the average value of the keyframes.
- **A_UCF.PROD.0607**: the development data set for this run is the combination of the development data of TRECVID 2006 and 2007. The fusion method is direct product.

Based on the evaluation results, those runs which were trained on TRECVID 2007 development data and fused with different approaches achieved very close performance in terms of mean average precision. When looking into the fusion among the keyframes, the MAX fusion (run **A_UCF.W.PROD.MAX**) performs better than AVERAGE fusion (run **A_UCF.W.PROD.MEAN**). Although we got very good performance in validation phase using the combination of text feature and visual feature, the evaluation results show that the combination of text and visual features (run **A_UCF.W.PROD.ASR**) actually decrease the performance. We guess the classification based on text might be very sensitive to the data set. Besides, we expected to make the system more robust and efficient by combining the development data of TRECVID 2006 and 2007. However, this combination decrease the performance a lot. This can be another example to show that training-based classification is very sensitive to the difference of the content of the training and testing data set. Comparing run **A_UCF.W.PROD.MEAN** with run **A_UCF.W.PROD**, we can see the "weighted" product fusion of low-level features performs a little bit worse than non-training based product fusion.

1.2. Automatic Topic Search

This year for the search task we mainly focused on using visual content using a large number of concept detectors. Each concept detector is trained for a particular concept, and given a video shot it returns the confidence value about existence of that concept. Our approach is mainly composed of three steps. The first step is to find a proper representation of shots and queries. Since the visual content and query are two distinct forms of information, it is important to find suitable representations for each of them. In our system, each video shot is represented by a histogram in terms of concepts present in the shot and their confidence values extracted using the concept detectors. Similarly, each query is represented by another histogram in terms of query words and their information content. The next step is to compute the relevance between query and video shots using these histograms. Sine these two histograms are in two different spaces in order to compute the relevance between two histograms we have developed an Earth Movers Distance (EMD) based relevance metric. In order to compute the distance between any two histograms using EMD we need to assign distances from each bin in the first histogram to the each bin in the second histogram. Finally, the video shots are sorted based on their relevance to the given query in descending order and retrieved in this sequence.

We submitted six runs on fully automatic topic search tasks. They are listed as follows:

- **F_A_1_UCFVISION1**: we used text information with normalized text overlapping approach (MAP = 0.0052).
- **F_A_1_UCFVISION2**: automatic search using text information only by normalized text overlapping with stemming approach (MAP=0.0053).
- **F_A_1_UCFVISION3**: automatic search using visual features with semantic EMD approach. We used Vireo feature detectors provided by City University of HongKong (MAP = 0.0314).
- **F_A_1_UCFVISION4**: automatic search using visual features and text information with semantic EMD approach. We also used the Vireo detectors in this run (MAP = 0.0220).

- **F_A_1_UCFVISION5**: this run is similar to run **F_A_1_UCFVISION3**, but we used the feature detectors provided by Columbia University.
- **F_A_1_UCFVISION6**: the difference between this run and run **F_A_1_UCFVISION4** is we used the feature detectors provided by Columbia University in this run.

Based on the returned evaluation results, the run which only used visual features achieved the best performance. The results using the feature detectors provided by Columbia University are abnormal. We found we made a mistake when we were running on their features. However, these results using Vireo detectors are enough for us to valuate our approach. As we see, our visual content based method is based method is 500% better than our text baseline. The combination of text and visual features could not help the search. This is same to our high level feature extraction using both visual and text features. Besides, we also evaluated our method on TRECVID 2006 testing dataset. Our visual based approach performed 80% better than the text baseline.

2. HIGH-LEVEL FEATURE EXTRACTION

In TRECVID 2007, we developed a unified high-level features extraction framwork. There are three main steps involved.

- Low-level feature extraction. We computed two types of visual features: spatial pyramid edge/color histogram and bag of visterms. Also, we used co-clustering to capture the text feature.
- Model training and selection. We adopted SVM as our classification method. First, we trained the individual SVM classifiers for each low-level feature. Then, non-training based model fusion and training-based model fusion were performed to combine the models learnt using single low-level feature.
- Apply the combined SVM classifiers to the TRECVID 2007 testing dataset.

2.1. Spatial Pyramid Edge/Color Histogram (SPEH/SPCH)

Let us look into how to measure the similarity of two images which are represented by bag of features. In general, we can simply generate the histogram of the features in the image, and then the similarity of the two histograms is considered as the similarity measurement. Actually, the histogram is the global distribution of the features. If we also want to capture the local distribution of the features, we can divide the image into sub-blocks like what people did in previous image classification and retrieval. Then, each block is somehow incorporated with spatial information. For instance, the feature from the sky normally occur at the top of the image, therefore the top sub-blocks will capture the most sky information, while the bottom sub-blocks contains less sky features. In this way, the feature matching will be more localized. Normally, this is called grid-based features.

Grid-based features might work better compared to global features. However, because it is localized the information, it can not handle rotation or transposition problem. It is true that at most time the features will be localized at the fixed relative position of the image. While some features did not have fixed feature location, for those features the global features probably works better. Hence, it will be wonderful if we can combine both method and let the classifier to figure out when to use the global information. This is the main idea of Pyramid Match Kernel. The feature can be either visterms[6] or other features like quantized edge feature and color feature.

Basically, we can divide the image into multiple level of pyramid. For instance, the global image is the bottom level of the pyramid, and then further divide the image to 2 by 2 sub-blocks which is the second level of the pyramid. By keeping dividing the image like this, we can represent the image with a multiple level of pyramid. When doing image match, we have to look all the feature distribution in different sub-blocks at different level. In fact, the match of features at different level can be thought as matching from coarse (global level) to fine (higher level). It is very straightforward to think the matching made at fine level is more important than the matching at coarse level. Therefore, the matching made at higher level is assigned higher weight.

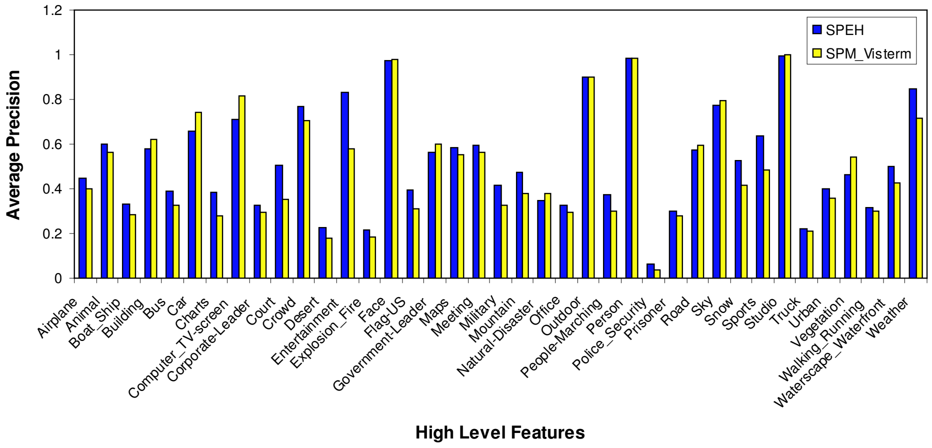


Figure 1. The performance comparison (in term of average precision) of SPM with visterms and SPEH.

Suppose the matching of features is measured by histogram intersection, we can evaluate the similarity at level l as follows,[7]

$$\mathcal{I}(H_X^l, H_Y^l) = \sum_{i=1}^{D} min(H_X^l, H_Y^l) \tag{1}$$

where H_X^l represents the histogram of the features at level l for image X, and D is the dimension number. Assume the weight assigned to level l is $\frac{1}{2^{L-l}}$, and remove the match already made at lower level, the match kernel between two images can be[7]:

$$\mathcal{K}^L(X,Y) = \mathcal{I}^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}}(\mathcal{L}^l - \mathcal{L}^{l+1}) = \frac{1}{2^L}\mathcal{L}^0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}}\mathcal{L}^l. \tag{2}$$

Actually, the weight can be incorporated into the histogram before histogram intersection. Thus, the procedure is simplified as computing weighted histogram by pyramid level.

Both PMK[9] and Spatial Pyramid Match (SPM)[7] which is the application of PMK have been used in object classification and scene classification. Also, very promising performance has been achieved. Most of the work used visterms (group of patches) as the visual features. This works well for small dataset, but working on large dataset like TRECVID data set, it can not achieve good performance. We conjecture there are two main reasons. First, it is hard to get meaningful visterms using K-means on large data set because of the sampling may be not representative. Second, normally we have to extract thousands of visterms to get good performance. This will make the feature generated by SPM has a large number of dimensionality. This affects both the performance and computation efficiency. In stead of using visterms, edge histogram and color histogram were used in our experiments. We called them as spatial pyramid edge/color histogram. For edge histogram, we quantized the edge into 8 bins in direction and 8 bins in gradient magnitude. Also, we quantized the color into 20+20+20 in HSV space. Compared to visterms, the dimension is low and no complicated quantization is required.

We did experiments on TRECVID 2006 development data set to verify our conjecture. We divided the development data set into two parts. Three fourthes are used for training, and one fourth is used for validation. From the training dataset, we extracted 1,500 visterms following the process described in section 2.2. For the SPM with visterms, we used two levels, and the number of dimension is 9,000. While for the SPEH, we used three pyramid levels, and the number of dimension is 1344. Fig. shows the performance comparison between these two types of features. They are competitive. However, SPEH is 3.7% better than the SPM with visterms in terms of MAP.

2.2. Bag of visterms

In order to sample meaningful patches, we sampled 100 keyframes from each of the 36 categories, and 500 keyframes out of the 36 categories. Grid-base sampling technique was used with sampling space of 10, which means every 10 pixels, we extract one patch. SIFT descriptor[10] is used to represent each patch. The size of patch is varied from 10 to 30 pixels, which is randomly selected. The codebook of visterms is formed using k-means clustering. Finally, we generated 1,500 visterms from the training dataset.

2.3. Text information using co-clustering

Co-clustering[11] via the Maximization of Mutual Information (MMI) is a successful strategy to group words into semantic concept clusters (e.g. "pitching", "score","teams" etc. can be clustered into "baseball" concept and "biker", "wheel", "ride" might be clustered into "motorcycle".), which has been successfully used in text classification[11] and image classification area.[6] The critical point is to simultaneously maximize the mutual information (MI) of the words and documents when clustering these words into semantic concepts. In our experiments, we can take the video shots as documents. As we see, the words histogram of each shot is very parse, which affects the classification performance. Hence, using co-clustering to further group the words into semantic clusters can overcome this problem. We briefly describe the approach in next paragraph.

Given two discrete random variables X and Y, the MI between them is defined as:

$$I(X;Y) = \sum_{y \in Y, x \in X} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{3}$$

where $p(x,y)$ is the joint distribution of X and Y, $p(x)$ and $p(y)$ are the probability distributions of X and Y respectively.sing Kullback-Leibler divergence, also known as relative entropy, the MI also can be expressed as:

$$I(X,Y) = D_{KL}(p(x,y) \parallel p(x)p(y)), \tag{4}$$

where D_{KL} computes the distance between two distributions.Consider a training image dataset $\mathcal{Y}$ with c categories, and its associated codebook $\mathcal{X}$ with n *visterms*, we seek to simultaneously cluster Y into c categories $\hat{\mathcal{Y}} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_c}\}$, and X into w disjoint clusters $\hat{\mathcal{X}} = \{\hat{x_1}, \hat{x_2}, ..., \hat{x_w}\}$. Actually, we can consider the clustering as two mapping functions $\hat{X} = C_X(X)$ and $\hat{Y} = C_Y(Y)$. In order to evaluate the quality of clustering, we utilize the following mutual information loss:

$$\Delta MI = I(X;Y) - I(\hat{X};\hat{Y}). \tag{5}$$

Because $I(X;Y)$ is fixed for specified data collections, the optimal co-clustering actually attempts to maximize $I(\hat{X};\hat{Y})$, given the number of clusters c for Y, and w for X respectively. It is straightforward to verify that the MI loss also can be expressed in the following form[11]:

$$\Delta MI = D_{KL}\big(p(x,y) \parallel q(x,y)\big), \tag{6}$$

where $q(x,y) = p(\hat{x},\hat{y})p(x|\hat{x})p(y|\hat{y})$. This is the objective function when performing co-clustering. The input to co-clustering algorithm is the joint distribution $p(x,y)$, which records the probability of occurrence of a particular *visterm* x in a given image y. The aim is to determine clusters with distribution $q(x,y)$ which is as close as possible to $p(x,y)$.

2.4. SVM-based Training and Model Selection

Support Vector Machine (SVM)[5] was used for classification in our experiments. In the training procedure, we have two phases. First phase, we trained three SVM models for each concepts in the three feature spaces. In this phase, the development dataset was divided into two parts with two thirds for training and one third for validation. Second phase, we fused the three models for each concept. We have two ways, fusion with training and fusion without training. For fusion with training we further divided the validation dataset into two equal parts, which are used to train and validate in the fusion phase. When training the classifiers in the three visual feature space, SVMs with a Radial Basis Function(RBF) kernel are used. We noticed that the classification performance of SVMs varies with different parameters. In our experiments, we used "grid-search"[8] method to find out the proper parameter γ and C for RBF kernel. Since the dataset is very unbalanced between the number of positive and negative key-frames, we also tuned the "weight" parameter, which represents the relative significance of positive samples to negative samples. In our experiments, we set this parameter to be the ratio of negative to positive samples in the dataset.

2.5. Score Normalization and Fusion

The SVM models are separately trained from color, edge or iamge patch features. We noticed that models built using color, edge or image patch features has different performance for each individual high-level features. For instance classifiers that use color statistics achieve better performance for "sky" and "sports", while classifiers trained on edge features work better for "building" and "crowds". Therefore, it is helpful to combine the output of individual classifiers.

Before results fusion, we have to normalize the classification score returned by different classifiers. In our experiments, we used Z-score normalization method listed as follows,

$$S_{new} = \frac{S - mean}{standard\ deviation} \tag{7}$$

where S_{new} denotes the normalized score and S is the classifier output score.

In order to fuse different classification score returned by different classifiers on one keyframe, we used two type of direct fusion approaches as follows,

- Average Score: $S_{new} = \frac{\sum_{i=1}^{N} S_i}{N}$,
- Product Score: $S_{new} = \prod_{i=1}^{N} S_i$.

Actually, the above two fusion methods did not consider all the classifiers equal. However, this might not be true for most cases. So we also tried the weighted fusion. More specifically, we learn the weights assigned to each classifier of each high level feature from the training dataset. Therefore, we also have weighted average fusion and weighted product fusion method.

In the testing phase, we used multiple keyframes from the shot to represent it in stead of one or two keyframes. We used uniform sampling method to get the keyframes from the shot. As our observation, this method is enough to extracted the representative keyframes from the shots. The final classification score of one shot is computed by the "average" or "max" value of all classification scores of the keyframes.

2.6. Results and Discussion

We divided the development dataset of TRECVID into two parts. One fourthes were used as training set, and the rest were used as validation data set. We have tried several fusion methods with/without text features. Fig 2 gives the detail comparison among different fusion approaches. We can see, there is not fusion method which can works good for all the high level features. Fig. 3 shows the comparison of different fusion approach in terms of MAP. There is not much difference between the performance of "product" fusion and "average" fusion. While all the "weighted" can get better performance compared to the corresponding non-weighted fusion. In the validation data set, text information also help the deacidification.

We submitted the following six runs to this year's TRECVID:

- **A_UCF.W.PROD.ASR**: "weighted" product fusion of the classifiers using low-level features and text features. The final score of one shot is the average of scores of the keyframes;
- **A_UCF.W.PROD.MEAN**: this run only use visual features compared to run **A_UCF.W.PROD.ASR**;
- **A_UCF.W.PROD.MAX**: "weighted" product fusion of the classifiers using the low-level features, and the final score of each shot is the max score of the keyframes;

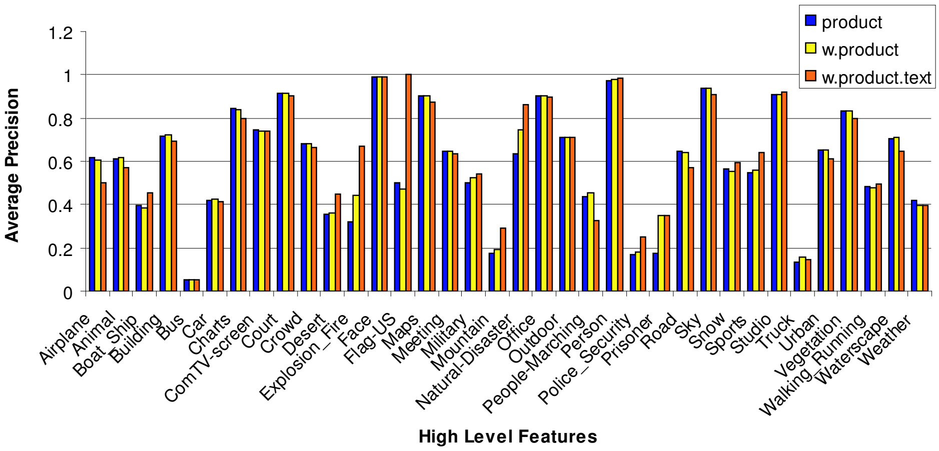


Figure 2. This figure compares the performance of three approaches in details: product fusion, weighted product fusion and weighted product fusion with text features. These classifiers were tested on the validation dataset for all 36 high level features

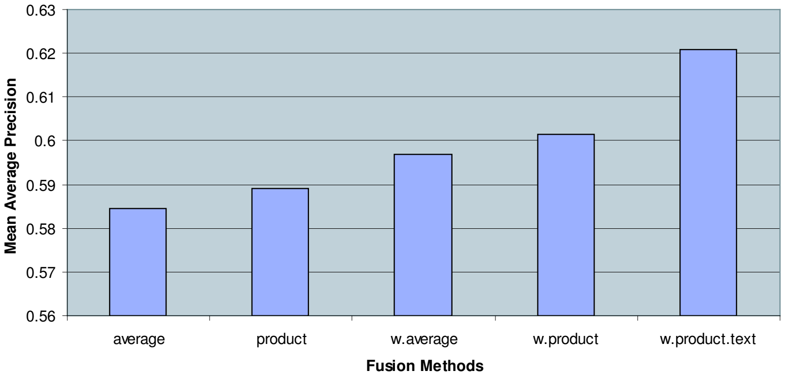


Figure 3. The performance comparison (in term of mean average precision) among different fusion approaches. The dataset is the validation dataset.

- **A_UCF.W.AVERAGE**: "weighted" average fusion of the classifiers using the low-level features. Final score of each shot is the mean value of the scores on the keyframes;
- **A_UCF.PROD**: direct product fusion of the output of the classifiers using low-level features. Final shot score is the average value of the keyframes.
- **A_UCF.PROD.0607**: the development data set for this run is the combination of the development data of TRECVID 2006 and 2007. The fusion method is direct product.

Figure 4 displays the performance of each run compared to all the runs in the TRECVID 2007. Overall, most of our runs achieved better performance compared to the median line, and some features hit or approach the best results. The comparison in term of Mean of inferred AP among all the runs is shown in Fig. 5. Compared to the performance of all the fusion methods on validation dataset, there are two main exceptions. One is the "weighted" method (run *A_UCF.W.PROD.MEAN*) perform worse than the "non-weighted" approach (*A_UCF.PROD*). Another one is unlike the validation dataset the text feature did not help the classification. When looking into the fusion among the keyframes, the MAX fusion (run *A_UCF.W.PROD.MAX*) performs better than AVERAGE fusion (run *A_UCF.W.PROD.MEAN*). Besides, we expected to make the system more robust and efficient by combining the development data of TRECVID 2006 and 2007. However, this combination decrease the performance a lot. This can be another example to show that training-based classification is very sensitive to the difference of the content of the training and testing data set.

3. AUTOMATIC TOPIC SEARCH

This year for the search task we mainly focused on visual content based retrieval using a large number of concept detectors. We also applied two text retrieval methods using ASR-MT information for comparison and multimodal fusion.

3.1. Text Based Methods

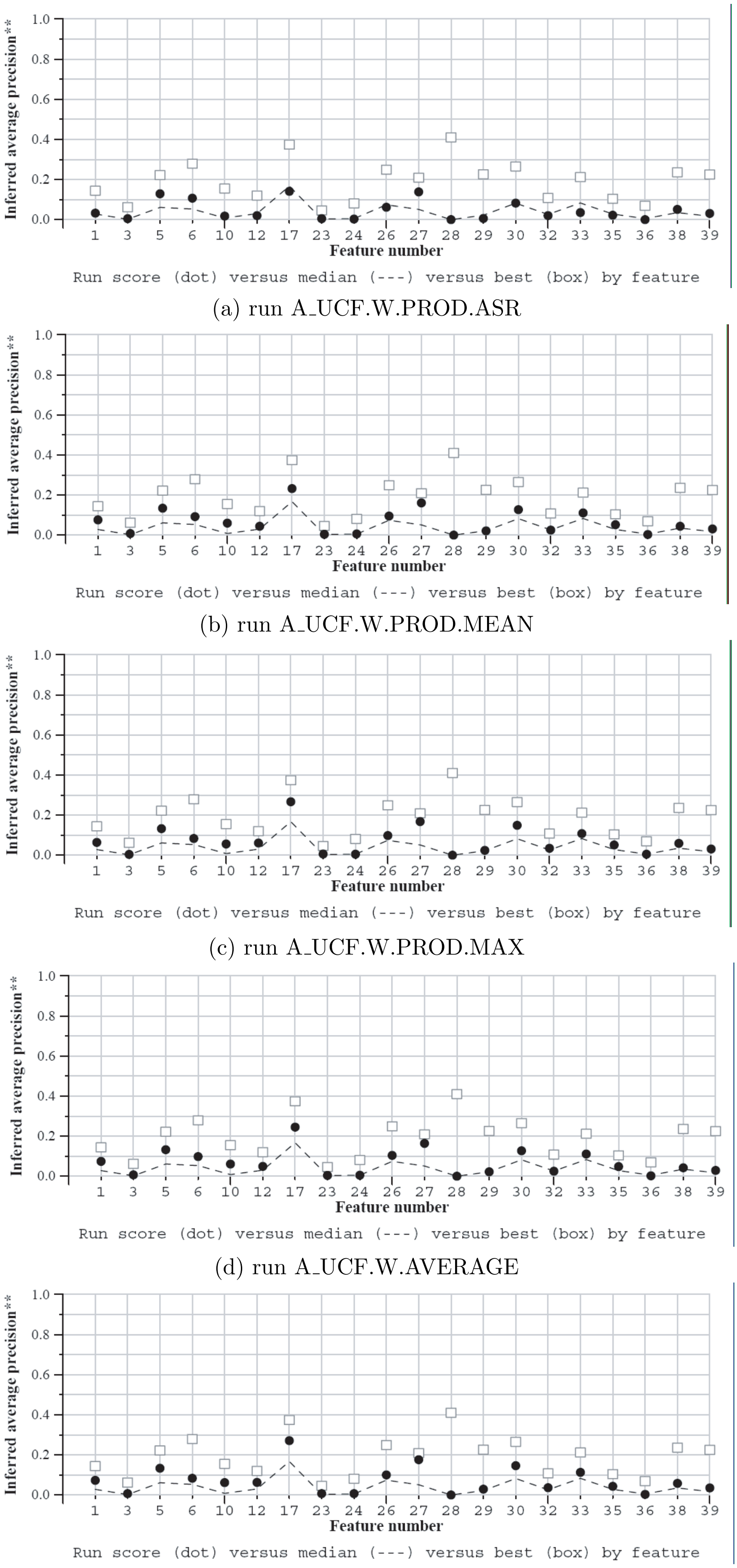
We evaluated two text based retrieval methods which use ASR-MT (Automatic Speech Recognition & Machine Translation) information[1] as text data. In order to have a stronger context, ASR-MT text for a particular shot is determined as a combination of ASR-MT information within a five shots window. Initially we removed the stop words from both queries and ASR-MT text for each shot. In our first run, $F\_A\_1\_UCFVISION1$, relevance of the shot for the given query is computed as the intersection of query words and ASR-MT words, normalized by the union of them. Additionally each word is weighted with its length. This weighting depends on the hypothesis that, in general, longer words are more likely to represent the subject of a text string than are shorter words. In our second run, $F\_A\_1\_UCFVISION2$, addition to the previous method we applied stemming for the words. This year, due to the complex nature of queries, text based methods were not as effective as visual content based methods.

3.2. Visual Content Based Methods

Visual content of the video shots are utilized through concept detectors. Each concept detector is trained for a particular concept, and given a video shot it returns the confidence value about existence of that concept. In order to be easily comparable with other approaches, we used two sets of publicly available 374 concept detectors. These are Columbia detectors released by Columbia University[3] and Vireo detectors released by City University of Hong Kong.[4]

Our approach is mainly composed of three steps (Fig.6 shows the overview of our automatic search system). Initially, the query and video shots are expressed using appropriate histograms. Then for a given query the relevance between the query and each video shot is computed. Finally, the video shots are ranked and retrieved based on these relevance scores. Most of the previous approaches that use visual concept detectors map the queries into the concept space and compute the relevance in this space. We believe that during this transformation some valuable information could be lost. Therefore, we propose a novel way to compute the similarity between two different semantic spaces, the query word space and the concept space.

In our approach, the first step is to find a proper representation of shots and queries. Since the visual content of the shot and the query are two distinct forms of information, it is I important to find suitable representations for each of them. In our system, each video shot is represented by a histogram in terms of concepts present in the shot and their confidence values extracted using the concept detectors. Similarly, each query is represented by another histogram in terms of query words (other than stop words) and their information content. The next step is to compute the relevance between query and video shots using these histograms. Sine these two histograms (representation of a video shot and a query), are in two different spaces in order to compute the relevance between two histograms we have developed an Earth Movers Distance (EMD) based relevance metric. In order to compute the distance between any two histograms using EMD we need to assign distances from each bin in the first histogram to the each bin in the second histogram. In our case these bins correspond to concepts in video shot and words in the query. In order to determine the distances between each concept and word pair we use the inverse of semantic word similarity. Specifically, for semantic word similarity we use the Pointwise Mutual Information extracted from the Information Retrieval Data[2] (PMI-IR) introduced by Turney in 2001. Then the relevance between a video shot and the query is computed as the inverse of EMD distance between their corresponding representations. Finally, the video shots are sorted based on their relevance to the given query in descending order and retrieved in this sequence.



(a) run A_UCF.W.PROD.ASR

(b) run A_UCF.W.PROD.MEAN

(c) run A_UCF.W.PROD.MAX

(d) run A_UCF.W.AVERAGE

(e) run A_UCF.PROD

Figure 4. Performance of our five runs compared to all the TRECVID 2007 runs. Dot, box and dotted line represent our result, the best result and the median result respectively.

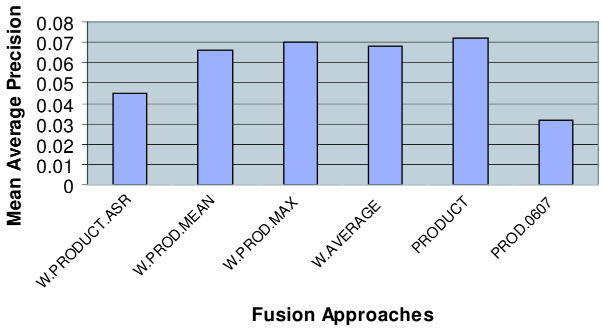


Figure 5. The returned evaluation performance comparison (in term of mean average precision) among different fusion approaches.

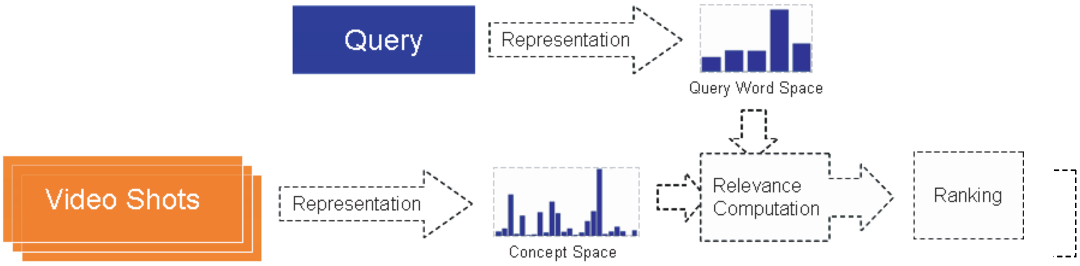


Figure 6. Automatic topic search system overview.

We submitted two runs using discussed visual content based approach. In the run $F\_A\_2\_UCFVISION3$ we used Vireo detectors and in $F\_A\_2\_UCFVISION5$ we used Columbia detectors. Unfortunately we found an error in $F\_A\_2\_UCFVISION5$ submission therefore the official results doesn't reflect the correct evaluation for this submission. Nevertheless we compared these two methods with our own evaluation using TRECVID 2007 data.

3.3. Multi-modal Fusion

For combining text and visual information, we applied an EMD based fusion method. It is very similar to the visual content based retrieval approach. Addition to it, we extract the overlapping words between ASR-MT text and query, and we assume that these words are also detected as concepts with the maximum confidence. So, representation of the query is same with the previous approach but representation of the video shot includes some additional words which are the overlapping words. Then we apply the same procedure for relevance computation and ranking. We submitted two runs for this method, $F\_A\_2\_UCFVISION4$ using Vireo detectors and $F\_A\_2\_UCFVISION\_6$ using Columbia detectors. Due to the same problem mentioned before $F\_A\_2\_UCFVISION6$ is corrupted.

3.4. Search Results

We evaluated our visual content based retrieval method using two sets of publicly available 374 concept detector models, Columbia and VIREO detector models. In the official scores of TRECVID 2007 our visual content based method (using Vireo detectors) is 500% better than our text baseline method. The performance of our submissions is shown in fig. 7.

Results of visual content based retrieval using Vireo detectors and Columbia detectors were very close. Specifically for some queries Vireo detectors are better harnessed then Columbia detectors. Comparison of Vireo detectors and Columbia detectors for each query can be seen in the fig8.

Overall, the best results are obtained using visual content only method with Vireo detectors. Comparison of best visual only, text and fusion method for each query is shown in fig9.

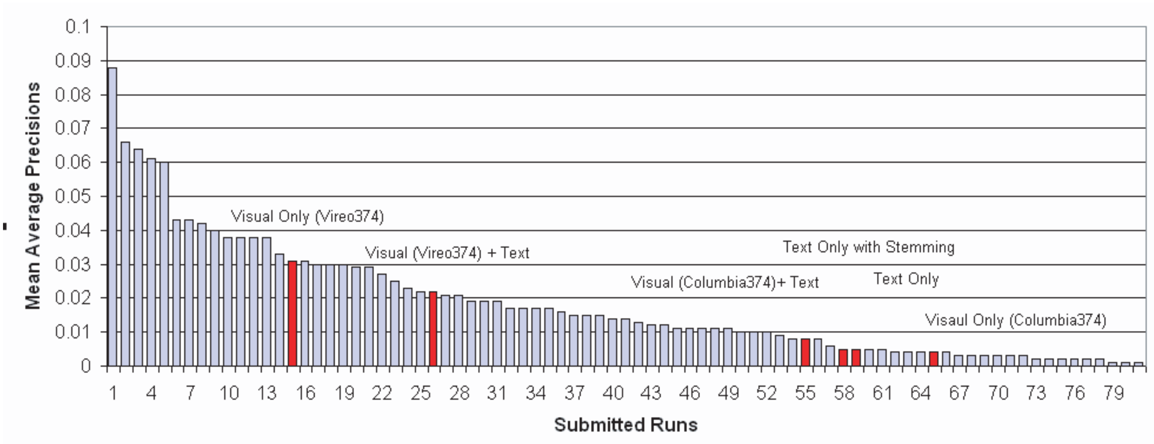


Figure 7. Mean Average Precision results for all TRECVID 2007 search submissions including our runs (red bars).

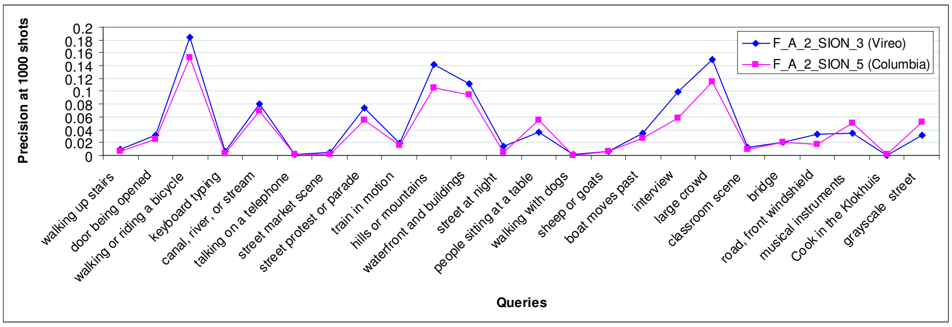


Figure 8. Visual content based retrieval results using Columbia and Vireo detectors.

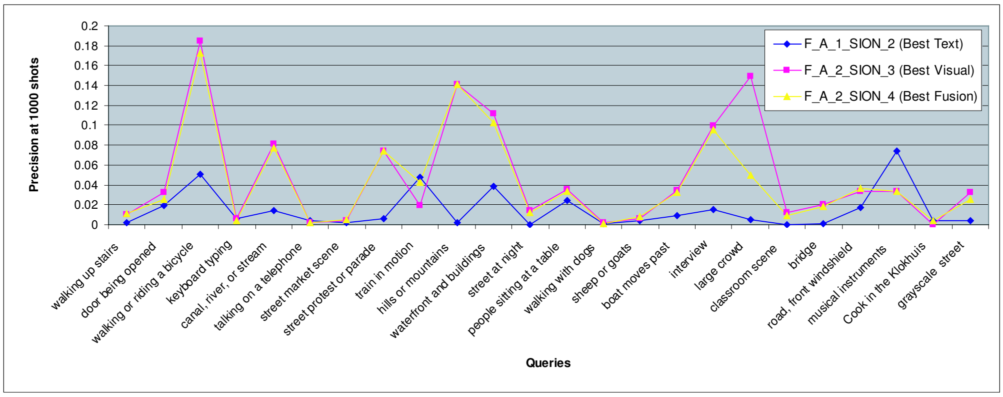


Figure 9. Performance comparison between visual content based retrieval results using Vireo detectors and text baseline.

REFERENCES

1. Marijn Huijbregts and Roeland Ordelman and Franciska de Jong , Annotation of Heterogeneous Multimedia Content Using Automatic Speech Recognition, Proceedings of the second international conference on Semantics And digital Media (SAMT), 2007
2. Turney, P. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning (ECML'01). 2001
3. A. Yanagawa, Shih-Fu Chang, Lyndon Kennedy and Winston Hsu, Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts, Columbia University ADVENT Technical Report 222-2006-8, March 20, 2007
4. Yu-Gang Jiang, Chong-Wah Ngo, Jun Yang Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval, ACM International Conference on Image and Video Retrieval (CIVR'07), Amsterdam, The Netherlands, 2007
5. B. E. Boster, I. Guyon and V. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In COLT, pp. 144-152, 1992.
6. J. Liu and M. Shah, Scene Modeling Using Co-Clustering, ICCV 2007.
7. S. Lazebnik, C. Schmid and J. Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", CVPR, 2006.
8. C. Hsu, C. Chang, and C. Lin. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf.
9. K. Grauman and T. Darrell, The Pyramid Match: Efficient Matching for Retrieval and Recognition. ICCV 2005.
10. D. G. Lowe. "Distinctive Image Features from scale-invariant keypoints". IJCV, 60(2):91-110,2004.
11. I. S. Dhillon, S. Mallela and D. S. Modha. "Information-Theoretic Co-clustering", ACM SIGKDD 2003.