Glasgow University at TRECVID 2008

P. Punitha, T. Urruty, Y. Feng, M. Halvey, A. Goyal, D. Hannah, I. Klampanos, V. Stathopoulos, R. Villa, J. Jose
{punitha,thierry,yuefeng,halvey,anuj,hannahd,iraklis,sthathv,villar,jj}@dcs.gla.ac.uk
Department of Computing Science, University of Glasgow, UK

Abstract

In this paper we describe our experiments in the automatic and interactive search tasks of TRECVID 2008. We submitted six runs, five of them are automatic and one is interactive. The automatic runs include, a text baseline, two runs based on visual features, and two runs that combine high level features and visual features. For our interactive search submission we developed a search interface based on both textual and low level features, called the Group Interface.

1. Introduction

This year Glasgow University submitted five fully automatic runs and one interactive run. The automatic runs included two baseline submissions: one mandatory run based on text only (UG-ASR-6) and another using only low level features (UG-AnLLF_5). For the other runs: UG-AnHLF_4 used a combination of high level and low level visual features, UG-TYRun1_2 used a classification based approach with low level features, UG-TYRun2_3 focused on a faster retrieval with a multidimensional indexing structure using a dimensionality reduction methodology on the same features as in UG-TYRun1_2. The interactive search run UG-Int_1 used text and visual features. The following list describes all submitted runs and the features used by them:

UG-ASR-6 Text baseline (required)

UG-AnLLF_5 Automatic search using only visual features (weighted feature selection)

UG-TYRun1_2 Automatic search using classifiers

UG-TYRun2_3 Automatic search using classifiers and indexing structures

UG-AnHLF_4 Automatic search using visual features and high level features

UG-Int_1 Interactive search run based on text and visual query examples.

All of our runs were of type c, and no other data provided were used for training. All runs were trained on the TRECVid 2007 development set only.

The remainder of the paper is organised as follows. In section 2, we describe the features used. The details of the submitted runs are given in section 3. Section 4 discusses the results and the paper concludes in section 5.

2. Feature Descriptors

2.1 Visual Features

MPEG-7 standard features, namely, Edge histogram, Homogenous texture, Colour Structure and Colour Layout features were used as the low level features. In addition to these a simple colour histogram in RGB and HSV colour spaces were used in various runs.

2.2 High Level Features

Out of the many Feature extraction donations made available for TRECVID participants, we chose the submissions of top five performing teams of 2007. A voting mechanism was then used to annotate the shots for our experiments.

2.3 Textual Features

We also used ASR and MT data. We used the shot boundary reference in order to segment video into shot-based units. Shots were then aligned with text from ASR and MT systems.

3. Search Methodologies

In this section we present the approaches which resulted with the various runs we submitted.

3.1 Automatic Runs

This section explains briefly the methods employed for the various automatic runs submitted.

3.1.1 Baseline runs

UG-ASR-6

For the retrieval based on textual features, we solely used the accompanying machine-translated text of the videos provided. Due to the scarcity of text for the various key shots, we implemented a retrieval approach taking into account consecutive shot-windows. Our aim was to obtain larger text descriptions of the retrieval units as well as to increase the frequency of potentially important terms within these descriptions. The submitted results were obtained by describing each shot by its own text, augmented by the text describing its two previous and its two following shots, thus obtaining a window of five shots altogether. In our implementation we did not take into account changes in the content, changes of programs, etc. Instead, we augmented the text accompanying the various shots according to their time-stamps, in a continuous way. This could affect boundary shots between programs, but we felt they would be too few in order to significantly affect retrieval effectiveness. As a retrieval tool, we used the Lemur Toolkit for Language Modeling and Information Retrieval [1] and its default Okapi-BM25 implementation.

Attempting to retrieve information from general video collections, especially when these come from different television programmes, based solely on extracted textual features is a nearly impossible task. The simple reason behind this is that the visual clues given form an alternative and usually independent information channel to that of speech. Since the general assumption made by television is that people perceive the supplied information visually as well as auditory, having speech describing the imagery in all genres would be redundant and tiresome. However, the topics provided for evaluation require the retrieval of visual clues, which are very rarely also described in spoken words. In other words, the textual content accompanying the videos is typically orthogonal to the visual content described by the given topics. This observation does not necessarily hold for all genres, documentaries possibly being one; however we believe that it does generally hold for popular television. Additionally, the extraction mechanisms of automatic speech recognition as well as the consequent machine translation are both techniques inherently prone to errors. It is due to these factors that the poor text-only results should not come as a surprise.

Even though the text-based results are largely poor for nearly all the topics, some of them lead to relatively good performance. The best-performing topic is #245 with textual description "Find shots of a person watching a television screen - no keyboard visible". This particular topic yields a P@20 of 0.25 and a MAP of 0.31. The first thing to notice about this topic is the word "television". Since we are dealing with shots taken from television, we can expect people in quite a few shots to be mentioning this word. A second query-specific property is that it also contains the term "screen". The terms "television" and "screen" can be expected to occur together relatively often. This helps shots containing both terms to be ranked higher in the list. At the top of the result list for topic #245 we managed to retrieve the relevant shot 115-32, as well as others around it because of our windowing approach. These shots are enough to push the indicators of retrieval effectiveness high compared to other topics. However, sampling a few of the rest of the results it becomes obvious that this is probably incidental. Even this, seemingly successful, topic supports our previous argument, i.e. that in general television, the accompanying textual content is not descriptive of the visual content and therefore retrieval solely based on textual features in this domain cannot be expected to perform well but would certainly make a difference when accompanied with visual features and user feedbacks as evident from the results of Interactive search.

UG-AnLLF_5

This method used many visual features, dependent distance measure, query diversity based feature selection and finally a fusing mechanism to obtain the final results.

Feature dependent distance measures: Depending on the features, we changed the distance measure used to compute dissimilarities between the query images and the images in the collection. Since the MPEG-7 Edge Histogram *h*, represents local edge distribution in image, the elements in the vector of 80 dimensions

can be used to obtain semiglobal and global histograms, to improve the matching performance. Hence we use the distance measure given in equation (1) to compute the similarity between two images A and B,

$$D(A,B)=\sum_{i=0}^{79}\left|h_A(i)-h_B(i)\right|+5\sum_{i=0}^{4}\left|h_A^g(i)-h_B^g(i)\right|+\sum_{i=0}^{64}\left|h_A^S(i)-h_B^S(i)\right| \qquad \text{.....(1)}$$

In order to compute distances in homogenous texture space, *T* between two images A and B, we use the sum of absolute distances as given by (2)

$$D^T(A,B)=\sum_{k=0}^{61}\left|T_A(k)-T_B(k)\right| \qquad \text{.....(2)}$$

The distance in Colour Layout features in *Y, Cb, Cr* space between two images A and B, is computed using the equation given in (3)

$$D^C(A,B)=\sqrt{\sum_i\left(Y_A(i)-Y_B(i)\right)^2}+\sqrt{\sum_i\left(Cb_A(i)-Cb_B(i)\right)^2}+\sqrt{\sum_i\left(Cr_A(i)-Cr_B(i)\right)^2} \qquad \text{..... (3)}$$

The distance in colour histogram space, *CH* between two images is computed using, the square chord distance [2], as given in (4), but for R, G, B and H, S, V bins,

$$D^{CH}(A,B)=\sum_{i=1}^{32}\left(\sqrt{CH_A(i)}-\sqrt{CH_B(i)}\right)^2 \qquad \text{.....(4)}$$

Exploitation of query examples: Query examples represent different visual contents but the same semantic. Hence, we consider each example as a separate query and then merge the results. One way of merging the results is by pooling in results of all query examples together and selecting the top the 1000 nearest neighbours according to their distance from the query example. Since there were all possibilities of losing images from many example images in above mentioned case, each result list of query examples was visited in a round robin fashion and the 1000 top results were chosen.

Feature Weight Determination: It is quite hard to decide which visual feature works better for which topic unless we are completely familiar with the data collection and the queries. With changing queries and data collection, as it happens with TRECVID corpus each year. Despite the change in corpus, the most relevant shots to the topics are the query examples provided to the TRECVID participants. As mentioned earlier since the different query contents, carry same semantics, we learn the diversity in the features using many query examples, in addition to a few frames extracted from the example shots. The pair-wise distance between the query examples for each feature using the corresponding distance measure is computed. The similarities are then normalised using the equation (5),

$$normalised\ \_x_i=\frac{x_i-\min\_x}{\max\_x-\min\_x} \qquad \text{.....(5)}$$

Thus, in order to weight the features based on the topics we find the mean deviation in similarities between the query examples, due to a visual feature *F*, as the sum of absolute differences using (6),

$$F\_Weight=\sum\left|D_i-\overline{D}\right| \qquad \text{.....(6)}$$

Higher the value of F_Weight, more useful the feature *F* is, in capturing diverse relevant images.

Automatic Result Fusion: We compute voting for each image in the result list. Images retrieved for all features are highly ranked, and then the images retrieved for two higher weighted features are ranked next highest and so on. If there is no common image in the result lists due to different features, the final result is obtained by combining the unique results from different features in a round robin fashion, by following the highest to lowest feature weights.

The run has an overall MAP of 0.0092 and P@1000 of 0.0191. Although, the results are not significant, it happens to be almost the best for some specific topics, topics, #221, #223, #232, #236, #238, #239, #240, #242, #255, and #262, #264, #266, #267. Especially for #262 P@5 is 0.6.

3.1.2 Additional Runs

In this section we present the methodologies used for the other runs submitted in comparison to the baseline runs.

UG-TYRun1_2/ UG-TYRun2_3

The principle of this algorithm is to exploit the visual content of video key frames in order to speed up the retrieval process without losing too much precision. The low level visual features are used to represent the key frame structure information, which can help the classification algorithm to understand the video categories. The methodology of the proposed algorithm is divided into two stages: indexing and retrieval. The indexing stage consists in a spatial estimation designed to classify all key frames of the collection set. The result of this classification process is used in the retrieval stage.

A SVM algorithm is applied to classify the query image into a group inside the collection set. Then, an image similarity measure is used to retrieve the query results from the estimated group using a weighted combination of low-level image features.

Image classification using spatial feature in frequency domain

Images can be recognized in various ways from different aspects, such as from the scene and from the content. A number of existing works have stated that the most efficient way for human being to identify an image is from coarse to fine, which means the coarse scene information is firstly obtained by human visual perception to understand the general topic of the image, then details of the image is acquired from the low-level descriptors such as colour, texture and edge to help brain fully understand the contents in the image.

Different images can be classified into scene groups based on their coarse scene information. In addition, scenes in the same group should have a similar global aspect, a similar global structure or similar elements. For instance, images of man-made scenes are characterized by geometry vertical and horizontal structure: urban outdoor scenes will have more vertical edges, with less in indoor scenes. Open natural scenes are characterized by a horizon line, and closed natural scenes contain a large amount of texture and boundary elements (such as mountains, trees)

Considering the possibility of classification via scene characters, global structure features estimated from frequency domain using Gabor filter is applied for classification. The Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function . Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function [6] and the Fourier transform of the Gaussian function as given in the equations (7,8 and 9).

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cos(2\pi \frac{x'}{\lambda} + \psi) \quad (7)$$

where

$$x' = x\cos\theta + y\sin\theta \quad (8)$$

and

$$y' = -x\sin\theta + y\cos\theta \quad (9)$$

where, λ represents the wave length of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

Image classification using SVM

For image classification, an image collection containing different categories is built for training purpose. Followed by spatial feature extraction, each image inside the collection and query image can be represented by the extracted feature vectors in the spatial domain.

Given the query image and image collection set, the problem becomes how to classify images from the similar scene into the same category. There are many possible classifiers which may be used, including

decision tree learning, support vector machine (SVM) , and Bayesian learning, etc. For the purposes of our approach we consider SVM to be the most appropriate as it has a high generalization performance without the need for any prior knowledge of the dataset, even when the dimension of the input space is high.

The general idea behind SVM is to map the given data into a high-dimensional feature space via a nonlinear mapping and perform a linear regression in this space. Every image is treated as a data point in the space using the extracted spatial vectors. Considering every image inside the dataset belongs to one of the predefined category, we are interested in whether we can separate them with several dimensional hyperplanes. In order to achieve maximum separation (margin) between different classes, the hyperplane is constructed under the condition that the distance from the hyperplane to the nearest data point is maximized, which means the nearest distance between a point in one separated hyperplane and a point in the other separated hyperplane is maximized.

We begin by using a set of training images from each category to train the SVM for obtaining the structure properties of each category using the extracted spatial vectors. These images used for training will not be reused in the following retrieval part. For every unclassified testing image in the collection, the SVM finds the distance between the image and the hyperplane of each category, and the hyperplane with minimum distance will be regarded as a relevant category for the image to be classified in. After looping though all the remaining images in the test set, all the images are classified into a certain category and an index file with image name and which category it belongs to was created. The organization of training, testing and query image set is detailed following.

Organization of training, testing sets

Our methodology is mainly based on a classification algorithm to first determine classes that will be used for the retrieval and also the weight of low level features for the ranking algorithm. We define seven wide categories, which are "city", "human", "indoor", "nature", "night", "outdoor", and "vehicle". These categories mostly correspond to the semantic meaning of the topics of TRECVID 2008.

The training set has been chosen from the TRECVID 2007 collection. Given the ground truth of relevant key frames for each topic in 2007 collection, we randomly chose 50 relevant key frames for each of our pre-defined categories for training purposes.

The testing collection is built by using the key frames extracted from the TRECVID 2008 collection. We apply the rule of one key frame per second for each video. Thus, in total, our test collection contains more than750,000 key-frames.

Given the training image set and the classification algorithm, the property of each category can be generated. By comparing the similarity with the category properties, each testing key frame can be classified by the method described above.

Besides, we use the SVM classification algorithm on the training and testing sets for each visual feature (colour structure, colour layout, homogeneous texture and edge histogram) in order to rank the *potential* of each low level visual feature with respect to each class. This ranking allows us to weight differently the low level features for the retrieval process.

Retrieval process of UG-TYRun1_2

The query set of each topic is formed by using the topic key frame examples given with each topic. The retrieval is done is several steps and this process is the same for all topics. Each key frame example of the topic is used separately. First, our algorithm classifies the visual query with respect to the classification previously described. The classification result provides a set of k groups, where k has been set to 2 (out of 7) in our experiments. This subset of the collection is used next by a sequential scan that retrieves the N nearest neighbors to the visual query. The ranking process of the nearest neighbors uses the two best low level visual features of each class previously determined by our classification methodology.

We obtain a set of N nearest neighbors for each query example given for each topic. Then a pooling methodology is used to extract the N video shots returned as relevant results for each topic.

Indexing and retrieval process of UG-TYRun2_3

This search run focuses on reducing the query processing time. In order to do so, we index exactly the same way as the first run. The difference between the two run appears in the low level visual features used. We use the same low level visual features, but we projected the data set on *R* (set to five) random orthogonal lines. Thus, we obtain a smaller dimensionality for each low level feature. The retrieval process proceeds in the same way as the first run, but uses all projected low level features to retrieve the N nearest neighbors with the Euclidean distance.

Conclusion, results and advantages

Our first run classifies the visual query which helps to extract a subset of the whole collection that better corresponds to the query. Then, a selection of low level visual features is made with respect to the determined class. The second run proceeds the same way but ranks the results on all projected low level features. A pooling methodology is used for both runs to retrieve the final results based on maximum five query example.

These two runs focus on the query processing time. Indeed, selecting a subset of the collection and then a subset of features or reducing the dimensionality of features by projection reduces dramatically the query processing time. Our results for the first run are although slightly lesser but similar to the base line results using only low level features, its advantage is in the search time. Since the search is performed only in specific domains based on the classifier feedback, the time is reduced to 2% of the time taken by the baseline run using only visual features. For the second run, which uses a multidimensional indexing structure, the time further gets reduced and is 1% of the baseline submission. As in all scenarios, we have tradeoff in time and effectiveness.

UG-AnHLF_4

The UG-AnLLF_5 run mentioned above used only the visual features. For this particular run, UG-AnHLF, we combined the high level features provided to the TRECVID participants. We also had a face detector to not only detect the presence of face but also to count the number of faces, which were helpful for some specific topics. These high level annotation features were used in combination to the low level features, and a similar weighting and result fusing mechanism as that for UG-AnLLF was used for retrieval.

These high level features improved the retrieval performance of many topics, #222, #225, #226, #227, #230, #246, #248, #249, #250, #257, #263 as evident in Fig. 2 and Fig. 3.

3.2 Interactive Run

For our interactive search submission we developed a search interface based on both textual and low level features, which we call the Group Interface. What is unique about our system is that in order to find a solution for many problems associated with video search we are offering an alternative search strategy. The Group Interface also allows users to create groups of videos while carrying out video search tasks. This encourages the user to break the task up into related aspects to organise their ideas and concepts. The user labels or tags these groups adding some semantic value to the group. Using this paradigm the user can concentrate on solving specific tasks rather than trying to solve large and difficult tasks or having to create ideal queries in accordance with the retrieval mechanism. These groups also can be used as starting points for further searches and exploration of the collection. The Group Interface also allows users to carry out multiple searches but also gives users more tools for organising their search results and also tagging these results and attaching some semantic value.

The Group Interface system draws inspiration from similar successful systems for image retrieval and organisation [3][4]. More recently a video search system has been developed that allow users to articulate different aspects of a search task, and organise the results accordingly [5]. However this system does not allow the same levels of interaction as the GI.

Interface

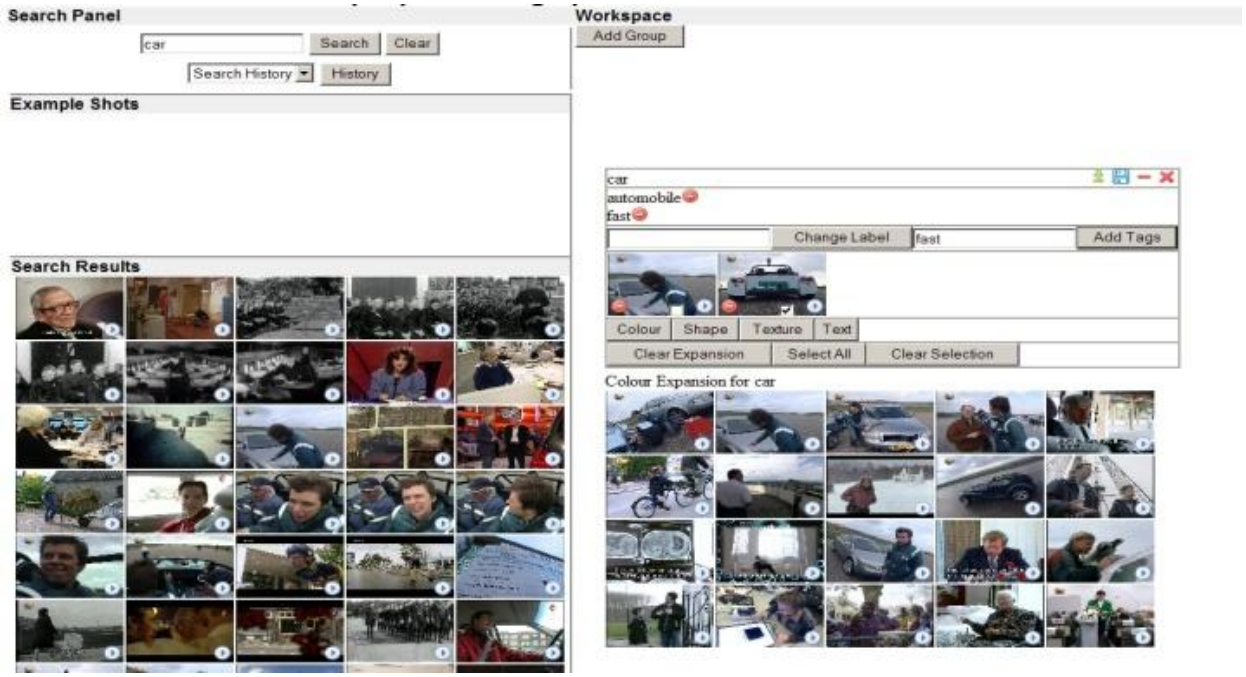

Figure 1 The Group Interface showing a search panel, result display area and the group named 'car'

The Group Interface comprises of a search panel, results display area and workspace. These facilities enable the user to both search and organise results effectively. In the Group Interface users enter a text based query in the search panel to begin their search. The users can add images from the results as examples or enter text to reformulate their queries in order to continue the search process. The result panel is where users can view the search results. Users can drag shots from this panel and add them as example shots to reformulate their query, users can also drag shots from this panel and add them as relevant images for the topic. In all panels additional information about each video shot can be retrieved. Hovering the mouse cursor over a video keyframe, will result in that keyframe being highlighted, along with neighbouring keyframes and any text associated with the highlighted keyframe. If a user clicks on the play button a popup panel appears to play the highlighted video shot. As a video is playing it is possible to view the current keyframe for that shot, any text associated with that keyframe and the neighbouring keyframes. Users can play, pause, stop and navigate through the video as they can on a normal media player.

Similar to the ImageGrouper [3] and EGO [4] systems, the main component of the Group Interface is the provision of a workspace. The workspace serves as an organisation ground for the user to construct groupings of images. Groups can be created by clicking on the create group button, before them adding a textual label for the group. Users can potentially add an infinite number of annotations to the group, but each group must have at least one annotation. Drag-and-drop techniques allow the user to drag videos into a group or reposition the group on the workspace. It should be noted that any video can belong to multiple groups simultaneously. Each group can be used as a starting point for further search queries. Users can select particular videos and can choose to view similar videos based on one or all of a set of feature categories (colour layout, homogenous texture, edge histogram or text, respectively). The workspace is designed as a potentially infinite space to accommodate a large number of groups.

User Evaluation

The 24 topics for the TRECVID interactive search task were carried out by one expert user. The user had a maximum of 10 minutes to carry out each search task. The retrieval interface logged all of the actions of the user; actions logged included shots marked as relevant, queries executed and the interaction with all of the interface elements.

4. Results

Table 1: Resultant performance of various runs

Run ID	MAP	P(10)	R-prec	Recall
UG-ASR	0.0124	0.0787	0.0390	0.0149
UG-AnLLF	0.0092	0.0792	0.0413	0.0191
UG-TYRun1	0.0058	0.0479	0.0285	0.0174
UG-TYRun2	0.0019	0.0250	0.0134	0.0094
UG-AnHLF	0.0153	0.0937	0.0517	0.0253
UG-Int	0.0243	0.2792	0.0535	0.0071

Table 2: MAP per topic

Topic	UG-ASR	UG-AnLLF	UG-TYRun1	UG-TYRun2	UG-AnHLF	Topic	UG-ASR	UG-AnLLF	UG-TYRun1	UG-TYRun2	UG-AnHLF	UG-Int
221	0.0010	0.0202	0.0042	0.0007	0.0202	245	0.3109	0.012	0.0294	0	0.012	0.0051
222	0.0046	0.0054	0.0057	0.0012	0.0353	246	0.0215	0.0037	0.005	0.0005	0.022	0.0063
223	0.0101	0.0011	0.0004	0.0004	0.0007	247	0.0351	0.0059	0.0009	0.0004	0.0123	0.0194
224	0.0008	0.0019	0.0094	0.0029	0.0019	248	0.0007	0.0254	0.0172	0.0006	0.0855	0.0131
225	0.0026	0.025	0.001	0.0003	0.0023	249	0.0009	0.009	0.0021	0.0009	0.0351	0.0134
226	0.0046	0.0246	0.0236	0.0116	0.0108	250	0.0015	0.0079	0.0088	0.0008	0.0423	0.0132
227	0.0012	0.0194	0.0054	0.0019	0.0276	251	0.0173	0.0007	0.0006	0.0013	0.0007	0.0086
228	0.0011	0.0088	0.0127	0.0024	0.0088	252	0.0041	0.0039	0.0007	0.0033	0.0039	0.0542
229	0.0031	0.0043	0.0016	0.0014	0.0118	253	0.0021	0	0.001	0.0001	0.0001	0.012
230	0.0017	0.017	0.0087	0.0024	0.0328	254	0.0011	0.002	0.0015	0.0011	0.002	0.0145
231	0.0024	0.008	0.001	0.0004	0.008	255	0.0094	0.0128	0.0029	0.0003	0.0128	0.0819
232	0.0001	0.008	0.0069	0.0009	0.008	256	0.0226	0.0028	0.0016	0.0006	0.0028	0
233	0.0006	0.0004	0.001	0.0018	0.0004	257	0.0001	0.0211	0.0249	0.0037	0.0506	0.0266
234	0.0004	0.0028	0.0025	0.0013	0.0082	258	0.0008	0.0023	0.0014	0.006	0.0023	0.0235
235	0.0001	0.0032	0.0003	0.0006	0.012	259	0.0116	0.019	0.004	0.0007	0.0204	0.001
236	0.0002	0	0.0001	0.0014	0	260	0.0069	0.0019	0.0023	0.0007	0.0019	0.0238
237	0.0011	0.0029	0.0072	0.003	0.0042	261	0.0005	0.003	0.0029	0.0006	0.003	0.0293
238	0.0000	0.0008	0.0031	0.0003	0.0008	262	0.0272	0.0641	0.0013	0	0.0641	0.067
239	0.0023	0.0092	0.006	0.002	0.0164	263	0.005	0.0191	0.0121	0.001	0.0662	0.0227
240	0.0071	0.0061	0.0007	0.0001	0.0061	264	0.0061	0.0002	0.0008	0.002	0.0002	0.0385
241	0.0085	0.0048	0.0015	0.0015	0.0048	265	0.002	0.0144	0.0135	0.0051	0.0165	0.0031
242	0.0000	0.0013	0.0002	0.0003	0.006	266	0.0064	0.0047	0.012	0.0002	0.0047	0.0068
243	0.0052	0.0002	0.0002	0.0021	0.0002	267	0.021	0.0233	0.0148	0.0143	0.0246	0.0753
244	0.0096	0.0037	0.0028	0.0005	0.0168	268	0.0124	0.0045	0.0106	0.0046	0.0045	0.0249

The results of the submitted runs are given in Table 1, which compares mean average precision (MAP), precision at 10 (P(10)), precision at total relevant shots (R-prec) and recall averaged over all topics. The MAP results for first 24 topics are shown in Table 2 and the next 24 topics which were specific for interactive search are tabulated in Table 3.

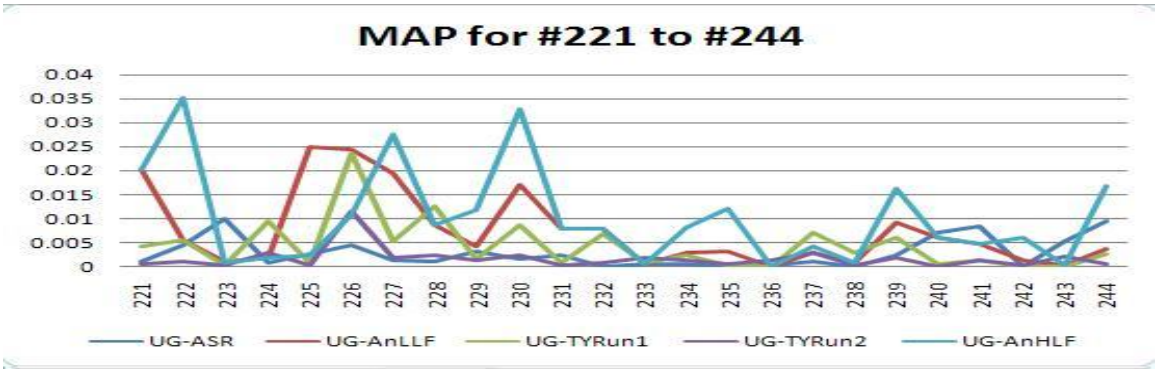

Figure 2 MAP for first 24 topics from automatic runs

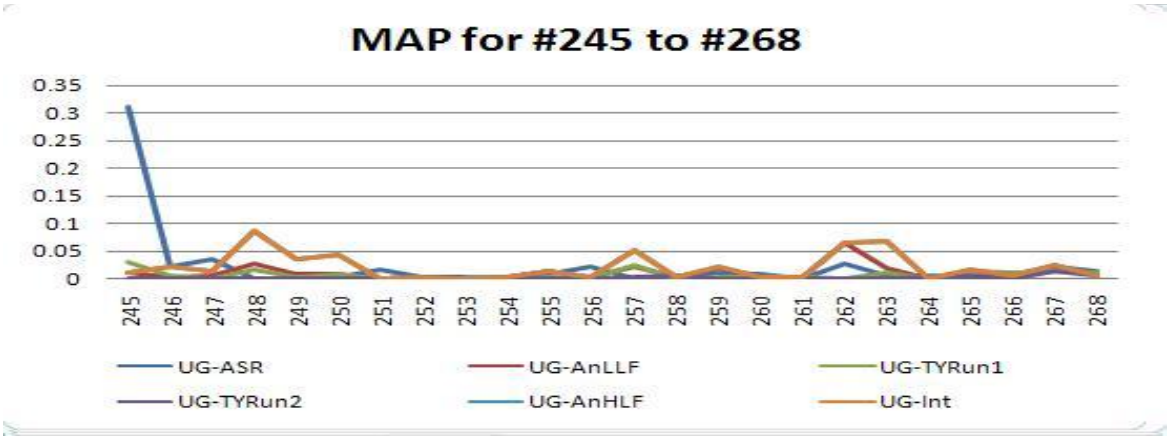

Figure 3 MAP for last 24 topics due to automatic runs and Interactive runs

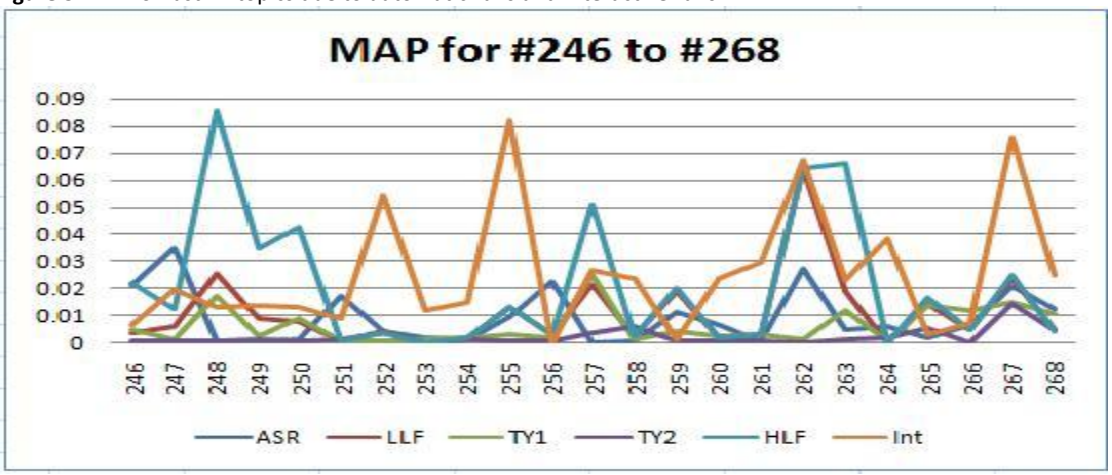

Figure 4 MAP for last 23 topics

The overall results as can be seen from Table 1, shows automatic runs using high level features performed better than the other runs. However, it cannot be compared on the values of overall resulting measurement. As each system has its own best performance from its design perspectives. In addition, every system performs better for various topics.

From the overall performance, UG-Int has the best P@10. Precision at total relevant shots, for UG-Int and UG-AnHLF is almost the same, with UG-AnLLF and UG-ASR having similar performance. UG-Int has the best overall MAP, closely followed by UG-AnHLF and UG-ASR. When it comes to Recall, one can see that UG-AnLLF and UG-TYRun1 have almost the same performance achieving 19% and 17% of recall respectively, but, when these are compared with respect to the time response, UG-TYRun1 outperforms, any runs as it consumes hardly 2% of the time taken by UG-AnLLF. The best recall achieved is 25% from UG-AnHLF run submission.

In specific to the queries we have a varied performance as can be seen from the figures 2, 3 and 4. UG-ASR has the best results for the topic #245, 'shots of a person watching a television screen - no keyboard visible' with MAP of 0.3109. MAP for any other topic is below 0.1. ASR has also higher MAP for #246, #247 and #256 which are the topics 'shots of one or more people with one or more animals' and 'shots of one or more people, singing and/or playing a musical instrument'. UG-AnLLF is best for topics #221, 'shots of a person opening a door', #225, 'shots of a bridge', and #227, 'shots of a person's face filling more than half of the frame area' which are 'more natural and consistent in colour, texture and edge features. UG-AnHLF worked better for many topics #222, #230, #246, #248, #249, #250, #257, #262, and #263, which were shots related to, '3 or fewer people sitting at a table', 'one or more vehicles passing the camera', 'one or more people in a kitchen', 'a crowd of people, outdoors, filling more than half of the frame area', 'classroom scene', 'an airplane exterior', 'a plant that is the main object inside the frame area', 'one or more people in white lab coats', and 'one or more ships or boats in the water'. Most of these topics

being based on number of people benefited from the face detector. The other topics are benefited from the high level features, specifically, classroom, Boat-ship, and airplane. UG_Int also performed better for many topics #255, #256, #257, #258, #260, #261,#262, #264, #267, #268, such as 'just one person getting out of or getting into a vehicle', 'one or more people, singing and/or playing a musical instrument' 'one or more people sitting outdoors', 'one or more animals - no people visible ', 'one or more coloured photographs, filling more than half of the frame area', 'the camera zooming in on a person's face', 'one or more signs with lettering' which are more semantic driven and difficult to retrieve with only low level. It is still surprising to see why the interactive runs had a poor performance for the topics for which ASR worked better. A few topics have the same MAP irrespective of the method and feature used, for instance, UG-ASR and UG-AnHLF for #246, UG-Int and UG-AnHLF for #262. Though, UG-TYRun1 and UG-TYRun2 do not have the best MAP, it still falls in slightly below with the performance of UG-AnLLF and has the same MAP for topic #226. However, when it comes to response time, it is incomparable with any other runs and its retrieval time irrespective of the query is approximately two seconds which indeed is what the user would expect than waiting for minutes to get a list of results.

5. Conclusions

The Glasgow University team submitted 5 fully automatic runs. Two of these runs were base line runs which used only ASR and only low level features. Among the other three runs, one run used the high level features in addition to low level feature and the other two were based on classifiers and multidimensional indexing structure where the main focus was to retain the precision but reduce the response time. In addition to these automatic runs, University of Glasgow also submitted an Interactive run based on low level features and the ASR data. Each of these systems worked better for different topics, only providing evidence on what topics are easy and what are difficult in addition to what features are better for what topics.

6. Acknowledgements

The research leading to this paper was supported by European Commission under contracts FP6-027026(K-Space) and FP6-027122(Salero).

References

[1]. www.lemurproject.org/
[2]. Liu H., Song D., Ruger S., Uren V.: Comparing dissimilarity measures for Image retrieval. In Proceedings of CIKM, (2007).
[3]. Nakazato, M., Manola, L., Huang, T.S.: ImageGrouper: a group-oriented user interface for content-based image retrieval and digital image arrangement. J. Vis. Lang. Comput. 14, 363–386, (2003).
[4]. Urban, J and Jose, J.M.: EGO: A Personalised Multimedia Management and Retrieval Tool**.** International Journal of Intelligent Systems (Special issue on "Intelligent Multimedia Retrieval"), Wiley, Vol: 21, Issue: 7, 725-745, (2006).
[5]. Villa, R., Gildea, N. and Jose, J.M.: A faceted interface for multimedia search. In Proceedings of ACM SIGIR 2008, 775-776, (2008).
[6]. Torralba A and Oliva A, Statistics of natural image categories, Network: Comput. Neural Syst. 14391-412, (2003).