CWI at the TREC-2002 video track

Thijs Westerveld,Arjen de Vries, Alex van Ballegooij
CWI
PO Box 94079, 1090 GB Amsterdam
The Netherlands
{thijs,arjen,alexb}@cwi.nl

November 4, 2002

1 Introduction

We present a probabilistic model for the retrieval of multimodal documents. The model is based on Bayes decision theory and combines models for text based search with models for visual search. The textual model, applied to the LIMSI transcripts, is based on the language modeling approach to text retrieval. The visual model, a mixture of Gaussian densities, describes keyframes selected from shots. Both models have been proven successful on media specific retrieval tasks. Our contribution is the combination of both techniques in a unified model, ranking shots on ASR-data and visual features simultaneously.

Using this model, we tried to answer the following 3 questions.

- Is it useful to identify important parts in query images?
- Can using (additional) query images from outside the search collection[1] help improve retrieval results?
- Does it help to have multiple image examples for a query, or ar we better of using only one good example?

We mainly report on post-hoc experiments using the TREC-2002 data. The official results can be found in appendix A.

[1] Throughout this document, we will refer to the search collection used in the TREC-2002 video track as *the search collection*

2 Probabilistic Multimedia Retrieval

In a probabilistic retrieval setting, the goal is to find the document D^* with highest probability given a query Q:

$$D^* = \text{argmax}_i \; P(D_i|Q) = \text{argmax}_i \; \frac{P(Q|D_i)P(D_i)}{P(Q)} \tag{1}$$

Usually, (1) is used as a scoring function and a ranked list is returned rather than the one most probable document.

If we assume that all documents have equal prior probability, (1) reduces to the maximum likelihood (ML) criterion, which is approximated by the minimum KL-divergence between query model and document model: $D^* = \text{argmin}_\text{i} \; KL[P_q(\mathbf{x})||P_i(\mathbf{x})]$.

$$KL[P_q(\mathbf{x})||P_i(\mathbf{x})] = \int P(\mathbf{x}|D_q) log \frac{P(\mathbf{x}|D_q)}{P(\mathbf{x}|D_i)} d\mathbf{x}$$

$$= \int P(\mathbf{x}|D_q) log P(\mathbf{x}|D_q) d\mathbf{x} - \int P(\mathbf{x}|D_q) log P(\mathbf{x}|D_i) d\mathbf{x},$$

where $\mathbf{x}$ are feature vectors describing the documents.

The first integral is independent of D_i and can be

ignored, thus

$$D^* = \text{argmax}_i \int P(\mathbf{x}|D_q) log P(\mathbf{x}|D_i) d\mathbf{x} \quad (2)$$

Now suppose query and document models generate a mixture of textual features $\mathbf{x}_t$ and visual features $\mathbf{x}_v$

$$P(\mathbf{x}|D_i)) = P(\mathbf{x}_t|D_i)P(t) + P(\mathbf{x}_v|D_i)P(v)^2.$$

We can then integrate over these different featuresets sepearately and arrive at the following ranking formula for multimodal retrieval [6].

$$D^* = \text{argmax}_i \ [P(t) \int_{\mathbf{x}_t} P(\mathbf{x}_t|D_q) log P(\mathbf{x}_t|D_i) d\mathbf{x}_t + P(v) \int_{\mathbf{x}_v} P(\mathbf{x}_v|D_q) log P(\mathbf{x}_v|D_i) d\mathbf{x}_v] \quad (3)$$

2.1 Text Model

To describe the probability distributions of the textual terms, we take a language modelling approach to information retrieval [2]. Such a model operates on discrete signals (i.e. words), thus we can replace the integral from (3) by a sum. Moreover, the query model D_q is usually nothing more than the empirical distribution of the query[3], therefore we only need to sum over the words in the query. The document model is usually taken to be a mixture of foreground ($P(x_{t,j}|D_i)$) and background ($P(x_{t,j})$) probabilities for the query terms $x_{t,j}$, interpolated using mixing parameter λ (cf. Section 2.1.1). If our textual query consists of N_t terms $\mathbf{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,N_t})$ then the textual part of our ranking formula is the following.

$$D_t^* = \text{argmax}_i \frac{1}{N_t} \sum_{j=1}^{N_t} log[\lambda P(x_{t,j}|D_i) + (1-\lambda)P(x_{t,j})] \quad (4)$$

Using the statistical language modelling approach for video retrieval, we would like to exploit the hierarchical data model of video, in which a video is subdivided in scenes, which are subdivided in shots, which are in turn subdivided in frames. Statistical language models are particularly well-suited for modelling such complex representations of the data. We can simply extend the mixture to include the different levels of the hierarchy, with models for shots and scenes:[4]

$$\text{Shot}^* = \arg\max_i \frac{1}{N_t} \sum_{j=1}^{N_t} \log[\lambda_{\text{Shot}} P(x_{t,j}|\text{Shot}_i) + \lambda_{\text{Scene}} P(x_{t,j}|\text{Scene}_i) + \lambda_{\text{Coll}} P(x_{t,j})] \quad \text{with } \lambda_{\text{Coll}} = 1 - \lambda_{\text{Shot}} - \lambda_{\text{Scene}} \quad (5)$$

The main idea behind this approach is that a good shot contains the query terms and is part of a scene having more occurrences of the query terms. Also, by including scenes in the ranking function, we hope to retrieve the shot of interest, even if the video's speech describes the shot just before it begins or just after it is finished. Depending on the information need of the user, we might use a similar strategy to rank scenes or complete videos instead of shots, that is, the best scene might be a scene that contains a shot in which the query terms (co-)occur.

2.1.1 Estimating Parameters

The features in the textual part of our model are simply the words themselves. For the textual part of our retrieval function (5), we only need to estimate foreground ($P(x_{t,j}|D_i)$) and background ($P(x_{t,j})$) probabilities. Both measures are estimated in the standard way, by taking the term frequency and document frequency respectively [2]. We used the TREC-2002 video search collection to find the optimal values for the mixing parameters: $\lambda_{\text{Shot}} = 0.090$, $\lambda_{\text{Scene}} = 0.210$, and $\lambda_{\text{Coll}} = 0.700$.

[2] $P(t)$ and $P(v)$ are the prior probabilities of drawing respectively textual or visual features from a document; assumed uniform accross documents.

[3] For alternative query models cf. [3]

[4] We assume each shot is a separate class and replace ω_i with Shot_i.

2.2 Image Model

We use a Gaussian Mixture Model for describing document densities [5].

$$P(\mathbf{x}_v|D_i) = \sum_{c=1}^{C} P(\theta_{i,c})\ \mathcal{G}(\mathbf{x}_v, \mu_{i,c}, \mathbf{\Sigma}_{i,c}),$$

where C is the number of components in the mixture model, $\theta_{i,c}$ is component c of document model D_i and $\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})$ is the Gaussian density with mean vector μ and co-variance matrix $\mathbf{\Sigma}$:

$$\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} e^{-\frac{1}{2}\|\mathbf{x}-\mu\|_{\mathbf{\Sigma}}}, \qquad (6)$$
$$\text{where } |\mathbf{x}-\mu\|_{\mathbf{\Sigma}} = (\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mu)$$

and n is the length of the feature vector $\mathbf{x}$.

2.2.1 Bags of Blocks

Just like in our textual approach, for the query model, we can simply take the empirical distribution of the query samples. If a query-image x_v consists of N_v samples: $\mathbf{x}_v = (x_{v,1}, x_{v,2}, \ldots, x_{v,N_v})$ then $P(x_{v,i}|D_q) = \frac{1}{N_v}$. For the document model, we take a mixture of foreground and background probabilities, i.e. the (foreground) probability of drawing a query sample from the document's Gaussian mixture model, and the (background) probability of drawing it from any Gaussian mixture in the collection. In other words, the query image is viewed as a bag of blocks (BoB), and its probability is estimated as the joint probability of all its blocks. The BoB measure for query images then becomes:

$$D_v^* = \arg\max_i \frac{1}{N_v} \sum_{j=1}^{N_v} \log\left[\kappa P(x_{v,j}|_i) + (1-\kappa) P(x_{v,j})\right], \qquad (7)$$

where κ is a mixing parameter and the background probability $P(x_{v,j})$ can be found by marginalising over all M documents in the collection:

$$P(x_{v,j}) = \sum_{i=1}^{M} P(x_{v,j}|D_i) P(D_i).$$

Again we assume uniform document priors ($P(D_i) = \frac{1}{M}$ for all i). In text retrieval, one of the reasons for mixing the document model with a collection model is to assign non-zero probabilities to words that are not observed in a document. Smoothing is not necessary in the visual case, since the documents are modelled as mixtures of Gaussians, having infinite support. Another motivation for mixing is to weight term importance: a common sample x (i.e., a sample that occurs frequently in the collection) has a relatively high probability $P(x)$ (equal for all documents), and therefore $P(x|D)$ has only little influence on the probability estimate. In other words, common terms and common blocks influence the final ranking only marginally.

2.2.2 Asymptotic Likelihood Approximation

A disadvantage of using the BoB measure is its computational complexity. In order to rank the collection given a query, we need to compute the posterior probability $P(x_v|\omega_i)$ of each image block x_v in the query for each document ω_i in the collection. For evaluating a retrieval method this is fine, but for an interactive retrieval system, optimisation is necessary.

An alternative is to represent the query image, like the document image, as a Gaussian model (instead of by its empirical distribution as a bag of blocks), and then compare these two models using the KL-divergence. Yet, if we use Gaussians to model the class conditional densities of the mixture components, there is no closed-from solution for the visual part of the resulting ranking formula (3). As a solution, Vasconcelos assumes that the Gaussians are well separated and derives an approximation, ignoring the overlap between the mixture components: the asymptotic likelihood approximation (ALA) [5]. The ALA is the measure we used in our official TREC-2002 runs, (see Appendix A. However, in post hoc analysis, we found that one of the assumptions underlying the ALA is not plausable for the collection at hand and, moreover, using it decreases performance compared to the BoB measure (for details see [6]). In the remainder of this work we will concentrate on the BoB measure.

2.2.3 Estimating Parameters

For estimating the parameters of the gaussian mixture model, we used the EM algorithm [1]. We described a document as a set of samples, where each sample is described by a number of DCT coefficients in the YCbCr color space[5]. Then we used EM to fit a mixture of 8 gaussians (for details see [6]). Finally, we described the position in the image plane of each component as a 2D-gaussian with mean and covariance computed from the positions of the samples assigned to this component. We evaluated different values for mixing parameter κ on the TREC-2002 video search collection and found the optimal value: $\kappa = 0.9$.

3 Selecting Query Images

In general, it is hard to guess what would be a good example image for a specific query. If we look for shots of the *Golden Gate bridge*, we might not care from what angle the bridge was filmed, or if the clip was filmed on a sunny or a cloudy day; visually however, such examples may be very different (Figure 1). If a user has presented three examples and no additional information, the best we can do is try to find documents that describe all example images well. Unfortunately, a document may be ranked low even though it models the samples from one example image well, as it may not explain the samples from the other images.

For each topic, we computed which of the example images would have given the best results if it had been used as the only example for that topic. We compared these *best example* results to the *full topic* results in which we used all available visual examples. The experiment was done using both the ALA and the BoB measure. In the *full topic* case, the set of available topics was regarded as one large bag of blocks. For the ALA measure, we built one mixture model to describe all available visual blocks. For BoB, we ranked documents by their probability of generating all blocks in all query images. For the single image queries in the *best example* case, we built a separate mixture model from each example and used it for ALA-ranking. For BoB ranking, we used all samples from the single visual example. Since it is problematic to use multiple examples in a query, we wanted to see if it is possible to guess in advance what would be a good example for a specific topic. Therefore, we hand-picked for each topic a single representative from the available examples and compared these *manual example* results to the other two result sets.

The results for the different settings are listed in Table 1. A first thing to notice is that all scores are rather low. When we take a closer look at the topics with higher average precision scores, we see that these mainly contain examples from the search collection. In other words, we can find similar shots from within the same video, but generalisation is a problem.

The fact that using the best image example outperforms the use of all examples shows that indeed combining results from different visual examples can degrade results. Looking at the results, manually selecting good examples seems a non-trivial task, but the drop in performance is partly due to the generalisation problem. If one of the image examples happens to come from the collection it scores high. If we fail to select that particular example, the score for the manual example run drops. Simply counting how often the manually selected example was the same as the best performing example, we see that this was the case for 8 out of 13 topics.[6]

4 Selecting Important Regions

In last year's video track, we saw that query articulation, i.e. the manual identification of important parts in a query image, can help improve retrieval results [4]. We also noticed however, that this requires an enormous effort from a user. In our probabilistic setting, selecting important (and coherent) regions is much easier. If we build a query-model, in the same way we build document models, a user can simply select one or more meaningful components from the query-model. In retrieval, we can then use only the

[5] We use the first 10 coefficients form the Y channel and the only the DC coefficient from both the Cb and the Cr channel

[6] We ignored the topics for which there is only one example and the ones for which the best example scored 0.

Figure 1: Visual examples of the Golden Gate bridge.

	full topic	best example	manual example
vt075	0.0038	**0.2438**	**0.2438**
vt076	**0.4854**	0.4323	0.1760
vt077	0.0000	0.0000	0.0000
vt078	0.0000	0.0000	0.0000
vt079	0.0000	**0.0040**	0.0000
vt080	0.0048	**0.0977**	**0.0977**
vt081	0.0000	0.0000	0.0000
vt082	**0.0330**	0.0234	0.0234
vt083	0.0000	0.0000	0.0000
vt084	**0.0046**	**0.0046**	**0.0046**
vt085	0.0000	0.0000	0.0000
vt086	0.0053	**0.0704**	**0.0704**
vt087	0.0000	0.0000	0.0000
vt088	0.0046	**0.0069**	**0.0069**
vt089	0.0000	0.0000	0.0000
vt090	0.0000	**0.0305**	**0.0305**
vt091	0.0095	**0.0095**	**0.0095**
vt092	0.0003	**0.0106**	0.0000
vt093	0.0006	**0.0006**	0.0000
vt094	0.0021	**0.0021**	**0.0021**
vt095	0.0000	0.0000	0.0000
vt096	**0.0323**	**0.0323**	**0.0323**
vt097	0.1312	**0.1408**	0.0000
vt098	0.0000	**0.0003**	**0.0003**
vt099	0.0000	0.0000	0.0000
MAP	0.0287	**0.0444**	0.0279

Table 1: MAP for Full Topics, Best Examples and Manual Examples

Bag of Blocks corresponding to the selected component(s). For example in Figure 2a, we selected the components that together form the US flag. Similarly, we can indicate we want different parts to be present in the target shots, e.g. *boat* and *water* and *sky* (Figure 2b). Note that even though the union of the sets of samples is (in this case) the full image, this differs from simply taking the using all samples as a query. If the full image were used, we would have looked for shots with relatively few water samples; the selection of components compensates for that and looks for documents that explain all 3 concepts equally well.

From each of the query images, we selected meaningfull components and we used the corresponding samples as queries. We saw that selecting components sometimes is useful. For some topics, when we used the full queries, we didn't find any relevant shots, but after specifying important parts, we did retrieve some. Further investigation is necessary to to draw general conclusions.

5 Using Query Images from Outside the Collection

In Section 3, we argued that selecting the right query image is important. On the one hand therefore, one would like to expand a query to have as many different query images as possible. On the other hand though, we saw that it is difficult to combine multiple examples in one query. We experimented with query expansion by adding additional example images found using Google image search. We investigate whether using (additional) examples from outside the collection can improve retrieval effectiveness. We expect that this is not the case; in previous experiments [6, 4] we saw that we can only find relevant shots if the query images are highly similar to the relevant shots, i.e. if they are from the same collection and preferably from the same video.

First of all, we had a look at the original examples provided by NIST. Most, if not all, of the video examples in this set come from either the search collection itself, or the higly comparable feature train or feature test set (in fact, these are distinct subsets of one larger collection). In addition, for some topics image examples from outside the video collection were available. However, only for one topic (vt084), the best example (see Table 1), actually came from outside the collection. If we extend our queries using Google image Search, for some topics, the best example results improve marginally. In general however, it seems hard to generalize accross collections.

5.1 Combining Textual and Visual runs

We combined textual and visual runs using our combined ranking formula (3). Since we had no data to estimate the parameters for mixing textual and visual information we used $P(t) = P(v) = 0.5$. For the textual part we tried both short and long queries, for the visual part we used full queries and best-example queries. Table 2 shows the results for combinations with the BoB measure. We also experimented with combinations with the ALA measure, but we found that in the ALA case it is difficult to combine textual and visual scores, because they are on different scales (see also Appendix A). The BoB measure is closer to the KL-divergence and, on top of that, more similar to our textual approach, and thus easier to combine with the textual scores.

For most of the topics, textual runs give the best results, however for some topics using the visual examples is useful. This is mainly the case when either the topics come from the search collection or when the relevant documents are outliers in the collection. This illustrates how difficult it is to search a generic video collection using visual information only. We only succeed if the relevant documents are either highly similar to the examples provided or very dissimilar from the other documents in the collection (and therefore relatively similar to the query examples). When both textual and visual runs have reasonable scores, combining the runs can improve on the individual runs, however, when one of them has inferior performance, a combination only adds noise and lowers the scores.

Topic	Tshort	Tlong	BoBfull	BoBbest	BoBfull +Tshort	BoBfull +Tlong	BoBbest +Tshort	BoBbest +Tlong
vt075	0.0000	0.0082	0.0038	0.2438	0.0189	0.0569	0.2405	**0.3537**
vt076	0.4075	0.6242	0.4854	0.4323	0.5931	**0.7039**	0.5757	0.6820
vt077	0.1225	**0.5556**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt078	0.1083	**0.2778**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt079	0.0003	0.0006	0.0000	0.0040	0.0003	0.0000	**0.0063**	0.0050
vt080	0.0000	0.0000	0.0048	**0.0977**	0.0066	0.0059	0.0845	0.0931
vt081	0.0154	0.**0333**	0.0000	0.0000	0.0037	0.0000	0.0000	0.0000
vt082	0.0080	0.0262	0.0330	0.0234	0.0181	**0.0335**	0.0145	0.0210
vt083	**0.1669**	**0.1669**	0.0000	0.0000	0.0962	0.0962	0.0078	0.0078
vt084	**0.7500**	**0.7500**	0.0046	0.0046	0.6875	0.6875	0.6875	0.6875
vt085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt086	0.0554	0.0676	0.0053	0.0704	0.0536	0.0215	**0.0791**	0.0600
vt087	**0.0591**	0.0295	0.0000	0.0000	0.0052	0.0003	0.0052	0.0003
vt088	**0.0148**	0.0005	0.0046	0.0069	0.0052	0.0046	0.0069	0.0069
vt089	**0.0764**	**0.0764**	0.0000	0.0000	0.0503	0.0503	0.0045	0.0045
vt090	0.0229	0.0473	0.0000	0.0305	0.0006	0.0075	0.0356	**0.0477**
vt091	0.0000	0.0000	**0.0095**	**0.0095**	0.0000	0.0086	0.0000	0.0086
vt092	0.0627	**0.0687**	0.0003	0.0106	0.0191	0.0010	0.0078	0.0106
vt093	**0.1977**	0.1147	0.0006	0.0006	0.0099	0.0021	0.0071	0.0012
vt094	0.0232	**0.0252**	0.0021	0.0021	0.0122	0.0036	0.0122	0.0036
vt095	**0.0034**	0.0021	0.0000	0.0000	0.0008	0.0012	0.0011	0.0010
vt096	0.0000	0.0000	**0.0323**	**0.0323**	0.0161	0.0161	**0.0323**	**0.0323**
vt097	0.1002	0.0853	0.1312	0.1408	0.1228	**0.1752**	0.1521	0.1474
vt098	**0.0225**	0.0086	0.0000	0.0003	0.0068	0.0000	0.0004	0.0003
vt099	**0.0726**	0.0606	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAP	0.0916	**0.1212**	0.0287	0.0444	0.0691	0.0750	0.0784	0.0870

Table 2: Average precision per topic, for Textual runs, BoB runs and combined runs

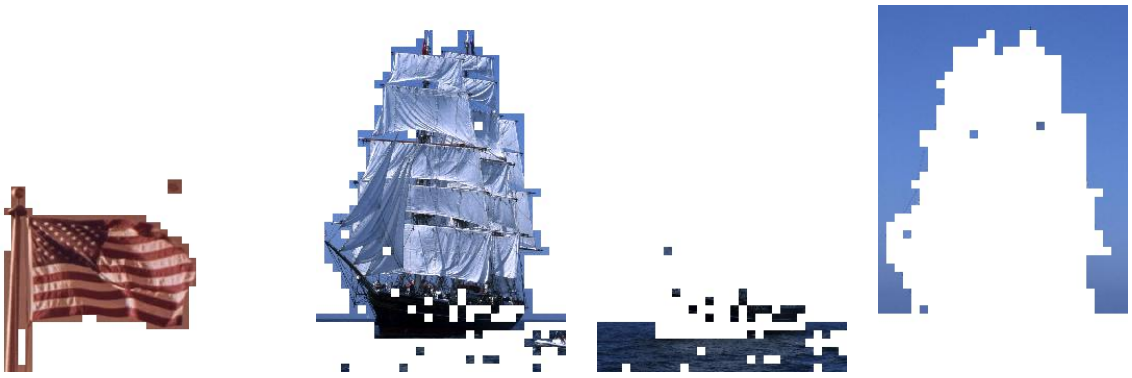

Figure 2: Selecting components from images

6 Conclusions

We presented a probabilistic framework for multimodal retrieval in which textual and visual retrieval models are integrated seamlessly and evaluated the framework using the search task from the TREC-2002 video track. We found that even though the topics were specifically designed for content-based retrieval, and relevance was defined visually, a textual search outperforms visual search for most topics. We saw that using one good visual example yields often better results than multiple examples. We also found that it is hard to generalize accross collections. Future work has to show how incorporating different sources of additional information (e.g. contextual frames, the movement in video or user interaction) can help improve results accross collections. We will also investigate in more detail what the influence is of selecting important image regions in a query.

Combining textual and visual runs seems possible using the presented framework. When one of the runs is poor, a combined run, including the noise, is less effective than the single best run. However, when the individual runs have reasonable scores, combining them improves retrieval effectiveness.

References

[1] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.

[2] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

[3] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of the 24rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2001)*, 2001.

[4] The Lowlands team. Lazy users and automatic video retrieval tools in (the) lowlands. In *The 10th Text Retrieval Conference (TREC-2001)*, 2002.

[5] N. Vasconcelos. *Bayesian Models for Visual Information Retrieval*. PhD thesis, Massachusetts Institut of Technology, 2000.

[6] T. Westerveld, A. P. de Vries, A. van Ballegooij, F. M. G. de Jong, and D. Hiemstra. A probabilistic multimedia retrieval modela and its evaluation. *EURASIP Journal on Applied Signal Processing, special issue on Unstructured Information Management from Multimedia Data Sources*, 2003, to appear.

A Official Results

In the official runs, we used the Asymptotic Likelihood Approximation (see section 2.2.2). We distinguished between the NIST images (the visual examples from the official topics) and Google images (additional examples we found with manual query expansion using Google and submitted four runs:

run1 Text only.

run2 Text + NIST images.

run3 Text + selected components from NIST images.

run4 Text + selected components from both NIST and Google images.

For run3 and run4, we manually selected *important* components from the query model (cf. Section 4). In all runs that involved visual examples, we computed a single new (8 component) Gaussian mixture model from all available visual blocks and we used that model in our ALA ranking formula. The results for the official runs and for the same runs after fixing some bugs[7] are shown in Table 3.

runName	MAP
run1	0.0917
run2	0.0016
run3	0.0022
run4	0.0038
run1 fixed	0.1212
run2 fixed	0.0082
run3 fixed	0.0137
run4 fixed	0.0069

Table 3: Official results and same runs after bug fix.

[7] A normalisation error in the training of the models and exchanding a few videos from the search and feature detection collections.