# TRECVID 2005 Experiments at Johns Hopkins University: Using Hidden Markov Models for Video Retrieval

Brock Pytlik Arnab Ghoshal Damianos Karakos Sanjeev Khudanpur

Center for Language and Speech Processing
Johns Hopkins University
3400 North Charles Street, Baltimore, MD 21218, U.S.A.
trecvid2005@clsp.jhu.edu

## ABSTRACT

The use of hidden Markov models (HMMs) for the high-level feature detection task of TRECVID 2005 is described. High-level features present in a keyframe are assumed to constitute the state-space of a Markov chain. The observed visual features of the keyframe, as well as the text accompanying the keyframe, are modeled as stochastic emissions from (unobserved) states of this Markov chain. Manual annotations of shots for the presence of these high-level features, as provided by NIST, constitute the data from which parameters of the HMM are estimated. The estimated HMM enables computation of the *a posteriori* probability that a particular high-level feature is present in a keyframe in the test collection, given the visual features of the keyframe and its accompanying closed-caption or transcribed text. It is demonstrated that different subsets of the set of visual and text features, and slightly different HMM-settings, are optimal for detecting different high-level features. Finally, it is demonstrated that the posterior probabilities of high-level features in a test keyframe, computed by the HMMs, may themselves be used as inputs to a support vector machine to further improve detection performance. Results for high-level feature detection are presented on a held-out portion of the manually annotated NIST corpus, as well as the TRECVID 2005 video search collection.

## 1. INTRODUCTION

The content of communications in the digital age is increasingly multi-modal in nature, with text, images and even speech or video being used in a single "document." Content based indexing and retrieval of multimedia is therefore becoming an increasingly important issue. Unlike text retrieval, where the modality in which the user usually specifies her information need is the same as the modality of the search collection, there is relatively little work in image and video retrieval based on textual queries. Important progress has been made in the last few years in content based image retrieval, as reported by Duygulu et al [6], Blei et al [4], Jeon et al [9] and others.

While the classical image understanding problem, *i.e.* the problem of recognizing all the objects in a given image, is very difficult due to several invariance issues, an aspect of the image and video indexing and retrieval problem that makes it relatively more tractable is the availability of *side information*: images in multimedia documents are often accompanied by descriptive text that a model may use to infer the content of an image, and video is often accompanied by speech. With this consideration, we [7] have recently developed a joint stochastic model, specifically a hidden Markov model (HMM), for images and their accompanying captions. HMM parameters are estimated from a manually annotated (training) collection of image+caption pairs; the caption-words are from a large but fixed vocabulary of objects or concepts.

The TRECVID 2005 high-level feature detection task provides an ideal testbed for investigating the strengths and limitations of the model of [7]. In this paper, we report the results of our investigations.

- We describe novel extensions of the HMM framework to model not only the visual features of an image, but also the accompanying text, which in this case is either the closed-caption accompanying the video, the output of an automatic speech recognition system, or the translation of one of these two from Arabic or Mandarin to English.
- We investigate various extensions to the HMM framework using *graphical models*, such as spatial propensities of various high-level features, or the dependence between the presence of a high-level feature and the video source (TV program).
- We show that of all the observed features of the video, different subsets are optimal for detecting different high-level features.
- We discriminatively combine the posterior probability vectors generated by various HMMs (each optimized to detect one or more high-level feature) using support vector machines (SVM), and obtain additional improvements in detection.

This paper is organized as follows. We review the basic HMM based image annotation approach in Section 2 and present results on a held out portion of the TRECVID 2005 development data. We report some novel twists on the HMM framework in Section 3. We describe the use of support vector machines for combining the outputs of several HMM annotators in Section 4. For comparative analysis, we present results for a maximum entropy based model developed at IBM in Section 5. Finally, we present retrieval results on the TRECVID 2005 high-level feature detection task, and conclude with some discussion, in Section 6.

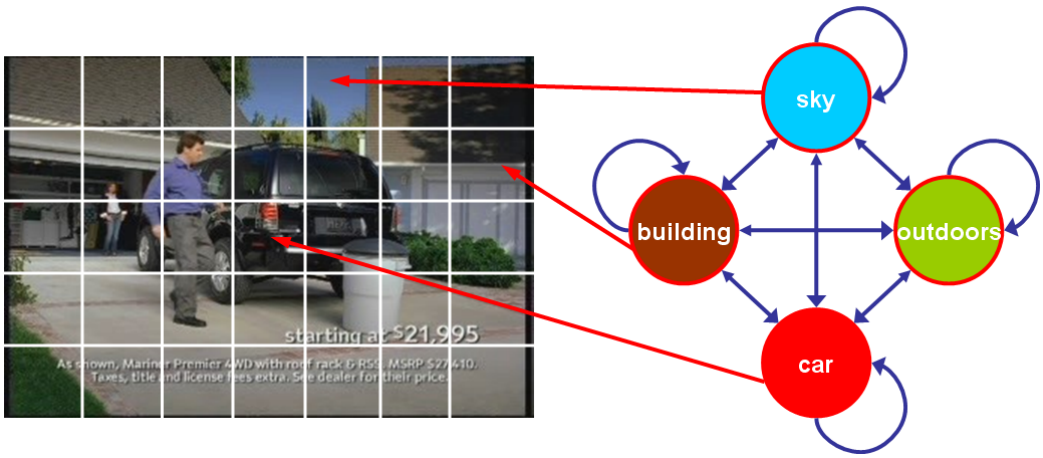


**Figure 1: Illustration of the state transitions graph and output for an image+caption HMM.**

## 2. THE BASELINE HMM SYSTEM

Let a collection $\mathcal{L} \equiv \{(I, C)\}$ of image+caption pairs be given. Let $I \equiv \{i_1, \ldots, i_T\}$ denote image-segments (image-regions), and $C \equiv \{c_1, \ldots, c_N\}$ the objects (concepts) present in that image, as specified by the label (caption). The $T$ image-regions may be object based, with each region corresponding to one semantically distinct object, or they may be a simple rectangular partition of the image into fixed-size blocks. For each image-region $i_t$, $t = 1, \ldots, T$, let $x_t \in \mathbb{R}^d$ represent color, texture, edges, shape and other salient visual features of the region. Let $\mathcal{V}$ denote the total vocabulary of the caption-words $c_n$ across the entire collection of images.

We propose to model the visual features $\{x_1, \ldots, x_T\}$ as a hidden Markov process, generated by an unobserved underlying Markov chain $\{s_t\}$ with a known initial state $s_0$ and transition probabilities $p(s_t|s_{t-1})$. We model the output density for each state $s$ as a mixture of multivariate Gaussian densities on $\mathbb{R}^d$:

$$f(x|s) = \sum_{m=1}^{M} w_{m,s} \frac{e^{-\frac{1}{2}(x-\mu_{m,s})^T \Sigma_{m,s}^{-1}(x-\mu_{m,s})}}{\sqrt{(2\pi)^d |\Sigma_{m,s}|}}, \qquad (1)$$

where $w_{m,s}$ is the mixture weight, $\mu_{m,s}$ the mean-vector and $\Sigma_{m,s}$ the diagonal covariance-matrix of the $m$-th mixture component of state $s$.

The joint likelihood of a state sequence $s_1^T \equiv \{s_1, \ldots, s_T\}$ and features $x_1^T \equiv \{x_1, \ldots, x_T\}$ is

$$f(x_1^T, s_1^T|s_0) = \prod_{t=1}^{T} f(x_t|s_t)\, p(s_t|s_{t-1}). \qquad (2)$$

The model proposed in [7] associates one state $s$ with each word in the concept vocabulary $\mathcal{V}$, as summarized in Section 2.1 for the sake of completeness, formalizing the notion that each image region is a stochastic realization of one of the concepts present in the image. Conditioning on the initial state $s_0$ is implicit but henceforth not shown to simplify notation.

### 2.1 Modeling Visual Features via HMMs

In the joint model, the states $\{s_t\}$ of the underlying Markov chain for an image $I$ take values in $C$, its caption. A label (or concept) $c \in \mathcal{V}$ appearing in two different images is modeled by the same state, and the HMMs for all images "share" states from a common pool of $|\mathcal{V}|$ *tied* states. For an image+caption pair $(I, C)$, $s_1^T \equiv \{s_1, \ldots, s_T\} \in C^T$, with $C \subset \mathcal{V}$.

Note that knowing the state sequence $\{s_t\}$ is equivalent to having the *alignment* of each image-region $i_t$ with one of the words in the caption. Even though this level of detail is generally not provided in captions, an HMM readily provides the joint likelihood of an image+caption pair $(I, C) \equiv (x_1^T, C)$ via the forward-algorithm.

$$f(x_1^T, C) = \sum_{s_1^T \in C^T} \prod_{t=1}^{T} f(x_t|s_t)\, p(s_t|s_{t-1}). \qquad (3)$$

Furthermore, given a training collection of image+caption pairs, *emission* densities $f(x|c)$ and *transition* probabilities $p(c'|c)$ of the HMM may be estimated to maximize the likelihood (3) of the training pairs. Details of this maximum likelihood estimation procedure are standard and therefore omitted (cf [11]).

For indexing a new image $I$, the HMM provides the conditional probability, given all the visual evidence $x_1^T$ in $I$, that an image-region $i_t$ was generated by a concept $c \in \mathcal{V}$, as

$$\begin{aligned} p(s_t = c|x_1^T) &= \frac{f(x_1^T, s_t = c)}{f(x_1^T)} \qquad (4) \\ &= \frac{\sum_{s_1^T : s_t = c} \prod_{t=1}^{T} f(x_t|s_t) p(s_t|s_{t-1})}{\sum_{s_1^T \in \mathcal{V}^T} \prod_{t=1}^{T} f(x_t|s_t) p(s_t|s_{t-1})}. \end{aligned}$$

This posterior probability calculation involves a ratio of two likelihoods, both calculated using the HMM. It has been noted in the speech recognition literature that scaling down the acoustic log-likelihoods by a *language model scale factor* before adding them to the language model log-probabilities is very helpful. While there is no theoretically pleasing explanation for this practice, its use is ubiquitous.

Note that it is the log-likelihood, and not the likelihood $f(x|c)$ that is scaled, and hence the posterior probability of (4) will be affected even if the numerator and denominator log-likelihoods are scaled by the same number. We therefore

| Program Name | Training | Check |
|---|---|---|
| CCTV4_DAILY_NEWS_CHN | 10 | 3 |
| CCTV4_NEWS3_CHN | 9 | 2 |
| CNN_AARONBROWN_ENG | 8 | 3 |
| CNN_LIVEFROM_ENG | 7 | 2 |
| LBC_LBCNAHAR_ARB | 6 | 1 |
| LBC_LBCNEWS2_ARB | 4 | 1 |
| LBC_LBCNEWS_ARB | 11 | 3 |
| MSNBC_MSNBCNEWS11_ENG | 9 | 2 |
| MSNBC_MSNBCNEWS13_ENG | 12 | 3 |
| NBC_NBCPHILA23_ENG | 12 | 3 |
| NBC_NIGHTLYNEWS_ENG | 5 | 2 |
| NTDTV_NTDNEWS12_CHN | 7 | 3 |
| NTDTV_NTDNEWS19_CHN | 7 | 2 |
| Total Number of Episodes | 107 | 30 |

**Table 1: Our training v/s check partitions.**

investigate the usefulness of this scaling in our HMM as well. In particular, we replace (4) with

$$p(s_t = c|x_1^T) = \frac{\sum_{s_1^T : s_t = c} \prod_{t=1}^{T} [f(x_t|s_t) p(s_t|s_{t-1})]^{\lambda}}{\sum_{s_1^T \in \mathcal{V}^T} \prod_{t=1}^{T} [f(x_t|s_t) p(s_t|s_{t-1})]^{\lambda}}, \quad (5)$$

and vary the scale factor $\lambda$ between 0.1 and 1.0.

The probability of a particular concept $c \in \mathcal{V}$ being present (somewhere) in an image may be calculated as

$$p(c|I) = \frac{1}{T} \sum_{t=1}^{T} p(s_t = c|x_1^T). \quad (6)$$

Unlabeled images in a test collection $\{I\}$ may therefore be ranked for the presence of any particular concept $c$ based on this posterior probability. In other words, the relevance score assigned to an image $I$ for a query $c$ is

$$\texttt{score}(I, c) = p(c|I). \quad (7)$$

See [7] for details of this model and its retrieval performance on the TRECVID 2003 data-set.

### 2.1.1 Specifics for the TRECVID 2005 Task

The framework described above is easily adapted to the TRECVID 2005 high-level feature detection task by treating each keyframe as the image $I$, and each high-level feature as a possible caption word $c \in \mathcal{V}$. Thus $|\mathcal{V}| = 39$ for our task, and the HMM provides a 39-dimensional score vector (7) for each keyframe in the search collection.

Visual features $x_1^T$ were extracted for each keyframe using a 5×7 rectangular partition, and provided to us by Giridharan Iyengar of IBM. The 80-dimensional features capture color moments, oriented-edges, and texture in each sub-image [1]. We performed principal components analysis (PCA) on these 80-dimensional features, not for dimensionality reduction, but to decorrelate the vector components for subsequent modeling using diagonal covariance matrices.

## 2.2 Our Training and Check Partitions

We used the manual annotations of high-level features, henceforth called *concepts*, as training material for our models. Following standard practice in statistical modeling, we first partitioned the set of 75,063 manually annotated keyframes into a *training* and a *check* set. The following considerations went into this partition.

**1.** The check set is sufficiently large to provide meaningful system comparisons.

**2.** The check set contains entire episodes of programs, preventing nearly identical keyframes of one shot (or two adjacent shots) from getting divided between the training and check sets.

**3.** The check set contains consecutive episodes of each of the 13 programs, with chronologically earlier episodes assigned to the training set, and later ones to the check set. This reflects the relationship of the development material to the eventual evaluation videos.

**4.** For each high-level feature, the check set has a sufficient number of keyframes that contain the feature.

If one arranges the annotated episodes in a matrix, with the rows corresponding to program-names and columns to broadcast-dates, then the first three criteria dictate that the training v/s check division be created by picking, for each program, a *dividing date* roughly 80% of the way from its first to last broadcast dates: episodes up to that date go into training, and episodes after that date into the check set. We did this, and found that the feature "prisoner" was highly underrepresented in the resulting check set, violating the fourth criterion above. We therefore moved back the dividing date of a couple of programs to specifically place episodes containing "prisoner" in the check set, and moved the dividing dates of other programs forward, so that the check set still had about 20% of the development data.

The resulting sizes of the training v/s check sets is shown in Table 1. Of the 57234 keyframes assigned to the training set, 39829 contain at least one of the 39 annotated features; these constitute the training data for our models. The check set contains 17289 keyframes from 10341 shots, and is our search collection for system development. We use each high-level feature as a query to retrieve shots from this check collection, and measure non-interpolated mean average precision (mAP) over either all 39 features, or the 10 benchmark features, as appropriate.

Note that while the units of retrieval are *entire shots*, the development data is annotated for the high-level features at the keyframe level. We assume a shot to be relevant if the feature is present in any of its keyframes. Furthermore, since our models assign scores to individual keyframes, not to entire shots, we have to transform the ranked list of keyframes returned by our models into a ranked list of shots. For this purpose, we perform rank-combination: the rank of a shot is the harmonic mean of the ranks of its keyframes. We also experimented with score combination, or ranking a shot according to the highest-scoring frame in the shot, but found rank-combination to be the most accurate.

**Bug Report:** The script we had used to generate a ranked-list of shots from a a ranked-list of keyframes with scores *had an error*, which we discovered a long time after submitting our official runs to NIST, and only a few days before the paper deadline for these proceedings. This bug resulted firstly in producing a slightly incorrect ranked-lists of shots for every system. More importantly, however, it also affected many design decisions and parameterization during system development, particularly the *concept specific systems* of Section 3.5 and the SVM based systems of Section 4. We have subsequently corrected the ranked-lists we submitted to NIST, and report *revised results* in Section 6 for

| Concept Name | Topic Number | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 | 0.0612 | 0.1082 | 0.2500 |
| explosion/fire | 1039 | 0.1302 | 0.1362 | 0.3000 |
| map | 1040 | 0.2933 | 0.2943 | 0.4000 |
| US flag | 1041 | 0.1766 | 0.1837 | 0.6500 |
| building exterior | 1042 | 0.0910 | 0.1587 | 0.4500 |
| waterscape | 1043 | 0.3653 | 0.3723 | 0.9500 |
| mountain | 1044 | 0.0425 | 0.0450 | 0.1000 |
| prisoner | 1045 | 0.0000 | 0.0003 | 0.0000 |
| sports | 1046 | 0.3327 | 0.3503 | 0.8500 |
| car | 1047 | 0.1574 | 0.1999 | 0.5500 |
| 10 Benchmark Concepts | | 0.1650 | 0.1849 | 0.4500 |
| All 39 Concepts | | 0.1728 | 0.2300 | 0.3782 |

**Table 2: Retrieval performance of the baseline 100G HMM system on the check set.**

| Concept Name | Topic Number | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 | 0.0601 | 0.1100 | 0.2000 |
| explosion/fire | 1039 | 0.1314 | 0.1380 | 0.2500 |
| map | 1040 | 0.3087 | 0.3119 | 0.5000 |
| US flag | 1041 | 0.1409 | 0.1492 | 0.4500 |
| building exterior | 1042 | 0.0982 | 0.1678 | 0.5000 |
| waterscape | 1043 | 0.3787 | 0.3863 | 0.9500 |
| mountain | 1044 | 0.0395 | 0.0424 | 0.1000 |
| prisoner | 1045 | 0.0000 | 0.0003 | 0.0000 |
| sports | 1046 | 0.3548 | 0.3728 | 1.0000 |
| car | 1047 | 0.1629 | 0.2059 | 0.7000 |
| 10 Benchmark Concepts | | 0.1675 | 0.1885 | 0.4650 |
| All 39 Concepts | | 0.1902 | 0.2477 | 0.4205 |

**Table 3: Retrieval performance of the row-specific 100G HMM system on the check set.**

the runs we submitted to TRECVID 2005. However, the suboptimal design decisions will take considerably longer to revise, and we expect to present the revised results at the TRECVID workshop or another suitable venue.

## 2.3 Retrieval Performance of Baseline System

The baseline HMM system was built starting with a single Gaussian density modeling each state-conditional emission probability, then mixing up progressively to 2, 4, 6, 8, 12, 16, 20, 25, 30, 40, 50, 65, 80, and 100 Gaussians. Each mixture was trained for four iterations before incrementing mixture size. In all cases, the 100 Gaussian systems performed better than the smaller systems, and larger mixtures were not tried primarily due to lack of time. During decoding, a scale $K = 0.5$ was found to be optimal. Table 2 shows the performance on the check set, across concepts.

# 3. INNOVATIONS TO THE HMM SYSTEM

This section describes some innovations on the baseline HMM system described above. Section 3.1 describes modeling spatial locality of various concepts in the keyframe, Section 3.2 describes modeling the observed speech transcriptions using a continuous-space representation of the text, Section 3.3 describes modeling the video source as another observable in an HMM, and Section 3.4 describes using only subsets of the visual feature vector. The graphical modeling toolkit GMTK [3, 2] was used extensively for these experiments.

## 3.1 Row Specific Models

Different concepts have different spatial distributions in an image. For example, grass is less likely at the top of an image than at its bottom and sky has the reverse distribution. This intuition was investigated via a class of row-specific HMMs. Three different components were modeled in a row specific manner: Gaussian means, Gaussian covariances, and a priori state probabilities. Formally, the joint likelihood equation (2) is no longer time-homogeneous:

$$f(x_1^T, s_1^T) = \prod_{t=1}^{T} f_{\mathrm{row}(t)}(x_t|s_t)\, p_{\mathrm{row}(t)}(s_t|s_{t-1}). \qquad (8)$$

The Gaussians $f(x|s)$ from the baseline system were used to initialize this model. A *block counter* was created to track the block $t$ of the image being processed, and was used to determine which collection $f_{\mathrm{row}(t)}(x|s)$ of Gaussians to use to calculate the likelihood of the observation $x$ given state $s$. This was accomplished in GMTK by using the *switching parents* construction,

$$f(a|b_1, \ldots, b_N, c) = \begin{cases} f_1(a|b_1) & \text{if } c \in \mathcal{S}_1, \\ \vdots & \\ f_N(a|b_N) & \text{if } c \in \mathcal{S}_N, \end{cases} \qquad (9)$$

which allows a random variable $a$ to depend on different parents $b_1, \ldots, b_N$ depending on the value of another random variable $c$. A special case of switching parents is when $c$ does not actually switch $a$'s parents, but simply switches the conditional density $f(a|b_1, \ldots, b_N)$. Using the block counter $t$ to determine switching, we implemented

$$f(x|s,t) = f_{\mathrm{row}(t)}(x|s) \quad \text{and} \quad p(s'|s,t) = p_{\mathrm{row}(t)}(s'|s),$$

where $\mathrm{row}(t)$ specifies which of the rows of the image the block $t$ is in.

### 3.1.1 Check Set Results

We note that the baseline system of Section 2.3 used uniform transition probabilities $p(s'|s)$, since estimating time-homogeneous transition probabilities did not yield any improvements in mAP. However, permitting the transition probabilities to be non-homogeneous in time as described above yields an improvement, as seen in Table 3. In particular, the results of Table 3 assume time-dependent but *state independent* transition probabilities $p(s'|s,t) = p_{\mathrm{row}(t)}(s')$. No additional gains were obtained by making the Gaussian densities row-dependent, but further investigation of this issue is in progress[1].

## 3.2 Text Vector Models

We believe that the audio content of a shot could be useful for annotating concepts in that shot. The text was transformed from a stream of words into a 500-dimensional $\mathbb{R}$-

[1]This system, submitted as System 2, was trained on only our training set instead of the full development set.

valued vector so that it could be modeled jointly with the image features within our HMM framework. This vector representation of the text was appended to each 80-dimensional image vector in the baseline HMM system. We initialized the emission densities using the Gaussians from the baseline system.

For English videos, ASR text was available. For Arabic video, machine translation of ASR text was available. For Chinese video, both ASR text and machine translation of ASR text was available. The text corresponding to each shot was extracted using the time stamps in the text files and master shot reference file provided by NIST [10]. A word in the English ASR output was associated with a shot if the start-time of the word was within the boundaries of the shot. For Arabic and Chinese videos, decisions were made on the basis of phrases rather than words, because timing information was only available per phrase. A phrase was considered to be associated with a shot if the beginning of the phrase occurred within the shot. For Chinese, if a particular shot had associated text in both the CMU-provided MT and the NIST-provided MT, the text was merged. All keyframes within a shot were assigned the same text.

To produce a text-vector representation for a shot, we counted how many times each word in the vocabulary appeared in the text associated that shot. We used the Okapi-BM25 weighting [8] to provide a continuous score for each word.

$$\mathrm{wt}(n,d) = \mathrm{tf}_d \times \frac{\log(\frac{N-n+0.5}{n+0.5})}{k_1 \times ((1-b) + b \times \frac{\mathrm{dl}_d}{\mathrm{avdl}} + \mathrm{tf}_d}},$$

with $k_1 = 2$ and $b = 0.75$. Each shot now had a score from 0-1 for each word creating a matrix ($M$) of size words $\times$ shots. This matrix, containing scores for both development and test shots, provided the scores on which to produce a pseudo-document using LSA [5]. First SVD was run on the matrix $M$, producing three matrices: $T$, $S$, and $D^{-1}$. SVD was limited to only the first 500 eigen-values. A pseudo-document for document $d$ was then produced by multiplying the transpose of the $d$-th column of $M$ by $T$ and $S^{-1}$. Each shot is represented by this 500-dimensional pseudo-document.

We tried using only this 500-dimensional text feature to rank the shots, but this gave performance that was much worse than the image systems (mAPs were in the .05 range). Instead, we decided to model the image and text together to rank the images. The 500-dimensional pseudo-document was made an additional observed stream, along-side the 80-dimensional vector of image features, in each block $t$ of the keyframe; i.e. the text vector was replicated $T$ times for the $T$ blocks of the image.

$$f(x_1^T, y_1^T, s_1^T) = \prod_{t=1}^{T} [f_{\mathrm{I}}(x_t|s_t) f_{\mathrm{T}}(y_t|s_t)]\ p(s_t|s_{t-1}).$$

The text-vector emission density $f_{\mathrm{T}}(y|s)$ was a single, 500-dimensional Gaussian. GMTK permits the log-likelihoods of different streams of observations to be scaled differently. We used this option during decoding and experimented with different scale factors $\lambda_v$ and $\lambda_t$, as

$$f(x_1^T, y_1^T, s_1^T) = \prod_{t=1}^{T} \left[ f_{\mathrm{I}}^{\lambda_v}(x_t|s_t) f_{\mathrm{T}}^{\lambda_t}(y_t|s_t) \right]\ p(s_t|s_{t-1}),$$

which were chosen for optimal performance on the check set.

| Concept Name | Topic Number | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 | 0.0944 | 0.1454 | 0.5000 |
| explosion/fire | 1039 | 0.1641 | 0.1699 | 0.3000 |
| map | 1040 | 0.3807 | 0.3837 | 0.4500 |
| US flag | 1041 | 0.1196 | 0.1274 | 0.3500 |
| building exterior | 1042 | 0.1046 | 0.1748 | 0.6500 |
| waterscape | 1043 | 0.3905 | 0.3968 | 0.9500 |
| mountain | 1044 | 0.0341 | 0.0366 | 0.1000 |
| prisoner | 1045 | 0.0000 | 0.0003 | 0.0000 |
| sports | 1046 | 0.3578 | 0.3757 | 1.0000 |
| car | 1047 | 0.1746 | 0.2186 | 0.6500 |
| 10 Benchmark Concepts | | 0.1820 | 0.2029 | 0.4950 |
| All 39 Concepts | | 0.1902 | 0.2490 | 0.4436 |

**Table 4: Retrieval performance of the joint visual-text 101G HMM system on the check set.**

### 3.2.1 *Check Set Results*

The visual and text system was the best single HMM system on our check data. Table 4 contains the detailed performance of this system. This system trained the concept transition probabilities. Training the transitions improved the performance by 0.0051 over all concepts but improved the benchmark concept performance by 0.0252. The scale setting on the $f(x|s)$ and $f(y|s)$ had little overall effect on the performance of the system, but did affect performance on specific concepts. For example, using scales of 0.1 reduced overall performance (by 0.007 overall, 0.0124 benchmark) but improved performance on the sports concept by 0.031 (0.3671 became 0.3981) over the next best system for sports. In the system submitted, 51 training iterations and a scale of 0.5 for both $f(x|s)$ and $f(y|s)$ were used. Unlike the baseline HMM system, which uses uniform transition probabilities, transition probabilities $p(s'|s)$ are learned during training and used during decoding in this system.

## 3.3 Source Models

Visual properties of keyframes vary systematically according to the program the keyframe originates from. To capture this phenomenon, we modeled the source-name as an additional observable feature of each image block of every keyframe in the HMM setting. Similar to the text-vector, the source-name was replicated $T$ times for each keyframe to augment the 80-dimensional visual feature vectors as another stream of observations $n_1^T$. Since there are only 13 sources, we modeled the source-name as a discrete random variable and estimated its state-conditional probability mass function.

$$f(x_1^T, y_1^T, n_1^T s_1^T) = \prod_{t=1}^{T} [f_{\mathrm{I}}(x_t|s_t) f_{\mathrm{T}}(y_t|s_t) p(n_t|s_t)]\ p(s_t|s_{t-1}),$$

This model did not result in any significant improvement in mean retrieval performance over the 39 concepts, but was found to improve over the joint visual-text system of Table 4 for some of the 10 benchmark concepts.

## 3.4 Using Subsets of the Visual Features

The raw 80 dimensional image feature vectors provided

| Concept Name | Topic Number | Sys1 | Sys2 | Sys3 | 100G ET | 100G TC | 100G CE |
|---|---|---|---|---|---|---|---|
| walking | 1038 | **0.0944** | 0.0601 | 0.0612 | 0.0643 | 0.0476 | 0.0732 |
| explosion/fire | 1039 | **0.1641** | 0.1314 | 0.1302 | 0.0665 | 0.1505 | 0.0703 |
| map | 1040 | **0.3807** | 0.3087 | 0.2933 | 0.0428 | 0.2726 | 0.3055 |
| US flag | 1041 | 0.1196 | 0.1409 | 0.1766 | 0.1353 | 0.2163 | **0.2590** |
| building exterior | 1042 | **0.1046** | 0.0982 | 0.0910 | 0.0524 | 0.0686 | 0.0552 |
| waterscape | 1043 | **0.3905** | 0.3787 | 0.3653 | 0.2435 | 0.3075 | 0.2868 |
| mountain | 1044 | 0.0341 | 0.0395 | 0.0425 | **0.0588** | 0.0289 | 0.0338 |
| prisoner | 1045 | 0.0000 | 0.0000 | 0.0000 | **0.0006** | 0.0000 | 0.0002 |
| sports | 1046 | **0.3578** | 0.3548 | 0.3327 | 0.2412 | 0.3091 | 0.2733 |
| car | 1047 | **0.1746** | 0.1629 | 0.1574 | 0.0667 | 0.1569 | 0.0801 |
| 10 Benchmark Concepts | | **0.1820** | 0.1675 | 0.1650 | 0.0972 | 0.1558 | 0.1437 |
| All 39 Concepts | | **0.1902** | **0.1902** | 0.1728 | 0.1349 | 0.1701 | 0.1469 |

**Table 5: Sub-vector based HMMs do not outperform the baseline HMM system in overall mAP, but are better suited for detecting a few individual concepts. All scores are mAP for the top-2000 images.**

to us by IBM may be divided into three sub-vectors: LAB color moments, edge histograms, and texture. If some concepts are mostly characterized by one of these visual features, then removing the irrelevant features may improve performance. For example, removing the texture components of the visual feature vector indeed improved performance on detecting "US flag" (see table 5). These systems are, of course, not expected to improve overall performance but instead to explore suitability of various image features for specific concepts. Instead of performing PCA on the 80-dimensional vectors, sub-vectors corresponding to color (C), texture (T) and edge (E) features are first extracted, and PCA was performed separately on the sub-vectors. The resulting decorrelated sub-vectors were then concatenated back (as needed) to form the appropriate sub-vectors. Systems were then trained on each of these sub-vectors.

We trained various versions of the baseline HMM systems of Section 2.1 using as the observed features $x_1^T$ either only the sub-vector C or T or E, as well the combinations CT, CE and TE. In this nomenclature, the system of Section 2.3 uses CTE as its visual features. As expected, no single sub-vector based system had better retrieval performance averaged over all 39 concepts (or even over the 10 benchmark concepts) than the baseline HMM system. However, specific visual features significantly outperformed the baseline HMM system for retrieving specific concepts.

This last observation leads naturally to the idea that perhaps different systems should be used to retrieve shots containing different high-level features, as described next.

## 3.5 Concept Specific HMM Design

It was observed during the investigation of the HMM variants of Section 3 that different configurations of visual features worked best for each concept. Table 5 shows the retrieval performance of some of the sub-vector based HMM systems of Section 3.4, and brings out this point concretely.

Based on this information, we set about to search, among all the HMM systems we had developed, the system that had the best performance on the check set for each of the 10 benchmark concepts. As expected, choosing the best systems for the check set improves performance on the check set, as shown in Table 6, where it is also indicated as to what kind of HMM system was best for the each of the 10 benchmark concepts. Mean average precision on benchmark concepts, in particular, jumps by 0.0226 over the best single HMM system, and mAP on all concepts increases by 0.0295. While these increments are large, they represent a large amount of tuning specifically for the check set, and there is a significant chance of over-fitting to the check data.

| Concept Name | Topic Number and HMM Used[a] | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 V+T | 0.0933 | 0.1452 | 0.5000 |
| explosion | 1039 V(R) | 0.1781 | 0.1827 | 0.4000 |
| map | 1040 V+T | 0.3931 | 0.3961 | 0.4500 |
| US flag | 1041 CE | 0.2590 | 0.2671 | 0.8500 |
| building | 1042 V+T | 0.1123 | 0.1879 | 0.4500 |
| waterscape | 1043 V+T | 0.3905 | 0.3968 | 0.9500 |
| mountain | 1044 TE | 0.0588 | 0.0611 | 0.1500 |
| prisoner | 1045 CT | 0.0014 | 0.0015 | 0.0024 |
| sports | 1046 V+T | 0.3863 | 0.4022 | 1.0000 |
| car | 1047 V+T+S | 0.1751 | 0.2150 | 0.6500 |
| 10 Benchmark Concepts | | 0.2048 | 0.2254 | 0.5402 |
| All 39 Concepts | | 0.2197 | 0.2772 | 0.5129 |

[a]V = models using 80 dimensional visual features
T = models using 500 dimensional textual features
S = models using the video source feature
R = models using row specific transitions and Gaussians
CT = models using color moments and texture
CE = models using color moments and edge histograms
TE = models using texture and edge histograms

**Table 6: Retrieval performance on the check set for post hoc concept-specific HMM selection.**

# 4. USING SUPPORT VECTOR MACHINES

It has been noted that accurate detection of one concept is often helpful in detecting other semantically associated concepts. For instance, "building exteriors" are unlikely to be seen in a keyframe that is also marked as "indoor." To explore models that use the posterior probabilities of other concepts to improve the detection of the benchmark concepts, we turned to SVMs. Specifically, for each keyframe of the check set, we used the posterior probability of all 39 concepts generated by the HMM based systems as a *feature*

*vector* of the keyframe. We then built a separate SVM classifier for each benchmark concept to discriminate between keyframes that contained the concept and keyframes that didn't.

Since we had already used up all annotated data in either training the HMMs or checking their performance, there was no additional held out data on which to develop the SVMs. We therefore resorted to a cross-validation on the check set to measure SVM performance.

Feature vectors of 39 dimensions each were generated for each keyframe in the check set using each of the 10 concept specific HMMs of Table 6. Hence, 17289 feature vectors, each of 390-dimensions, were generated in total. The 17289 keyframes were split into 30 parts, one for each *episode* in the check set. The check set was therefore split into 30 "training and test" pairs in a leave-one-out fashion: each pair comprised 29 shots for training an SVM and 1 shot for testing. A separate SVM was estimated for each of the 10 benchmark concepts.

We tried linear and Gaussian kernels, and the Gaussian kernels gave us most of the improvement. We searched over a set of regularization parameters R, ranging from 0.001 to 10, and Gaussian kernel parameters $\gamma = 1/(2\sigma^2)$, ranging from 0.001 to 1. The cross-validation experiments were done as follows:

- For each concept $c$ of the 10 benchmark concepts, we marked each feature vector (keyframe) as having the concept $c$ or not; that is, we gave a class label of +1 or -1 respectively.
  For each pair of parameters (R, $\gamma$), we trained a classification SVM with a Gaussian-kernel on 29 of the episodes, and we tested it on the remaining episode. Each keyframe in the 1-episode test set was given a score, equal to the confidence of the classification (signed distance from the decision boundary) returned by the SVM for $c$. We performed this training/testing 30 times, thus assigning a score to each keyframe in the check set. This score was a function of the tuple (R, $\gamma$, $c$), and all keyframes were sorted according to this score. From this ranked list, an overall mAP(R, $\gamma$, $c$) was then obtained.

- The SVM parameters (R, $\gamma$) that gave the highest mAP for each topic and system were obtained.

## 4.1 Check Set Results

The best fitting SVM parameters were in the vicinity of $R = 1$ and $\gamma = 0.001$ for each of the 10 SVM, reflecting remarkable robustness across concepts. The cross-validation performance of the SVM systems on the check set is reported in Table 7.

Note that the SVM system outperforms the best HMM based systems by a considerable margin. In particular, note that mAP on the concept "prisoner" jumps considerably, and this is almost certainly due to the exploitation of correlation between "prisoner" and other concepts, since the SVM is simply post-processing the posterior probability of the HMM based systems, and therefore does not have any additional evidence. Finally, note that the SVM does not always find the optimal classifier: the mAP on "US flag" has dropped even though the posterior probabilities of the best performing HMM for it was included in the input to the SVM.

| Concept Name | Topic Number | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 | 0.1442 | 0.1893 | 0.6500 |
| explosion/fire | 1039 | 0.1990 | 0.2055 | 0.3500 |
| map | 1040 | 0.6497 | 0.6528 | 1.0000 |
| US flag | 1041 | 0.1620 | 0.1705 | 0.7000 |
| building exterior | 1042 | 0.1802 | 0.2385 | 0.5000 |
| waterscape | 1043 | 0.4274 | 0.4341 | 0.9000 |
| mountain | 1044 | 0.1698 | 0.1732 | 0.3500 |
| prisoner | 1045 | 0.0028 | 0.0030 | 0.0000 |
| sports | 1046 | 0.5028 | 0.5172 | 1.0000 |
| car | 1047 | 0.2833 | 0.3216 | 0.8500 |
| 10 Benchmark Concepts | | 0.2721 | 0.2906 | 0.6300 |

**Table 7: Retrieval performance of SVM classifiers on the check set.**

| Concept Name | Topic Number | System Used[a] | mAP (top 2000 shots) |
|---|---|---|---|
| walking | 1038 | SVM | 0.1442 |
| explosion | 1039 | SVM | 0.1990 |
| map | 1040 | SVM | 0.6497 |
| US flag | 1041 | CE | 0.2590 |
| building | 1042 | SVM | 0.1802 |
| waterscape | 1043 | SVM | 0.4274 |
| mountain | 1044 | SVM | 0.1698 |
| prisoner | 1045 | SVM | 0.0028 |
| sports | 1046 | SVM | 0.5028 |
| car | 1047 | SVM | 0.2833 |
| 10 Benchmark Concepts | | | 0.2818 |

[a]CE = HMMs using color moments and edge histograms
SVM = models built by SVMs

**Table 8: Retrieval performance on the check set for post hoc concept-specific HMM/SVM selection.**

In the spirit of Section 3.5 we again built a set of concept-specific systems for retrieval on the check set, but now permitted the SVM systems as candidates. As obvious from a comparison of Tables 6 and 7, 9 out of the 10 concepts are best detected by an SVM system, and the remaining 1 by HMM systems. The best possible concept-specific system performance on the check set is reported in Table 8.

# 5. IBM'S MAXIMUM ENTROPY MODEL

Since our HMM and SVM systems all use the visual features provided by IBM, it is interesting to compare the retrieval performance of our systems with those of the maximum entropy based model of [1].

## 5.1 Check Set Results

We gratefully acknowledge the assistance of Giridharan Iyengar and Janne Argillander in carrying out this comparison. They trained their system on our training partition of the development data, and provided us ranked-lists of shots in the check set. The retrieval performance of their system

| Concept Name | Topic Number | mAP (top 2000 shots) | mAP (all shots) | Precision at 20 |
|---|---|---|---|---|
| walking | 1038 | 0.1143 | 0.1637 | 0.4000 |
| explosion/fire | 1039 | 0.1870 | 0.1908 | 0.4000 |
| map | 1040 | 0.1791 | 0.1801 | 0.0500 |
| US flag | 1041 | 0.1017 | 0.1085 | 0.3500 |
| building exterior | 1042 | 0.1458 | 0.2197 | 0.4000 |
| waterscape | 1043 | 0.2947 | 0.3013 | 0.7000 |
| mountain | 1044 | 0.0988 | 0.0998 | 0.2500 |
| prisoner | 1045 | 0.0072 | 0.0074 | 0.0000 |
| sports | 1046 | 0.3444 | 0.3614 | 0.6500 |
| car | 1047 | 0.2935 | 0.3345 | 0.9500 |
| 10 Benchmark Concepts | | 0.1767 | 0.1967 | 0.4150 |

**Table 9: Retrieval performance of the IBM system on the check set.**

is reported in Table 9.

Finally, we repeat the post hoc selection of the best system per concept, as explained in Section 3.5, but also include the IBM system in the selection. We do so, and report the retrieval performance in Table 10. Note that 2 out of the 10 concepts are better detected by the IBM system, 7 out of 10 by the SVM system, and 1 out of 10 by an HMM system.

# 6. TRECVID 2005 RESULTS

We submitted the following seven runs for the official TRECVID 2005 high-level feature detection task.

| Sys No | System Description |
|---|---|
| 1 | Visual+Text HMM System (Table 4) |
| 2 | Row Specific HMM System (Table 3) |
| 3 | Baseline HMM System (Table 2) |
| 4 | Concept-Specific HMM Systems (Table 6) |
| 5 | IBM's Maximum Entropy System (Table 9) |
| 6 | Concept-Specific HMM/SVM (Table 8) |
| 7 | Concept-Specific HMM/SVM/ME (Table 10) |

Table 11 shows the official retrieval performance of the systems submitted for the evaluation. As discussed in Section 2.2, the runs submitted to NIST were produced using a script that had a bug. Fixing the bug results in the regeneration of the ranked-lists on the test collection for Systems 1 through 4. The bug fix also results in significant changes to the SVMs selected, and hence significant changes to the results of Systems 6 and 7. System 5 is not affected by the bug, since our frame-rank to shot-rank conversion was not applied to the IBM system: it directly returned ranked shots. The *revised results* for our seven runs are presented in Table 12, and should be considered a more accurate reflection of the performance of our systems.

Comparing the performance of the seven systems on the 10 benchmark concepts on the check set, and the corresponding numbers in Table 12 on the NIST evaluation, the following remarks are in order.

1. The performance of our baseline System 3 dropped significantly on the test set. However, the relative improvements from row-specific modeling in System 2 and the joint visual-text model of System 1 were consistent, indicating that the improvements are robustly repeatable.

2. The post hoc selection of the best HMMs in System 4 yielded a large improvement over System 1 on the test set, but the performance of System 4 had more degradation (from the check set to the evaluation) than any of the individual systems. This strongly suggests that the system selection decisions were not well motivated, and the methodology needs to be revisited.

3. The IBM system performed better than expected on the test data, suggesting that even the design of Systems 1-3 may have been prone to overfitting on the check set. This also needs to be revisited in our follow-on research.

4. The HMM/SVM System 6 improved significantly over the best HMM System 4 on the test set. This again validates the robustness of the method. The improvement on the test set , however, is not as dramatic as it was on the check set, again suggesting possible overtuning.

5. The HMM/SVM/ME System 7 also shows signs of poor system selection based on the check set. Concepts on which the IBM system performs better have been retrieved using the worse-performing SVM, and vice versa. Therefore, overall retrieval performance of System 7 is worse than that of Systems 5 and 6, contrary to our expectations.

It seems clear that our concept-specific system selection approach was very prone to check-set oddities. Either a different approach is needed or a stricter criterion (for selecting a concept-specific system that is not high-scoring when averaged over many concepts) should be applied. Instead of only requiring the system to perform well for a concept, perhaps it should have to be (statistically) significantly better than the overall best system averaged over all concepts.

| Concept Name | Topic Number | System Used [a] | mAP (top 2000 shots) |
|---|---|---|---|
| walking | 1038 | SVM | 0.1442 |
| explosion | 1039 | SVM | 0.1990 |
| map | 1040 | SVM | 0.6497 |
| US flag | 1041 | CE | 0.2590 |
| building | 1042 | SVM | 0.1802 |
| waterscape | 1043 | SVM | 0.4274 |
| mountain | 1044 | SVM | 0.1698 |
| prisoner | 1045 | ME | 0.0072 |
| sports | 1046 | SVM | 0.5028 |
| car | 1047 | ME | 0.2935 |
| 10 Benchmark Concepts | | | 0.2833 |

[a]CE = HMMs using color moments and edge histograms
SVM = models built by SVMs
ME = the IBM maximum entropy model

**Table 10: Retrieval results on the check set for post hoc concept-specific selection of HMM/SVM/ME systems.**

| Concept Name | Topic Number | Sys1 | Sys2 | Sys3 | Sys4 | Sys5 | Sys6 | Sys7 |
|---|---|---|---|---|---|---|---|---|
| walking | 1038 | 0.143 | 0.098 | 0.074 | 0.147 | 0.187 | 0.086 | 0.086 |
| explosion/fire | 1039 | 0.038 | 0.039 | 0.038 | 0.050 | 0.095 | 0.050 | 0.095 |
| map | 1040 | 0.356 | 0.342 | 0.279 | 0.358 | 0.437 | 0.358 | 0.358 |
| US flag | 1041 | 0.082 | 0.091 | 0.094 | 0.001 | 0.058 | 0.001 | 0.001 |
| building exterior | 1042 | 0.152 | 0.156 | 0.113 | 0.191 | 0.245 | 0.191 | 0.245 |
| waterscape | 1043 | 0.190 | 0.196 | 0.184 | 0.190 | 0.278 | 0.190 | 0.190 |
| mountain | 1044 | 0.105 | 0.101 | 0.106 | 0.090 | 0.221 | 0.070 | 0.070 |
| prisoner | 1045 | 0.007 | 0.000 | 0.002 | 0.002 | 0.000 | 0.001 | 0.000 |
| sports | 1046 | 0.259 | 0.239 | 0.256 | 0.273 | 0.263 | 0.273 | 0.273 |
| car | 1047 | 0.120 | 0.110 | 0.097 | 0.104 | 0.200 | 0.080 | 0.200 |
| 10 Benchmark Concepts | | 0.1452 | 0.1372 | 0.1243 | 0.1406 | 0.1984 | 0.1300 | 0.1518 |

**Table 11: Official TRECVID 2005 results of JHU systems.**

| Concept Name | Topic Number | Sys1 | Sys2 | Sys3 | Sys4 | Sys5 | Sys6 | Sys7 |
|---|---|---|---|---|---|---|---|---|
| walking | 1038 | 0.1494 | 0.0984 | 0.0755 | 0.1234 | 0.187 | 0.1857 | 0.1857 |
| explosion/fire | 1039 | 0.0450 | 0.0436 | 0.0479 | 0.0505 | 0.095 | 0.0711 | 0.0711 |
| map | 1040 | 0.3516 | 0.3385 | 0.2802 | 0.3574 | 0.437 | 0.2094 | 0.2094 |
| US flag | 1041 | 0.0832 | 0.0871 | 0.0941 | 0.1460 | 0.058 | 0.1460 | 0.1460 |
| building exterior | 1042 | 0.1554 | 0.1615 | 0.1250 | 0.2143 | 0.245 | 0.3129 | 0.3129 |
| waterscape | 1043 | 0.1985 | 0.2050 | 0.1930 | 0.1879 | 0.278 | 0.2267 | 0.2267 |
| mountain | 1044 | 0.1128 | 0.1140 | 0.1190 | 0.0909 | 0.221 | 0.0944 | 0.0944 |
| prisoner | 1045 | 0.0050 | 0.0004 | 0.0020 | 0.0004 | 0.000 | 0.0006 | 0.000 |
| sports | 1046 | 0.2648 | 0.2489 | 0.2677 | 0.2920 | 0.263 | 0.4067 | 0.4067 |
| car | 1047 | 0.1314 | 0.1303 | 0.1098 | 0.1241 | 0.200 | 0.2184 | 0.200 |
| 10 Benchmark Concepts | | 0.1497 | 0.1428 | 0.1314 | 0.1587 | 0.1984 | 0.1872 | 0.1853 |

**Table 12: Revised (bug-fixed) TRECVID 2005 results of JHU systems.**

These and other topics will be pursued in the near future.

# 7. REFERENCES


[1] J. Argillander, G. Iyengar, and H. J. Nock. Semantic Annotation of Multimedia using Maximum Entropy Models. In *Proc. ICASSP*, March 2005.

[2] J. Bilmes and C. Bartels. On triangulating dynamic graphical models. *UAI*, 2003.

[3] J. Bilmes and G. Zweig. The graphical models toolkit: An open source software system for speech and time-series processing. *ICASSP*, Jun 2002.

[4] D. M. Blei and M. I. Jordan. Modeling Annotated Data. In *Proc. ACM SIGIR*, pages 127–134, 2003.

[5] S. Dennis, T. Landauer, W. Kintsch, and J. Quesada. Introduction to latent semantic analysis.

[6] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *Proc. ECCV*, volume 4, pages 97–112, 2002.

[7] A. Ghoshal, P. Ircing, and S. Khudanpur. Hidden Markov Models for Automatic Annotation and Content-Based Retrieval of Images and Video. In *Proc. ACM SIGIR*, pages 544–551, 2005.

[8] D. Hawking, T. Upstill, and N. Caswell. Toward better weighting of anchors.

[9] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In *Proc. ACM SIGIR*, pages 119–126, 2003.

[10] C. Petersohn. Fraunhofer hhi at trecvid 2004: Shot boundary detection system. In *TREC Video Retrieval Evaluation Online Proceedings*. TRECVID, 2004.

[11] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

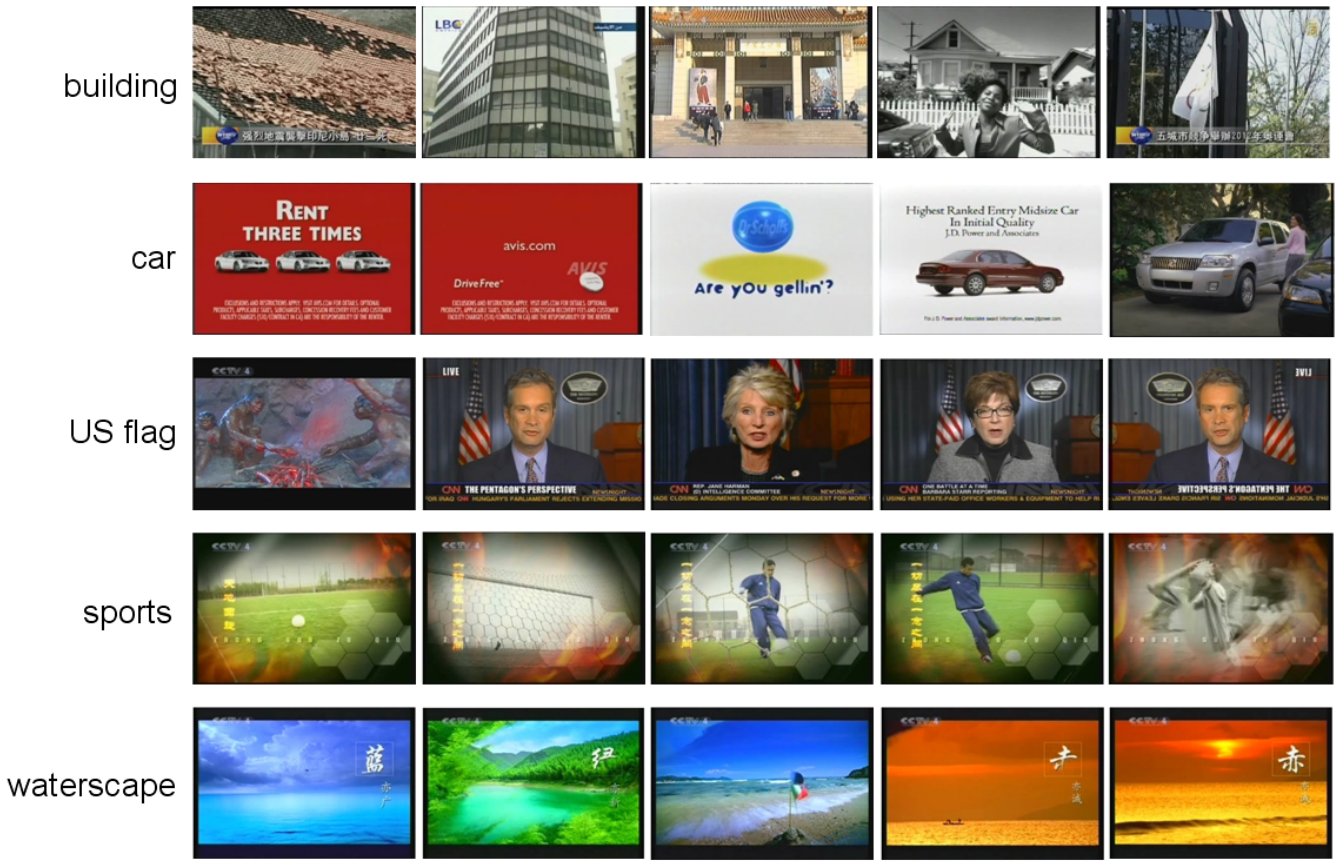


**Figure 2: Representative keyframes from the top-5 retrieved shots from the check set for select benchmark concepts.**