IEEE International Conference on Image Processing, Barcelona, Spain, September 2003

VideoAL: A Novel End-to-End MPEG-7 Video Automatic Labeling System

Ching-Yung Lin, Belle L. Tseng, Milind Naphade, Apostol Natsev and John R. Smith

IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA

ABSTRACT

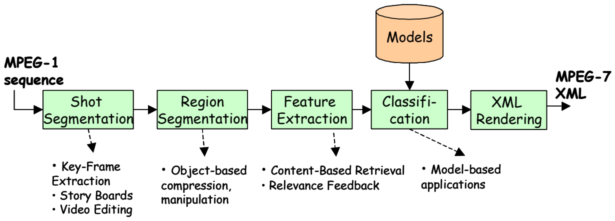
In this paper, we describe a novel end-to-end video automatic labeling system, which accepts MPEG-1 sequence inputs and generates MPEG-7 XML metadata files based on the prior established anchor models. Seven modules were developed for the system: Shot Segmentation, Region Segmentation, Annotation, Feature Extraction, Model Learning, Classification, and XML Rendering. The performance of this system has been tested in the NIST TREC-2002 video concept detection benchmark. The proposed system performs best in the mean average precision out of 18 worldwide participants.

1. INTRODUCTION

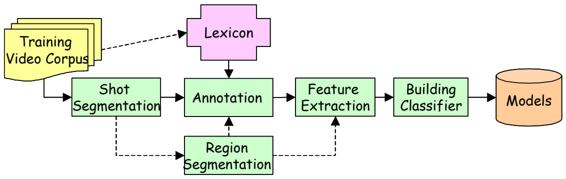
Recent years have witnessed an explosive growth in the generation and dissemination of multimedia data. While early research in the field involved extracting novel low-level features from the data set, a need for syntactic and semantic understanding is currently driving research paradigm into exploiting techniques from disparate disciplines that include signal processing, machine learning, computer vision, speech and sensor fusion. The focus is shifting towards extracting semantics from multimedia content. More detailed discussion on the history of semantic concept detection developments can be found in [6].

In this paper, we propose a novel end-to-end system that detects video concepts, such as *outdoors, indoors, sky, etc*., from MPEG-1 video sequences and automatically generate MPEG-7 metadata files. This system helps to extract semantic concepts from multimedia via generating a set of anchor concept detectors. We have generated 49 visual concept detectors. Combining with domain knowledge or visual grammars, these anchor detectors can serve as a basis for more generic semantic concept detection. The performance of this Video Automatic Labeling (VideoAL) system has been tested in the National Institute of Standards and Technology (NIST) TREC-2002 video concept detection benchmark. Among 18 submitted systems, the proposed system has the highest average mean average precision.

The paper is organized as follows. Section 2 describes the system overview. Section 3 covers the details of the modules for model training process. Section 4 describes the concept detection modules. In section 5, we show experimental results of the system and compare it to other concept detection systems.



(a)



(b)

Figure 1:System overview of VideoAL: (a) concept detection process and labeling modules; (b) training process and model-building modules

2. SYSTEM OVERVIEW

Seven modules were included in the whole system: *Shot Segmentation*, *Region Segmentation*, *Annotation*, *Feature Extraction*, *Model Learning*, *Classification*, and *MPEG-7 XML Rendering*. As shown in Figure 1, both model training and concept detection processes use five modules out of these seven modules. To learn concept models, in the first step, shot boundary detection is performed on the training video set. Semantic labels are then associated with each shot or regions using the annotation module. A feature extraction module extracts visual features from shots in different spatial-temporal granularity. Finally, a concept-learning module builds models for anchor concepts, *e.g*., outdoors, indoors, sky, snow, car, flag, *etc*.

The block diagram in Figure 1(a) shows the steps to perform the automatic semantic labeling process. The first three modules are the same as those modules of the training process. After features are extracted, a classification module tests the relevance of shots with the anchor concept models and results in a confidence value for each concept. These output concept values are then described using the MPEG-7 XML format. Only high confidence-value concepts are used to describe the content. This process is executed automatically from MPEG-1 video stream to MPEG-7 XML output. Also, there are side products from this process. For instance, we have used the result of shot segmentation to generate story boards, the result of region segmentation for MPEG-4 compression, and the low-level visual features for indexing on content-based retrieval.

3. TRAINING PROCESS AND MODEL-BUILDING MODULES

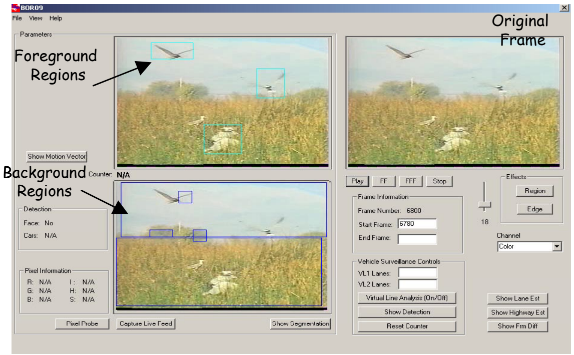
3.1 Shot Boundary Detection Module

Two algorithms can be selected for the shot boundary detection module. We developed a compress-domain based algorithm that can detect shot boundaries in 0.1x real-time. This algorithm uses sampled RGB color histograms in the I- and motion histograms in the P- frames of video sequences. Heuristic rules are designed to make the algorithms robust to flashes and noises. The second algorithm is developed by Amir *et. al*. for the

IBM CueVideo system[2]. This algorithm uses RGB histograms to compare pairs of frames that are one, three or seven frames apart. Statistics of frame differences are used to compute the adaptive thresholds. It classifies shot boundaries into Cuts, Fade-in, Fade-out, Dissolve and Other. In both algorithms, we select the middle frame of the shot as its keyframe.

3.2 Region Segmentation Module

We built an automatic segmentation system for visual background and object segmentation. It can be executed in real-time, including MPEG-I decoding, foreground object segmentation and background segmentation, at a PC with Intel Pentium III 750MHz CPU, 512M RAM and Windows 2000 OS. To segment background scene objects, we use a block-based region growing method on each decoded I- or P- frames in the video clip. The criteria of region growing are based on the color histogram, edge histogram, and directionality of the block. Five largest background regions per frame were used for later modules. For foreground object segmentation, we use a spiral searching technique to calculate the motion vectors of I- and P-frames, and use them to determine objects with region growing in the spatial domain and additional tracking constraints in the time domain. MPEG compressed-domain motion vectors were tested in our system. However, in most of our experiments, they are too noisy to generate reliable results. We also found that combining motion vectors, color, edge, and texture information does not usually generate better segmentation results than using only the motion info. Up to ten foreground objects are reported. For each region, only the coordinates of a rectangular bounding box, which fully contains the region, are reported to later modules.



Counter: N/A

Face: No
Cars: N/A

R: N/A I: N/A
G: N/A H: N/A
B: N/A S: N/A

Frame Number: 6800
Start Frame: 6780
End Frame:

Channel: Color

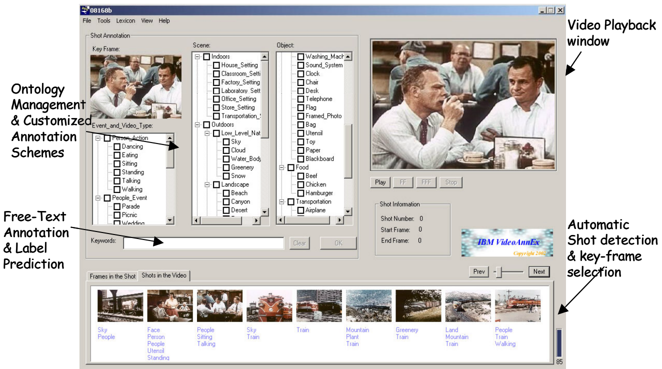
VL1 Lanes:
VL2 Lanes:

Figure 2: Real-Time Video Region Segmentation Tool

3.3 Annotation Module

In order to train visual concept models, the system requires labels being associated with training videos. We implemented a *VideoAnnEx* MPEG-7 annotation tool for authors to annotate video content with semantic descriptions[4]. It is one of the first MPEG-7 annotation tools being made publicly available. The tool explores a number of interesting capabilities including automatic shot detection, key-frame selection, automatic label propagation to similar shots, and importing, editing, and customizing of ontology and controlled term lists.

Given the lexicon and video shot boundaries, visual annotations can be assigned to each shot by a combination of label prediction and human interaction. Labels can be associated to a shot or a region on the keyframe. Regions can be manually selected from the keyframe or injected from the segmentation module. Annotation of a video is executed shot by shot without permuting their time order, which we consider an important factor for human annotators because of the time-dependent semantic meanings in videos. Label prediction utilizes clustering on the keyframes of video shots in the video corpus or within a video. By the time a shot is being annotated, the system predicts its labels by propagating the labels from the last shot in time within the same cluster. Annotator can accept these predicted labels or select new labels from the hierarchical controlled-term lists. All the annotation results and descriptions of ontology are stored as MPEG-7 XML files.



Shot Number: 0
Start Frame: 0
End Frame: 0

Figure 3: *VideoAnnEx* MPEG-7 Video Annotation Tool.

3.4 Visual Feature Extraction Module

The system extracts two sets of visual features for each video shot. These two sets are applied by two different modeling procedures that are described in Section 3.5. The first set includes: (1) color histogram (YCbCr, 8x3x3 dimensions), (2) Auto-correlograms (YCbCr, 8x3x3 dims), (3) Edge orientation histogram (32 dim), (4) Dudani's Moment Invariants (6 dims), (5) Normalized width and height of bounding box (2 dims), (6) Co-occurrence texture (48 dims). These visual features are all extracted from the keyframe.

The other set of visual features include: (1) Color histogram (RGB, 8x8x8 dims), (2) Color moments (7 dims), (3) Coarseness (1 dim), (4) Contrast (1 dim), (5) Directionality (1 dim), and (6) Motion vector histogram (6 dim). The first five features are extracted from the keyframe. The motion features are extracted from every I and P frames of the shot using the motion estimation method described in the region segmentation module.

Depending on the characteristics, some of the concept are consider as global, such as outdoors, indoors, factory setting, and office setting, while some of them are regional, such as sky, mountain, greenery, and car. Therefore, we extract these features on both the frame level and the region level.

3.5 Model-Learning Module

The architecture of the learning process of the models for the system is described in Figure 4. This architecture includes three main components: two different modeling procedures and a fusion process. These two modeling procedures are different in the use of different visual features and the use of features from different video sets. Both procedures use Support Vector Machine (SVM) as the classification methods [3], based on the performance results of our prior experiments. Because training classifier at a large data set is very time consuming, we did not use cross-validation for modeling. We partition a training video corpus into two sets: *Model Training* (MT) set and *Model Validate* (MV) set. The MT set is mainly used for training classifiers and the MV set is mainly used to verify the performance of individual classifiers for the selections of parameters on the fusion process. This set is also used for the SVM parameter selection (that includes *kernels*, *variance*, *margin*, and *cost factors*) on the first modeling procedure.

The first process uses the MT to training classifiers and uses MV to select the best parameters for each individual SVM classifier. This process generates a SVM classifier for each type of visual feature, *e.g.*, one classifier based on color histogram, one based on moments, and so forth. It also generates SVM classifiers based on heuristic combinations of features. Here, different kinds of features are cascaded to form larger dimensional feature vectors for training SVM. Each classifier is run on the MV set. The resulting list is then sorted based on the signed distance of each MV example from the separating hyperplane. For each feature and each feature combination, we then choose that parametric combination which resulted in the highest non-interpolated Average Precision [7] in the MV set. Because various parameter combinations are tested, this process down-samples the MT set to reduce overall training time. The input of this procedure are 6 feature vector sets ($f_1,\ldots,f_6$), and the output is 6 to 12 sets (m_1, …, m_{12}) of confidence values that are the results of these 6 to 12 models on the MV set.

The second modeling procedure is similar to the first procedure, except that a different set of visual features is used and the MV set is not used for parameter selection. In this procedure, fewer combinations of parameters are used (only kernel selections) and MT set is not down-sampled. This procedure takes the second visual feature vector sets ($f_7,\ldots, f_{12}$), and the output is 6 to 10 sets (m_{13}, …, m_{22}) of confidence values of the video shots at the MV set.

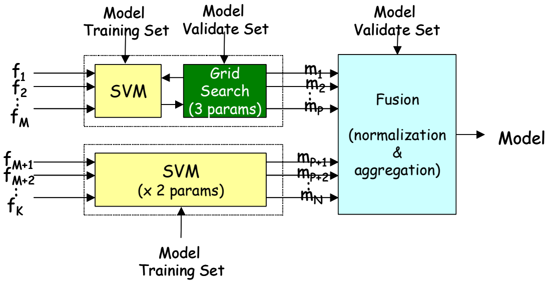


Figure 4: Separate classifier models build on different feature sets are combined in the classifier fusion.

The normalized ensemble fusion process consists of three major steps. The first step is normalization of resulting confidence scores from each classifier. The second step is the aggregation of the normalized confidence scores. The third step is the optimization over multiple score normalization and fusion functions to attain optimal performance.

Details of the fusion procedure are as follows. Each classifier generates an associated confidence score for the data in the validation set. These confidence scores are normalized to a range of [0, 1]. The normalization schemes include: (1) rank normalization, (2) range normalization, and (3) Gaussian normalization. After score normalization, the combiner function selects a permuted subset of different classifiers and operates on their normalized scores. We essentially identify the high-performing and complementary subsets of classifiers. Following, different functions to combine the normalized scores from each classifier are considered. The combiner functions we tried include: (1) minimum, (2) maximum, (3) average and (4) product. Subsequently, an optimal selection of the best performing normalized ensemble fusion is obtained by evaluating the average precision (AP) measure against the MV ground truth.

After the modeling stages, confidence values on each anchor concept toward each keyframe or background/foreground region can be generated at the MV set. For those concepts that are in the hierarchical root in the ontology, *e.g.*, outdoors, we then compare its AP values with a weighted aggregation on the confidence values from its region-based child concepts, *e.g.*, sky, mountain, *etc*. This comparison is used to select the final model to be used for each concept.

4. DETECTION PROCESS AND LABELING MODULES

Shown in Figure 1(a), the detection process takes MPEG-1 video sequence inputs and automatically generates MPEG-7 semantic labeling metadata XML files. The first three modules have been described in Section 3.

4.1 Classification Module

The classification module is similar to the model-learning module described in Section 3.5. The only difference is that the classification module takes the visual features of the test set, applies SVM classification in both procedures and uses the fusion method that has been selected in the training process.

4.2 MPEG-7 XML Rendering Module

In this module, MPEG-7 XML schemas can be plug-in to convert the confidence values from the output of classification module to MPEG-7 XML files. Based on a user-identified threshold, the system reports only those anchor concept with higher confidence values. The confidence values are associated with a video shot or regions on the keyframe. Sample XML outputs can be found in [4].

5. EXPERIMENTAL RESULTS

5.1 Overview of NIST TREC 2002 Concept Detection Task

The goal of TREC-2002 Video Track was to promote progress in content-based retrieval from digital video via open, metrics-based evaluation [7]. This year the track used 73.3 hours of publicly available digital video (MPEG-1) from the Internet Archive and the Open Video Project. The material comprised advertising, educational, industrial, and amateur films produced between the 1930's and the 1970's. Sixteen teams participated in one or more of three tasks: shot boundary determination, concept detection, and search. Results were scored by NIST using manually created truth data.

The concept detection benchmark was as follows. 23.26 hours (96 videos containing 7891 standard shots) were randomly chosen from the corpus, to be used solely for the development of concept detectors. 5.02 hours (23 videos containing 1848 standard shots) were randomly chosen from the remaining material for use as a Feature Test Set. (Concept detection task is officially called "Feature Extraction" by NIST). Given a standard set of shot boundaries for the Feature Test set and a list of concept definitions, participants were to return for each concept a list, at most the top 1000 video shots, ranked according to the highest possibility of detecting the presence of the concept. The ground-truth of the presence of each concept was assumed to be binary, i.e., it is either present or absent in a video shot. Ten concepts were defined in this benchmark: *Outdoors, Indoors, Face, People, Cityscape, Landscape, Text Overlay, Speech, Instrumental Sound*, and *Monologue*. They are defined by textual descriptions, *e.g*. the definition of text overlay is: "Video segments that contain superimposed text large enough to be read." Seven out of ten concepts are purely visual.

5.2 Comparisons on the system results

Thirteen groups participated the concept detection benchmark, including teams from Microsoft Research Asia, Carnegie Mellon University, University of Maryland, and other institutes from all over the world. Each group can submit one or more runs of detection result. Totally, eighteen systems were submitted to NIST for the benchmark. Some systems may not cover all 10 concept detectors

Detailed description of the individual systems can be seen in [7] and the paper published on the proceedings of TREC 2002 conference. For instance, Microsoft Asia implements an AdaBoost classifier based on color moments and edge direction histograms. MediaTeam Oulu and VTT used edge detection gradients in their feature-based classification. Fudan University found that: color and edge direction histograms with K-nearest neighbor works well for the Indoor/Outdoor concept; while skin-color segmentation and shape filtering is selected for Face/People detection; and neural network contributes to their text overlay detection.

The evaluation results are plotted in Figure 5, which shows Average Precision measured at a fixed number of documents (1000 for the test set) The "Avg." bars correspond to the performance averaged across all participants. The "Best" bars correspond to the system returning the highest Average Precision. The "IBM" bars correspond to the results of the proposed system. This system[1] performed relatively well giving highest Average Precision on 5 of the 7 visual concepts[2].

An example precision-recall curve is shown in Figure 6. Our outdoors detector performed an average precision of 0.609 on the NIST Feature Test Set. In addition to the seven visual concept detectors, we have developed 42 other visual detectors for this system. The precision values of several of these detectors are shown in Figure 7.

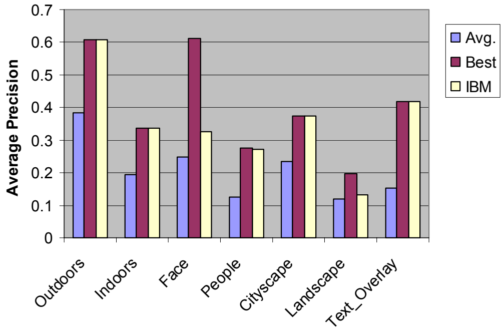


Figure 5: TREC-2002 visual concept detection performances in average precision.

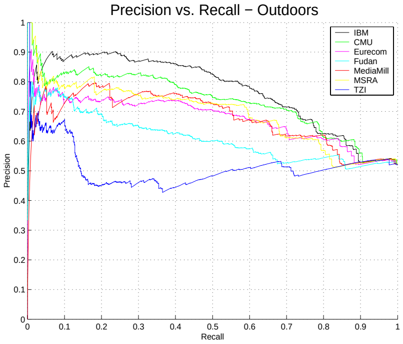


Figure 6: Precision-recall curves of the Outdoors detector on the benchmarking. Only the best system form each institute is shown.

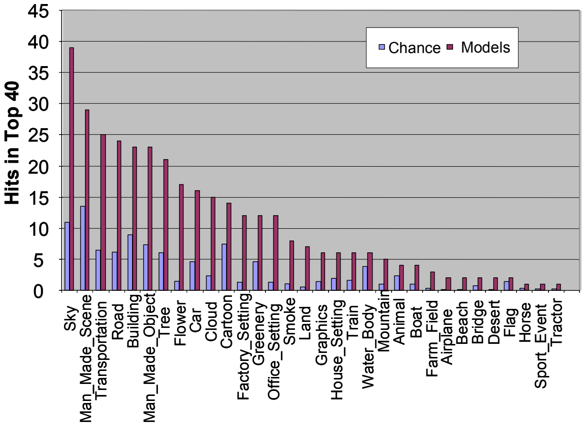


Figure 7: Precision of several other visual concept detectors.

6. CONCLUSION

We presented a novel end-to-end MPEG-7 video automatic labeling system. The system explores fully-automatic content analysis methods for shot detection, multi-modal feature extraction, statistical modeling for semantic concept detection. We described the experimental runs that are part of the TREC-2002 video concept detection benchmark. Our next step will focus on increasing the number of anchor concept detectors, developing methodology for generic concept detection, enhancing precision and decreasing the training requirements.

7. REFERENCES

[1] B. Adams *et. al*., "IBM Research TREC-2002 Video Retrieval System," Proc. of Text Retrieval Conference, Gaithersburg, MD, Nov. 2002.

[2] A. Amir *et. al*., "CueVideo Toolkit Version 2.1," http://www.almaden.ibm.com/cs/cuevideo/.

[3] T. Joachims, "SVM-light," http://svmlight.joachims.org.

[4] C.-Y. Lin, B. Tseng and J. Smith, "VideoAnnEx: IBM MPEG-7 Annotation Tool for Multimedia Indexing and Concept Learning," Proc. of ICME, Baltimore, Jul. 2003.

[5] M. Naphade, S. Basu, J. Smith, C.-Y. Lin, and B. Tseng. "Modeling semantic concepts to support query by keywords in video," ICIP 2002, Rochester, NY, Sept. 2002.

[6] M. Naphade, "Statistical Techniques in Video Data Management," MMSP 2002, Virgin Islands, Dec. 2002.

[7] A. F. Smeaton and P. Over, "The TREC-2002 Video Track Report," Proc. of Text Retrieval Conference, Gaithersburg, Maryland, Nov. 2002.

[1] Extra face detector developed by Giri Iyengar was applied on the final fusion of face detector. Two different types of classifiers developed by Chitra Dorai *et. al.* and Dongqing Zhang *et. al.*, respectively, are applied on the text overlay detector[1].

[2] For the people detector, our system performs 2nd among all systems but has the best AP among systems that did not use knowledge of testing content. Two out of 18 submitted systems use this knowledge.