RC23653 (W0507-026) July 5, 2005
Computer Science

IBM Research Report

Semantic Routing and Filtering for Large-Scale Video Streams Monitoring

Ching-Yung Lin, Olivier Verscheure, Lisa Amini
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. Ithas been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distributionoutside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g, payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home

SEMANTIC ROUTING AND FILTERING FOR LARGE-SCALE VIDEO STREAMS MONITORING

Ching-Yung Lin, Olivier Verscheure and Lisa Amini

IBM T. J. Watson Research Center
19 Skyline Dr., Hawthorne, NY 10532, USA
{chingyung, ov1, aminil,}@us.ibm.com

ABSTRACT

In this paper, we propose a semantic routing and filtering framework for large-scale monitoring of video streams. Our goal is to build a distributed system that at any given time is capable of simultaneously monitoring the content of multiple video streams being transmitted over the Internet (or a proprietary network). A key design requirement of such a system is the ability to handle tens of gigabytes of multimedia data per second. Traditional techniques have an important limitation. Once a bottleneck in terms of CPU or storage is reached, data is dropped indiscriminately.. In this paper, we propose distributed real-time semantic filters to route and filter video data. We propose a mechanism to alter the accuracy of classification with the complexity of execution; thus avoiding system failure during periods of overload. We propose a set of novel video features that perform better than our previous semantic classifiers. This system is capable of classifying over a hundred concepts. Experiments on 190 hours of pre-stored and live video streams validate the effectiveness of the proposed system.

1.INTRODUCTION

Monitoring user-interested information from large-scale video streams is a challenging problem. For instance, an intelligence agent may need to monitor foreign country military or political activities from hundreds of live broadcasting video channels. An administrator of video sensor system may needs to monitor the activities of hundreds of video cameras mounted on cars or soldiers and understand the context environment of them. Multimedia entertainment industry may want to monitor Internet traffics to know whether their movies are illegally distributed through the Internet. To achieve the above mentioned goals, a system needs to process large amount of data, understand/classify their semantic content meanings (e.g., what kind of scenes is the soldier looking, what type of videos is being played through an Internet source, where a foreign broadcasting news is mentioning the activities of their leader, etc.) in real-time, and route the interested information (e.g., videos) to the agent or the field commander for intelligence analysis or decision making. In these scenarios, the targeted video contents may range in a scale of tens of gigabits per second, and the system has to be able to conduct semantic classification on these video streams in real-time. Unfortunately, to the best of our knowledge, no existing system can achieve such a task. The major challenges reside in the needs for (1) the bandwidth of streaming videos for routing multimedia data to the classifiers and (2) the real-time requirement for semantic detection. In these application cases, traditional indexing and semantic concept detection techniques developed for databases usually cannot be easily extended for the dynamic nature of streams. Recently, real-time stream information classification is also getting more attention on other modalities (such as email activities, chat room monitoring, VoIP monitoring, etc.) because of its inherited challenges on the speed of classification, routing of information, etc.

Traditional approaches usually rely on the storage-and-process analyses. However, these techniques have their own limitations. Once the data amount or CPU/power/memory reaches a certain threshold, these systems tend to break down entirely. Therefore, the challenge here will be: *how can a system route or filter transmission video packets based on the semantic contents in a speed much faster enough and flexible under various resource constraints*? In [1], Madden et. al. use the semantic routing tree to route signal-level information on a resource-constrained sensor network. Routing is based on the signal properties and pre-defined decision trees. Comparing with [1], multimedia streaming data is more difficult to route/filter and detect concepts. Even in the raw video data domain without any resource constraint, video semantics detection is still an open issue [2].

The scope of this paper is to provide an overview of a novel semantic filtering and routing system that can be applied to large-scale content monitoring. Because of the page limit, we will have to leave algorithmic details to future reports. In this paper, we show a semantic filtering system that reduces the amount of transmission loads or routing video content packets based on semantic detection. We propose to use the complexity-accuracy curves to dynamically change the operating point so as to accommodate for the possibly varying amount of incoming data and/or processing resources. We also utilize a set of novel video features that results in better performance. Experiments validate the effectiveness of the proposed system.

This paper is organized as follows. In Section 2, we introduce our system architecture. In Section 3, we describe a novel feature set and a technique to configure classifiers for resource-constrained environments. Experiments are shown in Section 4. Section 5 concludes this work.

2. SYSTEM ARCHITECTURE

As shown in Figure 1, a new video semantic filtering system is implemented based on the novel CDS real-time classifiers. In this system, the feature extraction Processing Elements (PEs) and the display module do not need to reside on the same machine. This system handles TV broadcasts, VCR, DVD discs, video file databases, and webcam inputs. Our middleware extracts the shot-based features and sends those features to a server machine which implements one hundred concept detectors. A control module is used to match the user interests with the confidence output of the semantic model vectors. Then, the similarity values are stored as metadata and sent back to the display module to filter the content.

A PE is an independent executable thread which has specific ports for input and output streams. Thus, PEs can be distributed in different machines. In this system, we placed the GOP, feature extraction and shot segmentation functions on the client machine, which resides in a smart camera or an edge router in the network. There can be tens of such distributed (and parallel) clients sending feature packets to the distributed server classifiers. For each shot, a CDS feature packet, which is less than 1.4K bits is sent to a server router which multicasts these feature packets to the classifier PEs. Because the feature rate is less than 2.8 Kbps, the transmission load is only 56 kbps if the server PEs need to classify 200 video streams simultaneously. In our experiments, even if all classifier PEs are placed in one machine, a regular Pentium 2.4 GHz with 1GB RAM server can deal with 40 concurrent incoming streams in real-time with one hundred concept detectors.

3. REAL-TIME VIDEO SEMANTIC FILTER

3.1 Compressed-Domain Slice Features

In this paper, we propose a new feature set that results in better accuracy of concept classifiers with a shorter extraction time, comparing to the work in [2] and [3]. The reduction in computational load is significant: for a typical 320x240 MPEG-1 video stream, a decoder needs about 2 million multiplications to decode an I-frame. Also, the algorithms in [2, 3] require complex region/object segmentation and compute 12 sets of features. Feature selection of previous system was about 3 times slower than real-time, which was the bottleneck for real-time large-scale implementations.

We denote this new set of features as *Compressed-Domain Slice* (CDS) features. This feature set is extracted as follows:

1. *Parse the MPEG-1/2 packets and get the beginning of an I-frame or the closest I-frame of a pre-specified shot keyframe.*
2. *Using the VLC maps to map variable-length codes to the DCT-domain coefficients.*
3. *Within am MPEG slice or a union of slices, truncate selected DCT coefficients and calculate the histogram of these DCT coefficients.*
4. *Form a feature vector of the frame based on the histogram coefficients with multiple slices.*

In the above procedure, we can see that no multiplication operation is required to get these feature vectors. Only addition is needed for getting the histogram. In a typical situation, we partition a frame into three slices, and use the histograms from 3 DCT coefficients (1 DC and 2 lowest frequency AC coefficients) on each color plane Y, Cb and Cr. This forms a 576-dimensional feature vector. As in [2], we use SVM to train models and classification. In our experience, we notice that fusion of different sets of

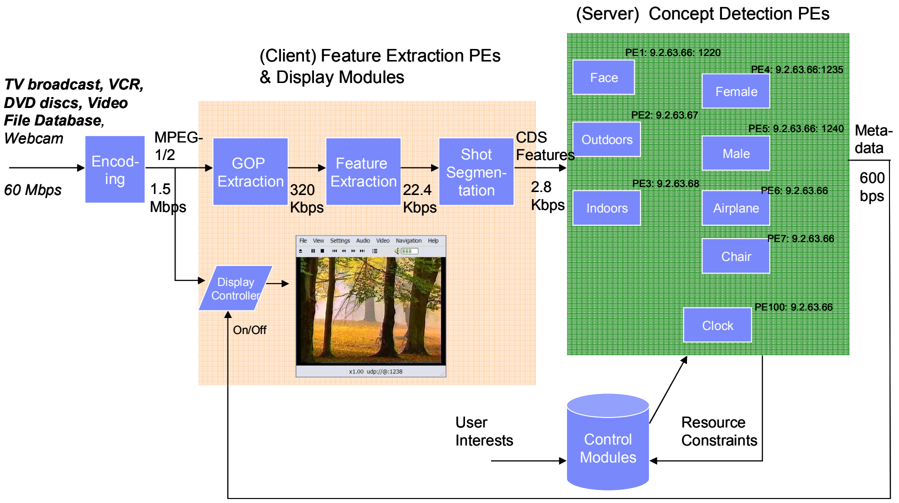

Figure 1: System Diagrams of Semantic Filtering on Distributed Systems

features (color, edge, motion, texture) is an open issue, which is also application-dependent. Complicated (feature or classifier) fusion does not necessary lead to better results [2]. These CDS features can be considered as an early-fusion method for classification.

3.2 Complexity-Accuracy Curves

Our goal is to find out whether specific types of classifiers can perform relatively well under all kinds of resource constraints, because many classification systems may not be able to do so. Suppose we are given training data $\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$, where $x_i \in \mathcal{X}$ denotes the input patterns, and $y_i \in \{-1,1\}$ denotes the binary labels. The goal for a supervised classifier is to find a function $f(x)$ that has at most ε deviation from y_i for all the training data, and is as flat as possible. In the training stage, if using SVM, the models can be built based on different kernel functions and cost ratios of error margin on positive and negative examples. The SVM classifier is the functional of:

$$f(x)=\sum_{i=1}^{S} a_i \cdot k(x,x_i)+b \tag{1}$$

where S is the number of support vectors, k(.,.) is a kernel function, e.g., the Gaussian kernel ,

$$k(x,x_i)=e^{-\frac{|x-x_i|^2}{r}} \tag{2}$$

and a_i's are the weightings of SVs and b is a constant threshold value. The goal of SVM is to find a hyperplane which best separates training examples with the minimum cost. The kernel function can be considered as a distance function between unknown vectors and SVs.

In the distributed system with independent PEs, we require PEs to be able to switch among various operating points with little overhead. One solution is to generate embedded classifiers. For different operating points, the lower complexity classifiers are subsets of high complexity classifiers w/o a few parameters' updates. For instance, from (1) and (2), we know that the complexity of SVM-based classifiers depends on the kernel, the feature dimensions and the number of support vectors. Regardless of the storage and I/O access requirements, if we consider the complexity c as the number of operations (multiplications, additions) required for classification, then the resource needed for such computation is:

$$c \propto S \cdot D \tag{3}$$

where D is the dimensionality of the feature vector, and S is the number of support vectors. The Processing Element (PE) achieves various operating points of the C-A curve by controlling the number of features to extract and the number of support vectors by setting unneeded SVs to zero. In our system, we assume models were only trained once without resource constraint consideration or models may be provided by third-party provider. Thus, the system can only generate these CA curves based on existing classifiers. We used four methods to determine these curves: selecting n SVs with n max a_i, $|a_i|$, randomly select, or clustering on SVs. The first three are embedded classifiers, while the fourth method is not. Operation points are determined by off-line training using a validation set.

If training samples are available, the system may use other methods, e.g., ν-SVM [4] with pre-determined thresholds on SVs and error margin. However, this shall cause additional system I/O load while switching between different operating points.

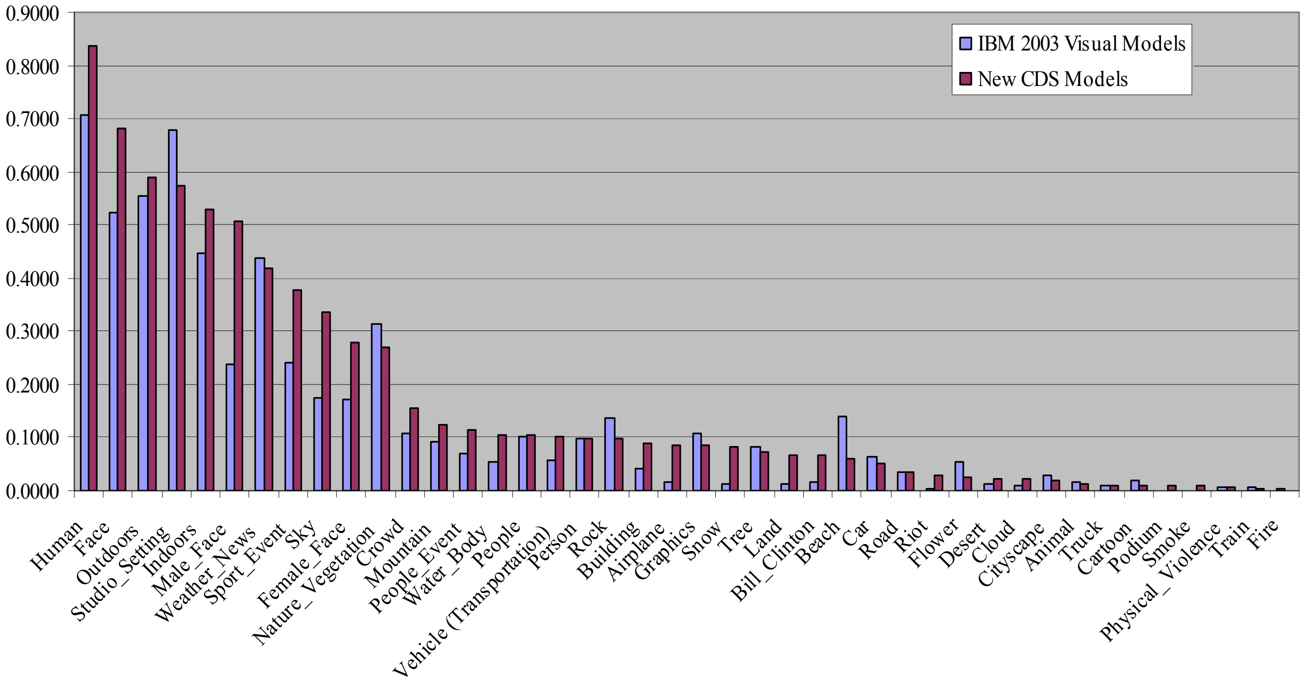

Figure 2: Comparison of the Average Precision of New CDS Visual Models with IBM 2003 Visual Models.

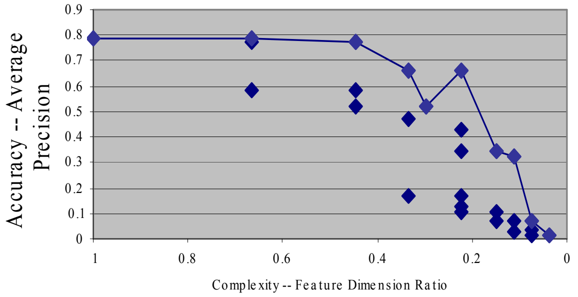

Fig. 3: Complexity-Accuracy curve on the feature vector dimension parameter of the "Weather_News" filter at CV set.

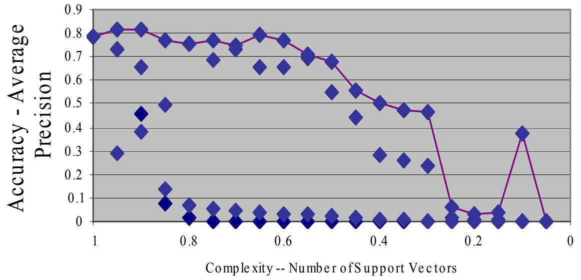

Fig. 4: Complexity-Accuracy curve on changing the number of support vectors of the "Weather_News" filter at the CV test set.

4. EXPERIMENTAL RESULTS

We mention the speed and the scalability of the system in Section 2. In Section 4, we demonstrate the accuracy of our semantic filtering classifiers and the adaptability of them. We used the NIST TRECVID 2003 corpus. In our experiments, we use the 62 hours of development set, which has been manually annotated with 133 audio and video concepts [5]. This set was divided into four parts: CR (38 hrs), CV (6 hr), CF1 (6 hr) and CF2 (12hr). As in [2], we train our visual models using the CR set, and select the modeling parameters using the CV set. The models are then tested on the unseen CF1 and CF2 sets. Note that the manual annotation of CF1 and CF2 sets is only used for measuring the system performance. We use the Average Precision (AP) value, which is the integral area under the precision-recall curve to measure the accuracy. AP is usually used by NIST to provide a single-value metric of the P-R curve. *Mean Average Precision* (MAP) is used by averaging the AP values of a system across all testing concepts [2] to compare the performance of systems.

The 576-dimensional CDS feature vectors of the 28,055 keyframes in the CR set were used for training. Each visual concept is trained independently. Positive examples of a training set were selected, if a shot is annotated with this label or any children label in the hierarchical semantic tree [5]. All other shots are considered as negative. The negative examples are sub-sampled by a constraint of maximum negative-positive ratio of 5:1. For each concept, 9 models of a hybrid of 3 different kernels (linear, polynomial, and Gaussian) and 3 cost functions (1, 10, and 100) are trained. A demo is at http://www.research.ibm.com/VideoDIG.

A performance comparison between the new models and the IBM 2003 visual concept models is shown in Figure 2. The IBM visual concept models were fused with the speech-based detectors to form the IBM multi-modality detectors that performed best in the TRECVID 2003 [2]. In 2003, 42 visual models were internally extensively evaluated using the CF2 set, with an MAP of 0.1404. The MAP of the corresponding 42 models based on the new CDS features is 0.1705, which is 21.48% better. If we only consider the 13 visual detectors specified by NIST, the gain of MAP values is 23.6% (0.2091 v.s. 0.1692).

In Figures 3 and 4, we show the accuracy and complexity curve of some preliminary experiments. In both cases, classifiers are all embedded, thus, only simple coefficient masking is used in the run-time system operations. In Figure 3, we show that if we reduce the dimensionality of feature vectors, the AP of classifier varies. For instance, if the system operates at 22% of the original resources (in terms of time and storage), then it can achieve an AP of 0.658, which is 83% of the best accuracy. For each complexity value, there may be several accuracy points available due to different feature dimension reduction techniques. E.g., in the above case, the operating point was selected with the feature values from all 3 slices, 1 set of color histograms (i.e., gray-level) and 2 sets of textures (i.e., 1 DC histogram and 1 AC histogram).

In Figure 4, we show an example of the accuracy-complexity curve based on the reduction of number of SVs. This model has 440 SVs. We see that, with 50% of the SVs, the classifier achieves 86.6% of the original accuracy. Similarly, there could be several operating points for each reduction ratio.

5. CONCLUSIONS

In this paper, we proposed a novel semantic routing/filtering system for large-scale video monitoring and reducing the amount of transmission loads. We also showed a set of novel visual features, which results in significant gains in both speed and accuracy. Complexity-accuracy curves for optimally choosing operating points are used. We shall investigate more effective multi-modal features and develop algorithms for user-profile management, graph management and planning issues.

6 REFERENCES

[1] S. Madden, M. Franklin, J. Hellerstein, W. Hong, "The Design of an Acquisitional Query Processor for Sensor Networks, " SIGMOD, San Diego, CA, June 2003.

[2] A. Amir, *et. al*., "IBM Research TRECVID-2003 Video Retrieval System," NIST TREC-2003 Nov. 2003.

[3] C.-Y. Lin, B. L. Tseng, M. Naphade, A. Natsev, J. R. Smith, "VideoAL: End-to-End System MPEG-7 Video Automatic Labeling System," ICIP 2003, Barcelona, Sept. 2003.

[4] B. Scholkopf, A. Smola, "New Support Vector Algorithms," NC2-TR-1998-031, Nov. 1998.

[5] C.-Y. Lin, B. L. Tseng and J. R. Smith, "Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets," NIST TRECVID Workshop, MD, Nov. 2003.