Detection and Extraction of Videos using Decision Trees

Sk.Abdul Nabi
Dept. of CSE
AVN Inst. of Engg. & Tech.
Hyderabad, India

Shaik Rasool
Dept. of CSE
S.C.E.T.
Hyderabad, India

Dr.P. Premchand
Dept. of CSE
University College of Engineering, OU
Hyderabad, India

***Abstract*— This paper addresses a new multimedia data mining framework for the extraction of events in videos by using decision tree logic. The aim of our DEVDT (Detection and Extraction of Videos using Decision Trees) system is for improving the indexing and retrieval of multimedia information. The extracted events can be used to index the videos. In this system we have considered C4.5 Decision tree algorithm [3] which is used for managing both continuous and discrete attributes. In this process, firstly we have adopted an advanced video event detection method to produce event boundaries and some important visual features. This rich multi-modal feature set is filtered by a pre-processing step to clean the noise as well as to reduce the irrelevant data. This will improve the performance of both Precision and Recall. After producing the cleaned data, it will be mined and classified by using a decision tree model. The learning and classification steps of this Decision tree are simple and fast. The Decision Tree has good accuracy. Subsequently, by using our system we will reach maximum Precision and Recall i.e. we will extract pure video events effectively and proficiently.**

Keywords- DEVDT; Data Processing; Data Pre-Processing; Decision Tree and Training Data.

I. INTRODUCTION

Over a period of time, data researchers have shown immense interest in the study of data mining. This is quite natural, as the database field started in the commercial community and this community still has much influence over the types of questions being studied. Digital multimedia differs from previous forms of combined media in that the bits that represent text, images, animations, audio, video and other signals can be treated as data by computer programs. One fact of this diverse data in terms of underlying models and formats is that it is synchronized and integrated. Hence, it can be treated as integral data records. Virtual communities (in the broad sense of this word, which includes any communities mediated by digital technologies) are another example where generated data constitutes an integral data record. Such data may include data about member profiles, the content generated by the virtual community and communication data in different formats including email, chat records, SMS messages and video conferencing records. Not all multimedia data is so diverse. An example of less diverse but larger in terms of the collected amount is the data generated by video surveillance systems [2 , 4] where each integral data record roughly consists of a set of time-stamped images – the video frames. In any case, the collection of such integral data records constitutes a multimedia data set. The challenge of extracting meaningful patterns from such data sets has led to the research and development in the area of multimedia data mining.

Multimedia databases are widespread and multimedia data sets are extremely large. There are tools for managing and searching within such collections but the need for tools to extract hidden useful knowledge embedded within multimedia data is becoming critical for many decision-making applications. The tools needed today are tools for discovering relationships between data items or segments within images, classifying images based on their content, extracting patterns from sound, categorizing speech and music, recognizing and tracking objects in video streams, relations between different multimedia components and cross-media object relations [1]. The overall design of a multimedia database differs markedly from that of a standard textual database. Browsing and querying in the former environment utilizes entities and attributes that are usually hidden from the casual user. Since many of the data mining tasks in standard databases concern associations between different attributes in a multimedia environment, the nature of the attributes over which associations are constructed becomes quite important.

II. MULTIMEDIA DATA MINING MODELS

Multimedia data mining is a challenging field due to the non-structured nature of multimedia data [5]. Such ubiquitous data is required, if not essential in many applications. Multimedia database design differs distinctly from that of a standard textual database. Browsing and querying in the former environment utilizes entities and attributes that are usually hidden from the casual user. Since many of the data mining tasks in standard databases concern associations between different attributes. In a multimedia environment, the nature of the attributes over which associations are constructed becomes quite important. For example, it is possible to mine a rule of the form like this: consumers who remove Brand A paper towels from the display and examine the package for at least 20 seconds will also purchase it [s%, c%] [6] using textual information which only appears in relational tables. This would entail having purchasing information entered into a back-end database via a point-of-sale terminal as well as a person observing for how long people examine various products and entering this information in the same database. However, this rule can also be mined by content-based retrievals of video(s) taken of the shopping experience of various shoppers during some period of time. For this latter approach, it is not obvious what attributes of the overall database are being used for the mining task. In order to construct a more detailed characterization of the different sorts of data mining in a multimedia database environment, we now address the general notion of multimedia data models.

There are many multimedia data models in the literature [7 , 8]. However, all such data models are similar at a high enough level of abstraction. They all should represent the following types of information

1. The detailed structure of the various multimedia objects.

2. Structure dependent operations on multimedia objects.

3. Multimedia objects properties.

4. Relationships between multimedia objects and real-world objects.

5. Portions of multimedia objects that have representation relationships with real-world objects, the representation relationships themselves and the methods used to determine them.

6. Properties, relationships and operations on real-world objects.

Modeling the structure of a multimedia object is important for many reasons, not the least of which is that various operations are defined on these objects which depend on its structure. These operations are used to create derived multimedia objects for similarity matching (edge maps) as well as various composite multimedia objects from individual component multimedia objects (multimedia presentations). An example of a multimedia object property is the name of the object. For example, 'Titanic' is the name of a particular video object. A relationship between a multimedia object and a real-world object would be the stars-in relationship between the actor Victoria Foyt and the video Titanic. Suppose that Titanic ship is a real-world object being represented in the database and that a particular region of frame six of the video Titanic is known to show this object. This small portion of the byte span of the entire video is also considered to be a first-class database object, called a semcon [9, 10], for iconic data with semantics. Both the many-one relationship represents which holds between this semcon and Titanic ship object, as well as the many-many relationship appearing-in which holds between the Titanic ship object and the video Titanic should be captured either implicitly or explicitly by any multimedia data model.

These relationships enable metadata mediated browsing. Such behavior is exhibited when one clicks a mouse whose cursor is over a semcon representing a Titanic ship. As a database object, this man-made structure is represented in the database by a tuple in the Monument table. Doing a join, we may then get tuples representing information concerning the designers of this structure. Finally, we may view images of containing pictures of these people. The relationship represents enables the database system to navigate from the semcon of the Titanic ship to the database tuple representing this structure, while the relationship appearing-in enables the system to navigate from tuples concerning the designers of the Titanic ship to their images. Semcons, as first-class database objects and have attributes. These attributes include various features extracted from them that can be used for similarity matching over other multimedia objects. Features should be first-class database objects as well and include such things as color histograms and texture maps. In a multimedia database environment, querying consists of utilizing semcons for searching for multimedia objects corresponding to the same real-world object. If two semcons have similar features, the semcons themselves are similar.

III. Decision Tree Model

A Decision tree is a flowchart-like tree structure where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test and each leaf node (terminal node) holds a class label. The topmost node in a tree is the root node. The example of decision tree is shown in figure:

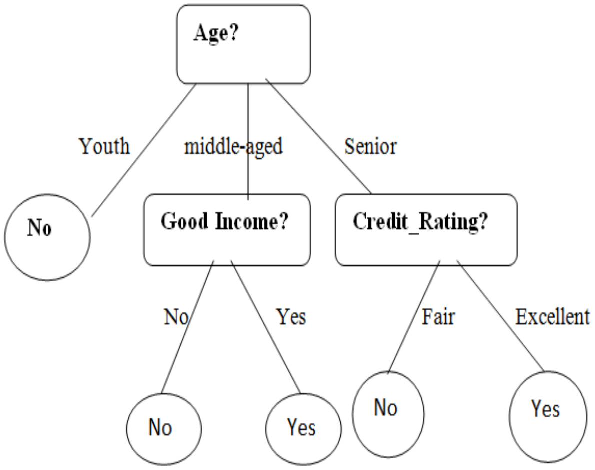

Figure: 1 Decision Tree for purchase_house

An example (shown in the figure: 1) represents the decision tree for purchasing the House (Own house), in which each internal node represents a test on an attribute. Each leaf node represents a class (i.e. either purchase_house = yes or purchase_house = no).

A decision tree [11] is a decision-making device which assigns a probability to each of the possible choices based on the context of the decision: P (f / h), where f is an element of the future attributes (the set of choices) and h is a history (the context of the decision). This probability P(f / h) is determined by asking a sequence of questions ql q2 ... qn about the context, where the i^{th} question asked is uniquely determined by the answers to the i - 1 previous questions. Each question asked by the decision tree is represented by a tree node and the possible answers to this question are associated with branches emanating from the node. Each node defines a probability distribution on the space of possible decisions. A node at which the decision tree stops asking questions is a leaf node. The leaf nodes represent the unique states in the decision-making problem, i.e. all contexts which lead to the same leaf node have the same probability distribution for the decision.

The power of decision-tree model is not in their expressiveness but instead in how they can be automatically acquired for very large modeling problems. The decision-tree learning algorithm increases the size of a model only as the training data allows. The leaf distributions in decision trees are empirical estimates, i.e. relative-frequency counts from the training data. Unfortunately, they assign probability zero to events which can possibly occur. Therefore, it is necessary to smooth empirical decision-tree models. For that we have considered C4.5 Decision Tree System, which is used for classification from a set of Trained Data.

C4.5 builds decision trees from a set of training data in the same way as ID3 using the concept of information entropy. The training data is a set $S = s_1,s_2,...$ of already classified samples. Each sample $s_i = x_1,x_2,...$ is a vector where $x_1,x_2, ...$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1,c_2,...$ where $c_1,c_2, ...$ represent the class to which each sample belongs. At each node of the tree, it chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. This algorithm then recurs on the smaller sub lists.

IV. Proposed System

In this paper, we have proposed a new system DEVDT for detection and extraction of video events by using Decision Tree classifiers (shown in fig.2). The aim of this system is to detect the relevant and pure events and extracted portion of video events effectively and efficiently.

The training data for data mining is the multimodal features (visual and audio) extracted for each video event. It is event-based because video events are the basic indexing unit for video content analysis [12, 13, 14]. In addition, we have adopted an advanced video event detection method, having the advantage of producing some important visual features and mid-level features (e.g., object information) during event detections. However, due to the small percentage (e.g., 1%) of the positive samples with the huge amount of negative samples domain knowledge utilizing visual and audio clues has been used in our data pre-filtering step to clean the original feature data set in order to provide a reasonable input training data set for the data mining component. To our best knowledge, there is hardly any work addressing this issue. Finally, the decision tree model generated by the data mining process will be tested and the overall performance is evaluated by using large amounts of long video sequences with different styles and produced by different broadcasters. By using our DEVDT system, we will reach up to 92% for both Recall and Precision.

The architecture of our system is shown in Figure 2. As can be seen from this figure, the proposed framework consists of the following three major components: Video Processing, Data Pre-Processing and Data Mining.

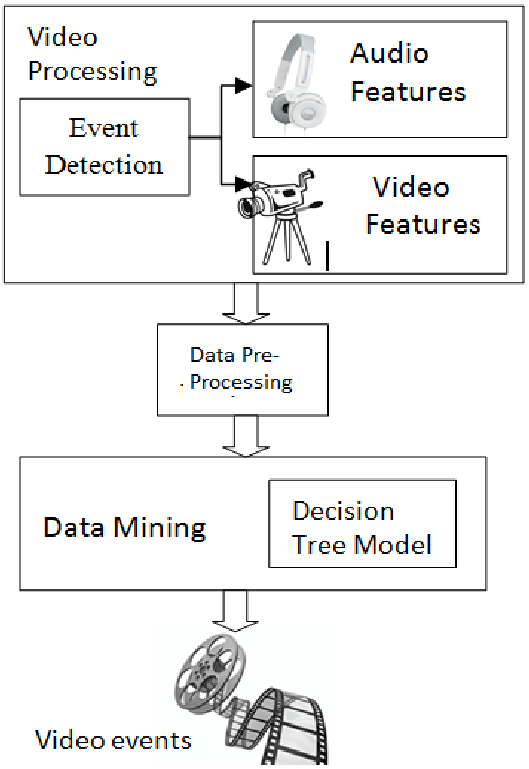

Figure: 2 DEVDT (Detection & Extraction of Video Events using Decision Tree)

A. *Video Processing*

Parse the raw video sequences by using a video event detection subcomponent. It not only detects video event boundaries but also produces some important visual features during event detection. The detected event boundaries are passed to feature extraction, where the complete multimodal features (visual and audio) are extracted for each event.

B. *Data Pre Processing*

Use domain knowledge such as visual audio clues to eliminate the noise data and reduce the irrelevant data from the original feature set since the ratio of actual events over the non-related events is very small (e.g., 1 actual event out of 100 events). By data pre- Processing, the ratio of positive samples over negative samples can be increased to 1:20.

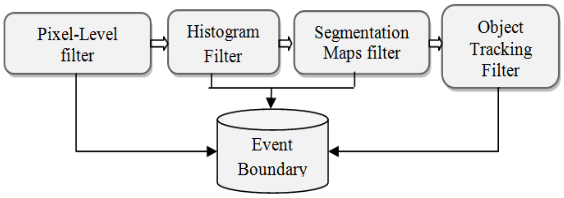

Figure 3: Data Pre-Processing for video event detection

C. *Video Event Detection*

The first step for video processing and the detected event boundaries is the basic unit for video feature extraction. In this

System, we have proposed a Data Preprocessing (multi-filtering architecture) including the pixel-level comparison, histogram comparison and segmentation map techniques (as shown in Figure 3). The first two filters can compensate for each other in reducing the numbers of both false positives and false negatives. In addition, since the object segmentation and tracking techniques are much less sensitive to luminance change and object motion. They are used as the last filter in this multi-filtering architecture to help determine the actual event boundaries. The advantages of this method are: Firstly, It has high precision and recall values. This overall performance is considered based on more than 1,500 testing events approximately and then secondly it can generate a set of important visual features for each event during the process of event detection. Thus the computation for extracting visual features can be greatly reduced.

D. Visual Event Extraction

In addition to event boundaries, the process of video event detection also generates a rich set of visual features associated with each video event. Among these visual features, pixel-change represents the average percent of the changed pixels between frames within an event which is output by the first filter (Pixel-Level Filter). The feature histogram change indicates the mean value of the histogram difference between frames within an event and is output by the second filter (Histogram Filter). Both of the two global features are important indications for camera motions and object motions. Other mid-level features such as the mean (back-mean) and the variance (back-var) values of the background pixels can be obtained via the segmentation filter.

E. Audio Feature Processing

Both time-domain and frequency-domain audio features are considered in our framework. Since the semantic meaning of an audio track is better represented by the audio features of a relatively longer period. We also explore both the clip-level and shot-level audio features. In this study, we define an audio clip with a fixed length of one second, which usually contains a continuous sequence of audio frames.

The generic audio features are divided into three groups: volume features (volume), energy features (energy), and Spectrum Flux features (sf). For each generic audio feature, the audio files are processed to obtain the audio features at both clip-level and shot level. The audio data is sampled at a sampling rate of 16,000 HZ. An audio frame contains 512 samples, which lasts 32ms under a sampling rate of 16,000 HZ. Within each clip, the neighboring frames overlap 128 samples with each other. In order to model the energy properties more accurately, four energy sub-bands are also used in this study. In this process we have used all IO audio features (i.e.1 volume feature, 5 energy features, and 4 spectrum flux features) to improve the performance.

F. Mining Video Events using Decision Trees

In our DEVDT system, the decision tree logic is adopted for mining events in videos. In this phase, we will take the 'cleaned' feature data as the training data and build a decision tree model suitable for video event detection. An interior node in a decision tree involves testing a particular attribute and the branches that fork from that node correspond to all possible outcomes of a test. Eventually, a leaf node is formed which carries a class label that indicates the majority class within the final partition. The classification phase works like traversing a path in the tree. Starting from the root, the instance's value of a certain attribute decides which branch to go at each internal node. Whenever a leaf node is reached its associated class label is assigned to the instance. The algorithm exploited in this study is adopted from the C4.5 decision tree [3].

In the decision tree generation process, the information gain ratio [15] criterion is used to determine the most appropriate attribute for partitioning due to its efficiency and simplicity. Numeric attributes are accommodated by a two-way split, which means one single breakpoint is located and serves as a threshold to separate the instances into two groups. The voting of the best breakpoint is based on the information gain value.

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. It is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification. In pseudo code, the general algorithm for building decision trees is [16]:

```
1) Check for base cases
2) For each attribute a
3) Find the normalized information gain from splitting on a
4) Let a_best be the attribute with the highest normalized
   information gain
5) Create a decision node that splits on a_best
6) Recurse on the sub lists obtained by splitting on a_best,
   and add those nodes as children of node
```

The advantages of using Decision Tree are it doesn't require any domain knowledge or parameter setting. Therefore it is appropriate for exploratory knowledge. C4.5 Decision trees can handle high dimensional data. It can be used for both continuous and discrete attributes. In order to handle continuous attributes it creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it. The learning and classification steps of this Decision Tree are simple and fast. Accurate results can be obtained with this Decision Tree method.

V. CONCLUSION AND FUTURE ENHANCEMENTS

This paper reviewed the importance of multimedia data mining and concludes that one of the major issues of multimedia data mining is the accuracy and efficiency of getting results for extraction of audio and video events from raw data. In this paper we have discussed about the multimedia mining modals and the decision tree logic concepts. We have proposed a framework DEVDT, Which uses data mining concept of Decision Tree classifier model for detection and extraction of video and audio events to improve efficiency and accuracy of extraction.

The construction of Decision Tree is performed by recursively partitioning the training set with respect to certain criteria until all the instances in a partition have the same class label or no more attributes can be used for further partitioning.

This can be done by using C4.5 Decision Tree system. The advantage of using this Decision tree system is to handle both continuous and discrete attributes. It can also handle training data with missing attributes. It allows attribute values to be marked as '?' for missing values. Missing attribute values are simply not used in gain and entropy calculations.

Our DEVDT system consists of three major phases - video processing, data pre-processing and data mining. In the first phase it detects the boundary of video events and also observes some important features of video events. In the second phase, it mainly cleans the data i.e. it eliminates the noise data and reduce the irrelevant data from the original feature and produces as training data. In the final phase, it mines the video events from training data and finally it produces pure and relevant events. In the data mining, the information gain ratio criterion is used to determine the most appropriate attribute for extraction due to its efficiency and simplicity for decision tree generation process.

In our future work, this framework will be tested and extended in various types of video events like movies, news, traffic videos (raw video events) and medical video events like ultra sound videos.

ACKNOWLEDGMENT

We would like to thank everyone, who has motivated and supported us for preparing this Manuscript.

REFERENCES

[1] Simeon J. Simoff, Chabane Djeraba and Osmar R. Zaïane,"MDM/KDD2002: Multimedia Data Mining between Promises and Problems", SIGKDD Explorations, 2002.

[2] Shroff. N, Turaga.P, and Chellappa.R, Video Prices: Highlighting Diverse Aspects of Videos, Multimedia, IEEE Transactions, and December 2010.

[3] J.R. Quinlan. **C4.5:** Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[4] X. Zhu, X. Wu, A. K. Elmagarmid, Z. Feng, and L. Wu, "Video data mining: Semantic indexing and event detection from the association perspective," IEEE Trans.2005.

[5] Dianhui Wang, Yong-Soo Kim, Seok Cheon Park, Chul Soo Lee and Yoon Kyung Han, "Learning Based Neural Similarity Metrics for Multimedia Data Mining" Soft Computing, February 2007.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 1994 International Conference on Very Large Databases, Santiago, Chile, September 1994.

[7] V. Gudivada, V.V. Raghavan, and K. Vanapipat, 'A Unified Approach to Data Modeling and Retrieval for a Class of Image Database Applications,' In Multimedia Database Systems, V.S. Subrahmanian and S. Jajodia (Eds.), Springer-Verlag, Berlin, Germany, 1996.

[8] Benoit Huet, Alan Smeaton, Ketan Mayer-Patel and Yannis Avrithis "Advances in Multimedia Modeling"Proc. of International Multimedia Modeling Conference, MMM 2009.

[9] W.I. Grosky, F. Fotouhi, and Z. Jiang, "Using Metadata for the Intelligent Browsing of Structured Media Objects," In Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data A. Sheth and W. Klas (Eds.), McGraw Hill Publishing Company, New York, 1998.

[10] W.I. Grosky, "Managing Multimedia Information in Database Systems", Communications of the ACM, Volume 40, Number 12 (December 1997).

[11] MichaelN; Large database decision tree classifiers; Decision Trees & Data Mining, March 2006.

[12] S.C. Chen, M.L. Shy, C. Zhang, L. Luo, and M. Chen, "Detection of Soccer Goal Events using Joint Multimedia Features and Classification Rules", Proc. of International Workshop on Multimedia Data Mining (MDWKDD '2003).

[13] Rosenfeld A., D. Doermann, D. DeMenthon, Eds., Video Mining, Kluwer, 2003.

[14] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala, "Soccer highlights detection and recognition using HMMs", proc. of IEEE International Conference on multimedia and Expo,2002.

[15] Deng,H.; Runger, G.; Tuv, E. "Bias of importance measures for multi-valued attributes and solutions". Proc. of the 21st International Conference on Artificial Neural Networks (ICANN), 2011.

[16] S.B. Kotsiantis; Supervised Machine Learning: A Review of Classification Techniques, Informatica 31(2007), July 2007.

[17] Shu-Ching Chen Mei-Ling Shyu Min Chen, Chengcui Zhang. A Decision Tree-based Multimodal Data Mining Framework for Soccer Goal Detection.

AUTHORS PROFILE

Dr P. Premchand is a professor in department of Computer Science & Engineering, Osmania University, Hyderabad, A.P and India. He completed his ME (Computer Science) from Andhra University, A.P. He has received Ph.D degree from Andhra University, A.P. He guided many scholars towards the award of Ph.D degree from various Universities. He was a Director of AICTE, New Delhi, during 1998-99. He also worked as Head of the Dept of CSE and Additional Controller of Examinations, Osmania University, AP. Now currently he is a chairman of BOS, Faculty of Engineering, and O.U. from 2007 to tilldate.

Prof Shaik.Abdul Nabi is the Head of the Dept. of Computer Science & Engineering, AVN Inst.Of Engg.Tech, Hyderabad, AP, and India. He completed his B.E (Computer Science) from Osmania University, A.P. He received his M.Tech. From JNTU College of Engg.,Hyderabad and currently pursuing Ph.D. in the area of Data Mining from Acharya Nagarjuna University, Guntur, AP and India. He is a certified professional by Microsoft. His expertise areas are Data warehousing and Data Mining, Data Structures & UNIX Networking Programming.

Mr.Shaik Rasool is working as a Asst.Prof in Dept of Computer Science &Engineering in S.C.E.T, Hyderabad, India. He received the Bachelor of Technology in Computer Science & Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in 2008. He is completed Master of Technology in Computer Science & Engineering from S.C.E.T., Hyderabad, India. His main research interest includes Data mining, Network Security, Information Security, cloud computing.