Nonparametric Video Retrieval and Frame Classification using Tiny Videos

A.K. M. Shanawas Fathima,
PG Student,
Department of CSE
GCE, Tirunelveli.

R. Kanthavel,
Department of CSE,
Government College of Engineering,
Tirunelveli.

ABSTRACT

A nonparametric video retrieval and frame classification systm that uses affinity propagation algorithm is proposed. The main goal of the proposed system is to develop “tiny video” that achieves high video compression rates while retaining the overall visual appearance of video. The proposed video retrieval system utilizes the strengths of affinity propagation algorithm that uses exemplar based clustering to achieve a trade off between compression and video recall. By using this large collection of user labelled videos in conjunction with simple data mining techniques to perform related video retrieval, as well as classification of images and video frames. The main applications of this proposed system is video copy detection and video recognotion.

Index Terms

Image classification, content-based retrieval, tiny videos, tiny images, data mining, nearest-neighbor methods**.**

1. INTRODUCTION

Today, video has become a prominent component of many news, entertainment, information, blogging, and personal Web sites. Today people have access to a tremendous amount of video information in the internet. With a large video data collection, it is infeasible for a human to classify or cluster the video scenes or to find either the appropriate video scene, or the desired portions of the video.

Video retrieval is an essential technology to design video search engines with serve as an information filter and shifts out an initial set of relevant videos from database. A large number of approaches have been attempted for forming automatic content-based retrieval of video. The approaches are

- Videosegmentation-based features.
- Motion-based features.

1.1 Video segmentation:

Video segmentation is a fundamental step in analyzing video sequence content and in devising methods for efficient access, retrieval and browsing of large video databases. There are two typical video segmentation methods

- Shot-based.
- Object-based.

To date many works focus on breaking video into shots, and then searching the video for appropriate shots. Object-based methods segments video into objects and use some of suitable object features for hierarchical indexing of video contents. Motion based approaches use the trajectories of objects to indexing and browsing video contents.

In shot based video retrieval, finding the interested video scene requires an efficient video shot boundary detection algorithm. Shot is a sequence of frames captured by one camera in a single continuous action in time and space. Shot boundary detection is an important task in managing video database for indexing, browsing and other content-based operations. Video shot boundaries need to be determined possibly automatically to allow content based video retrieval manipulation.

Video has both spatial and temporal dimensions and hence a good video retrieval should capture the spatio-temporal contents of the scene. Therefore, when shot boundaries are detected, it is need to extract some spatio-temporal features from shots, which could be utilized to compare visual content of video shots for finding desired video scenes.

The video retrieval system is evaluated using two common measures, recall and precision, which are defined as follows:

- The Recall measure, also known as the true positive function or sensitivity, which corresponds to the ratio of correct experimental detections over the number of all true detections. It measures the ability of a system to present all relevant items.
- The Precision measure defined as the ratio of correct experimental detections over the number of all experimental detections. It measures the ability of a system to present only relevant items.

Despite readily available online video content, the majority of video retrieval and recognition research to this day employs much smaller data sets. Frequently used annotated research databases such as the Open Video Project [2] contain on the order of 5,000 videos. TREC Video Retrieval Evaluation (TRECVID) [8] uses about 200 hours of video footage. These small data sets, while convenient, do not capture the diversity of online video content viewed daily by the public. Large data sets of user-generated and annotated data are inherently more noisy and challenging.

In addition, efficient storage space utilization and computational complexity also play a major role. However, if some of these challenges are overcome, then the ubiquity and diversity of online visual data can be leveraged to aid a variety of computer vision problems. This is the case because a very large amount of data and simple algorithms can be used in place of sophisticated algorithms to model the complexity of the vision task at hand.

A number of recent research papers have used large collections of images for various computer vision tasks [6], [8], [10]. The tiny images database [10] is currently the largest labeled database of images. It consists of 79,302,017 images that were collected from the Internet and down-sampled to tiny 32 * 32 pixel size. In [10], Torralba et al. use this large data set of images and very simple nearest neighbour (NN) techniques to perform person detection and localization, scene recognition, automatic image annotation, as well as image colorization and orientation detection.

In this paper, a new method is proposed for using videos to classify objects, scenes, people, and activities. A database of 52,159 videos collected from YouTube and compressed to tiny size can be used to classify a wide range of categories using very simple nearest neighbour techniques.

Furthermore, the representation of tiny videos is compatible with the tiny image representation. This allows not only comparing, but also combining both data sets for a variety of classification tasks. Tiny videos perform better than tiny images for classification tasks involving sports activities and scenery. And also by combining both data sets, classification performance is improved for a wider range of categories and visual appearances.

This paper is organized into five parts. In part 2, existing approaches to video retrieval is explained. In part 3, proposed system and its modules are discussed. In part 4, the proposed algorithm is discussed. In part 5, experimentation results are presented. In part 6, we conclude with a discussion of future work.

2. EXISTING APPROACHES:

A large number of video summarization algorithms have been developed to perform temporal compression [1], [5], [7], [9]. Many of these algorithms are quite complex since they often depend on shot boundary detections, which are difficult to detect reliably due to gradual shot transitions such as blends and wipes [3], [11], [12]. In addition, false positives can arise from fast moving objects in front of the camera lens or fast motions of the camera itself (e.g., camera pans and dollies). Furthermore, frames coming from the same shot can appear more distinct than frames coming from different shots in the presence of camera motion.

2.1. Uniform sampling:

The most widely employed summarization approach is uniform sampling. In uniform sampling, frames are extracted at a constant interval. The main advantage of this approach is computational efficiency.

2.1.1. Limitations:

Uniform sampling tends to oversample long shots or skip short shots.

2.2 Intensity of motion sampling:

A widely used approach that adapts to changes in frame content is intensity of motion (IM) frame sampling. Intensity of motion has also been used as a feature vector for describing motion characteristics and also for detecting sharp boundary transitions in prior work [1]. Intensity of motion is defined as the mean of consecutive frame differences:

$$A\ (t) = 1/XY \sum_{x\ ,y} |L(x,y,t+1)\text{-}L(x,\ y\ ,t)|...(1)$$

Where X and Y are the dimensions of the video (X = Y = 32 in this case) and L(x, y, t) denotes the luminance value of pixel (x, y) of a frame at time t.

IM sampling allows the sampling rate to be controlled by adjusting the standard deviation of the Gaussian filter. Larger standard deviations lead to fewer extrema and, as a result, fewer key frames.

The intensity of motion key frame selection algorithm is robust to color and affine transformations. For a given shot, it also selects the same key frames regardless of any temporal shifts or the appearance of neighboring shots. These properties make it particularly suitable for content based copy detection (CBCD) and other video retrieval tasks as it generally samples the same set of frames for a shot occurring across multiple videos.

3. PROPOSED SYSTEM

In this proposed system, tiny videos dataset is used for video retrieval and frame classification tasks. And also affinity propagation algorithm is used here to solve the above problem by considering the similarities across multiple shots.
The proposed video retrieval and frame classification system composed of three modules.

1. Compressing temporal dimension of video using Affinity Propagation.
2. Video retrieval using tiny video dataset.
3. Video classification.

3.1. Compressing temporal dimension of video using affinity propagation:

In this module, first videos are collected from YouTube. After the video collection procedure, pre-processing the video and the tiny video representation is developed and then compressing the temporal dimension of video using affinity propagation.

3.1.1. Video collection procedure:

The videos were primarily collected in YouTube's News, Sports, People, Travel, and Technology sections. For each of these categories, YouTube finds about 350,000 results. However, the API allows to retrieve only the top 1,000 results. Therefore, to increase the number of videos collected per section, the API is used to sort the results by most viewed, top rated, relevance, and most recent. It is important to note that collecting videos in this non random way introduces slight biases in the sample of YouTube videos. For example, frequently viewed videos tend to be shorter in duration than rarely viewed videos. Hence, videos sorted by most viewed will increase the fraction of short videos in the data set.
For each video, all of the associated metadata returned by YouTube's API is also stored. The metadata includes such information as video duration, rating, view count, title, description, and user assigned labels (tags).

This module includes the following steps.

- Video Pre-processing
- Frame Resizing
- LUV Conversion
- Affinity Propagation

3.1.2. Video pre-processing procedure:

In this step, YouTube videos are stored in their native Flash video format. The frame rate of Flash videos is not constant and usually depends on the content (e.g., a video of slides from a presentation might only have one frame per slide spanning several seconds or even minutes). To ignore this complication, extracting frames at constant frame intervals rather than constant time intervals because of extracting unique-looking frames. It crop videos that were encoded with black bars. To detect horizontal black bars, the following formula is used.

$$F(y) = 1/M \sum_{x, c} |I_y(x, y, c) | \quad .... (2)$$

$$Y_{min}= \min [\arg_{y} \min (f(y) > t)] \quad .... (3)$$

$$Y_{max} = \max [\arg_{y} \max (f(y) > t)] \quad .... (4)$$

Where I_y is the derivative along y of frame I, which is an N × M × 3 matrix (where N and M correspond to the native resolution of the video).Then sum the gradient responses along the x-direction and the color channels c, yielding f(y). Large f(y) values are likely to correspond to a transition between the frame content and the black bar. Then pick the smallest y and largest y, for which f(y) is greater than a threshold t (a value of 0.5 was found to work well), as the locations at which the frame will be cropped (i.e., y_{min} and y_{max}, respectively). The bar being removed contains at least 80 percent black pixels is also checked. Otherwise, the region is not cropped. This technique is repeated for vertical black bars.

This technique removes frames that contain more than 80 percent of pixels of the same color. These frames generally correspond to title sequences and diagrams.

3.1.3. Frame resizing:

Since the tiny videos data set to be compatible with tiny images, the individual frames are resized to be 32 × 32 pixels in size. And then concatenate the three color channels and normalize the resulting frame vector to have zero mean and unit norm. This is done in order to reduce sensitivity to variations in illumination and is a common transformation in image processing. The resulting normalized tiny frame is compatible with the tiny images descriptor.

Unlike images, videos have an additional temporal dimension. The temporal dimension of most videos is densely sampled (usually at a rate of 24 frames per second) even when the motion in the shot is minuscule. As a result, videos can be strongly compressed temporally by retaining only distinct visual appearances.

3.1.4. LUV Conversion:

After frame resizing, the video is converted into luminance and chrominance format. Since luminance and chrominance values having better clarity than RGB values.

3.1.5. Affinity propagation:

In this paper, a video-summarization algorithm [4] that uses exemplar-based clustering to select only unique looking key frames is proposed. Similar to uniform and IM sampling, this approach does not rely on shot boundary detection. However, exemplar-based clustering not only captures within-shot visual appearance variations, but also consolidates similarities across multiple shots. This is particularly suitable for YouTube as its video clips are generally short and shots often alternate between a small set of scenes .This allows the visual range of most clips on YouTube to be captured with only a few unique-looking frames.

Affinity propagation [4] is used to cluster densely sampled frames into visually related groups. Only the exemplar (or "unique looking") frame within each cluster is retained and the rest are discarded.

Affinity propagation (AP) is particularly suitable here because it allows us to define what "unique looking" means in terms of the same frame similarity metrics that will be used later for video retrieval. As a result, AP selects exemplars such that similarity with their cluster members is maximized. By adjusting the preference parameter p, the number of exemplars (or keyframes) that AP sampling selects can be controlled.

3.1.6. Frame and video similarity metrics:

In [10], Torralba et al. define a basic distance measure between two tiny images I_a and I_b (tiny frames in this case) as their sum of squared differences:

$$D^2_{ssd}(I_a, I_b) = \sum_{x,y,c}(I_a(x,y,c) – I_b(x,y,c))^2 \quad ... (5)$$

Where I denotes a 32 × 32 × 3 dimensional zero mean, normalized tiny video frame, or tiny image. Furthermore, Torralba et al. show that recognition performance can be improved by allowing the pixels of the tiny image to shift slightly within a 5-pixel window. This reduces sensitivity to slight image misalignments, such as moving objects or variations in scale. Hence, the following distance metric is also used:

$$D^2_{shift}(I_a, I_b) = \sum_{x,y,c} \min_{| Dx,y | <= w} (I_a(x,y,c) – I_b^{\wedge}(x+ D_{x,}y + D_{y,}c))^2 \quad .... (6)$$

where w is the window size within which individual pixels can shift and $I^{\wedge}$ is a transformed version of frame I. Extend these frame distance measures to work for a pair of videos V_α and V_β by defining the following basic video distance measure:

$$D\wedge^2_{ssd/shift}(V_\alpha ,V_\beta) = \min_{Ia \text{ € } V\alpha,\ Ib \text{ € } V\beta}(D^2_{ssd/}shift(I_a , I_b)) \quad ..... (7)$$

In essence, the distance between two videos V_α and V_β is defined as the distance of the most similar pair of frames I_a and I_b belonging to these videos. Note that, if both videos V_α and V_β consist of a single frame, then the $D^{\wedge 2}$ ssd/shift distance metric reduces to D^2 ssd/shift. Furthermore, a single tiny image or tiny frame can be substituted for V_α in order to compute the distance between that image or frame and the tiny video V_β. The correlation is defined in terms of distance as follows:

$$\rho=1- ½ D^2_{ssd} \quad ..... (8)$$

$$\rho^{\wedge=} 1- ½ D\wedge^2_{ssd} \quad ..... (9)$$

This relationship can be trivially derived by expanding the expression for D^2_{ssd} , collecting the terms that sum to 1, and rearranging it. A correlation of ρ=1 for two frames implies that the frames are identical. For a pair of videos, $\rho^{\wedge=}$ 1signifies that the videos share at least one identical frame. A correlation of zero corresponds to completely dissimilar frames (i.e., the descriptors are orthogonal).

These distance measures allows finding a set of NN given an image or frame as input. In addition, Torralba et al. propose using PCA-compressed descriptors to facilitate faster neighbour retrieval. Finally, two additional similarity metrics that are particularly suitable for duplicate video retrieval defined.

$$S_1 (\alpha, \beta) = \sum_{Ia€Va} X (\rho^{\wedge} (I_\alpha ,V_\beta)) \quad ..... (10)$$

$S_2 (\alpha,\beta) = \sum_{Ia€Va} x(\rho^{\wedge}(I_\alpha ,V_\beta))\ .\ X (\rho\ (I_{a+1},I_{b+1})\ldots$ (11)

Here, S_1 counts the number of frames I_a in video V_α that match to video V_β with a correlation ρ^ that exceeds τ. S_2 adds an additional constraint, by only counting consecutively matching pairs of frames in V_α and V_β since such occurrences are less likely to arise by chance.

4. AFFINITY PROPAGATION ALGORITHM:

INPUT: a set of pair wise similarities, {s(i, k)}(i,k)€{1,...,N}2,i!=k where s(i, k)€R indicates how well-suited data point k is as an exemplar for data point i.

e.g. $s(i, k) = -\|xi - xk\|^2$, i !=k (squared Euclidean distance)

For each data point k, a real number s (k, k) indicating the a priori preference (negative cost of adding a cluster) that it be chosen as an exemplar.

e.g. s(k, k) = p ¥ k € {1, . . . ,N} (global preference)

INITIALIZATION: set availabilities to zero, ¥i, k: a (i , k) = 0.

REPEAT: responsibility and availability updates until convergence

¥i, k : r(i, k) = s(i, k) − max k′:k′!=k[s(i, k′) + a(i, k′)] (12)

$\sum_{i':i'!=i}$ max[0, r(i′, k)], for k=i

¥i, k : a(i, k) = min[0, r(k, k)+ $\sum_{i':i'!=\{i,k\}}$ max[0, r(i′, k)], for k!=i (13)

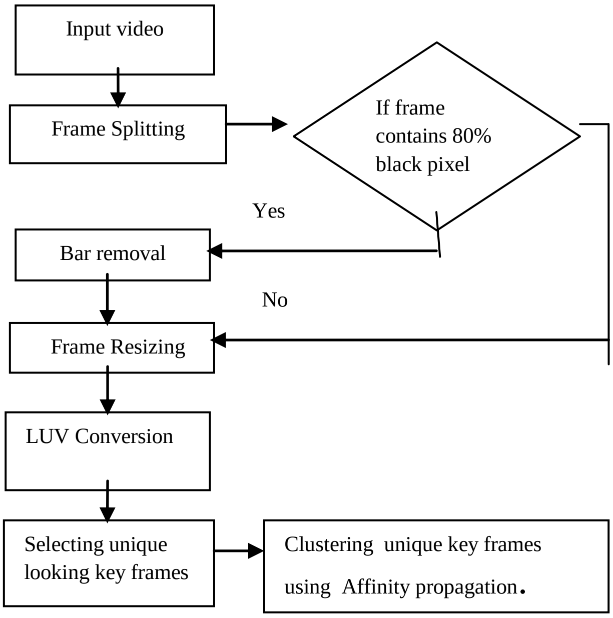
Input video
Frame Splitting
If frame contains 80% black pixel
Yes
Bar removal
No
Frame Resizing
LUV Conversion
Selecting unique looking key frames
Clustering unique key frames using Affinity propagation.

Figure: 1 Schematic representation of compressing temporal dimension of video.

The figure: 1 shows that the input video is pre-processed and if the video contains 80% of black pixels then perform bar removal. Otherwise frame resizing is performed. After frame resizing and LUV conversion, affinity propagation algorithm is used to select unique looking key frames from the input video. Finally, those unoique looking key frames are clustered.

5. EXPERIMENTAL RESULTS:

Any no. of input video can be examined and unique looking key frames from the input video are selected using exemplar based clustering.

Figure: 2 input video

Figure: 2 shows that the query video and this video is pre-processed and then that video is resized into 32 * 32 size. After frame resizing, the video is converted into LUV format. Since luminance and chrominance values having better clarity than RGB values.

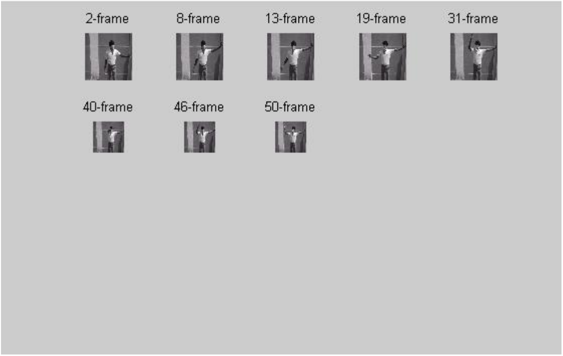

Figure: 3 unique looking key frames by using affinity propagation.

Figure: 3 shows that the unique looking key frames of the input video by using affinity propagation algorithm. This affinity propagation uses exemplar based clustering to select unique looking key frames.

6. FUTURE WORK AND CONCLUSION:

A method for compressing a large database of videos into a compact representation called "tiny videos" is presented. These tiny videos can be used effectively for content-based copy detection. In addition, by using this large data set of user-labeled online videos to perform a variety of classification tasks using only simple nearest-neighbour methods. After the development of tiny video representation, affinity propagation that utilizes exemplar based clustering is used to select only unique looking key frames in a fast and reliable fashion.In future work, related video retrieval and frame classification using tiny video dataset has to be done.

7. REFERENCES

[1] N. Dimitrova, T. McGee, and H. Elenbaas, “Video Keyframe Extraction and Filtering: A Keyframe Is Not a Keyframe to Everyone,” Proc. Sixth Int’l Conf. Information and Knowledge Management, pp. 113-120, 1997.

[2] G. Geisler and G. Marchionini, “The Open Video Project: A Research-Oriented Digital Video Repository,” Proc. ACM Digital Libraries, pp. 258-259,, 2000.

[3] A. Hampapur, R. Jain, and T.E. Weymouth, “Production Model Based Digital Video Segmentation,” Multimedia Tools Appl., vol. 1, no. 1, pp. 9-46, 1995.

[4] A. Karpenko and P. Aarabi, “Tiny Videos: A Large Data Set for Nonparametric Video Retrievaland Frame Classification,” IEEE transactions on Pattern analysis and machine intelligence, vol. 33, No. 3, March 2011.

[5] S. Lu, M.R. Lyu, and I. King, “Semantic Video Summarization Using Mutual Reinforcement Principle and Shot Arrangement Patterns,” Proc. 11th IEEE CS Int’l Multimedia Modelling Conf., pp. 60-67, 2005.

[6] D. Niste´r and H. Stewe´nius, “Scalable Recognition with a Vocabulary Tree,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2161-2168, 2006.

[7] B. Shahraray, “Scene Change Detection and Content-Based Sampling of Video Sequences,” Proc. SPIE Conf., pp. 2-13, Apr. 1995.

[8] N. Snavely, S.M. Seitz, and R. Szeliski, “Modeling the World from Internet Photo Collections,” Int’l J. Computer Vision, vol. 80, no. 2, pp. 189-210, Nov. 2008.

[9] C. Toklu, S.P. Liou, and M. Das, “Videoabstract: A Hybrid Approach to Generate Semantically Meaningful Video Summaries,” Proc. IEEE Int’l Conf. Multimedia and Expo, vol. 3, pp. 1333- 1336, 2000.

[10] A. Torralba, R. Fergus, and W.T. Freeman, “80 Million Tiny Images: A Large Data Set for Non-Parametric Object and Scene Recognition,” Technical Report MIT-CSAIL-TR-2007-024, 2007.

[11] R. Zabih, J. Miller, and K. Mai, “A Feature-Based Algorithm for Detecting and Classifying Scene Breaks,” Proc. ACM Multimedia Conf., pp. 189-200, 1995.

[12] R. Zabih, J. Miller, and K. Mai, “A Feature-Based Algorithm for Detecting and Classifying Production Effects,” Multimedia Systems, vol. 7, no. 2, pp. 119-128, 1999.