Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 15 (2): 256-263, 2015
ISSN 1812-5654

ansinet
Asian Network for Scientific Information

RESEARCH ARTICLE OPEN ACCESS

DOI: 10.3923/jas.2015.256.263

Automatic Video Annotation Framework Using Concept Detectors

Fereshteh Falah Chamasemani, Lilly Suriani Affendey, Norwati Mustapha and Fatimah Khalid
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, 43400, Selangor, Malaysia

ARTICLE INFO
Article History:
Received: August 21, 2014
Accepted: November 06, 2014

Corresponding Author:
Fereshteh Falah Chamasemani,
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia,
Serdang, 43400, Selangor, Malaysia
Tel: +603-89471709
Fax: +603-89466577

ABSTRACT
Automatic video annotation has received a great deal of attention from researchers working on video retrieval. This study presents a novel automatic video annotation framework to enhance the annotation accuracy and reduce the processing time in large-scale video data by utilizing semantic concepts. The proposed framework consists of three main modules i.e., pre-processing, video analysis and annotation module. The framework support an efficient search and retrieval for any video content analysis and video archive applications. The experimental results on widely used TRECVID dataset using concepts of Columbia374 demonstrate the effectiveness of the proposed framework in assigning appropriate and semantically representative annotations for any new video.

Key words: Content-based video retrieval, video annotation, semantic video annotation, image retrieval, video concept detection

INTRODUCTION

The importance of automatic video annotation has increased with emerging huge mass of digital video data which is producing by video capturing devices and storing in the Internet and storage devices. Indexing, mining and retrieving relevant videos using textual queries is not trivial tasks since many videos have none or irrelevant annotations. Therefore, automatic video annotation has been proposed to act as a mediator in the applications of Content-Based Video Retrieval (CBVR) and video management. It solves the aforementioned problem and bridge the semantic gap between high-level user's need (human perception) and low-level feature description (Bagdanov *et al*., 2007; Datta *et al*., 2008).

Video annotation aims to assign a set of suitable predefined concepts to video clips based on their semantic and visual content (Li and Wang, 2008; Wang *et al*., 2009). Video annotation also known as "Video semantic annotation" can be performed in three of the following techniques: Traditional manual annotation, rule-based annotation and machine learning technique (Naphade and Smith, 2004; Zhang, 2003). Manual annotation is laborious, time consuming, ambiguous, too subjective and error prone process. Furthermore, this technique has low efficiency and speed (Tang *et al*., 2007; Settles *et al*., 2008). Rule-based annotation technique classifies the annotation using expert knowledge. Since, commonly the undeveloped hierarchical annotation forms are used then it neither cover all the semantic content of video nor the versatile requirement of video annotations (Dorado *et al*., 2004). The third group, machine learning technique, can act as a supervised classification task to use in automatic video annotation to cope with the weaknesses of the other techniques (Tao *et al*., 2009).

This study is based on the machine learning technique to annotate video. Hence, the key idea of Automatic Video Annotation (AVA) is to construct the model(s) through automatic learning of semantic concept from many videos (even shots or keyframes), then to utilize these concept model(s) for predicting appropriate annotation/label for any new video. Later, these annotated videos can be retrieved by textual queries. However, the performance of AVA highly depends on: First, the video content representation; second, feature extraction; third, feature selection which means to choose more visual features for training classifiers result in: (1) Having various visual characteristic of video, (2) Improve the classifiers capability to recognize different video concepts and (3) Enhance classification accuracy. Fourth, employing an effective classification algorithm and proper dataset since any inappropriate use of dataset for constructing model during training stage will lead to deterioration in the performance of AVA due to the lack of adequ-ate concepts in their annotation vocabularies (Li *et al*., 2009).

Object-Based Video Annotation system was developed by Li *et al.* (2009) to perform the video annotation process in Internet by categorizing video in three types and annotating them in different ways. Their three video categories include first, hot videos which were downloaded many times; second, videos with lots of related information like, title of video, date, producer(s), actor(s) and so on; third, not hot videos with short of related information which are the rest of videos. They used e-Annotation architecture for manually annotating the first type of video while they automatically annotated the second video type using web mining methods. In the third category, they used video analysis model for detecting the video object and predicting their label simultaneously. Although their model revealed satisfactory annotation result on the real video gathered from the Internet, they did not evaluate their system with any large benchmark dataset.

Pan *et al.* (2004) proposed a correlation-based and uniqueness weighting scheme (with multiple alternative designs namely, Corr, Cos, SvdCorr and SvdCos) to find correlation between extracted image features and existing concepts using annotated images from only 10 Corel image datasets. Ding and Qin (2010) used 20 concepts over TRECVID video dataset and Corel image dataset to annotate new images or videos by reconstructing value from the sparse vector (utilizing a matching pursuit method after creating a dictionary matrix using training sets) and positive samples. Actually they adapted a common semantic classification problem into a compressed sensing theory (Ding and Qin, 2010).

Qiu *et al.* (2010) have presented an approach for annotating news video by extracting semantic context from their associated subtitles. This semantic context was formed by recognizing a set of significant terms that exist in subtitle of the given video. In the next step, they used these semantic concepts to refine annotation by measuring the semantic similarity (using Google distance and WordNet distance) among the context terms and candidate concepts. They conducted their experiments to annotate only 39 LSCOM concepts with TRECVID 2005 dataset (Qiu *et al.*, 2010).

The current methods on AVA commonly cannot deal with large-scale video dataset as well as various concepts (because of the inherent complexities that exist in video data and their semantic concepts) in term of computational cost and annotation accuracy. To cope with the above problems, this research develops a novel framework called Automatic Video Annotation using Concepts Detectors (AVAuCD) to automatically annotate a new input video with well trained classifiers using 374 concepts of Columbia374 (due to its broadness and public availability) as its base concept detectors in each extracted keyframes (Yanagawa *et al.*, 2007). The performance of the present proposed framework was assessed using standard video dataset TRECVID and Clipcanvas's video. In this study video clips are automatically annotated using 374 concepts therefore two words "concept" and "annotation" are used in the same manner.

MATERIALS AND METHODS

Overview of proposed framework: The AVAuCD framework is designed as an effective, efficient and convenient means for automatic video annotation based on large-scale video dataset as well as various concepts. Figure 1 illustrates the architecture of the proposed framework of AVAuCD which includes three main modules: (a) Pre-processing, (b) Video analysis and (c) Annotation. The main task of the pre-processing module is to provide feature vectors by extracting significant features from the dataset's keyframes. Upon receiving new video, the video analysis module shall extract features from the extracted keyframes using similar features used in the pre-processing module.

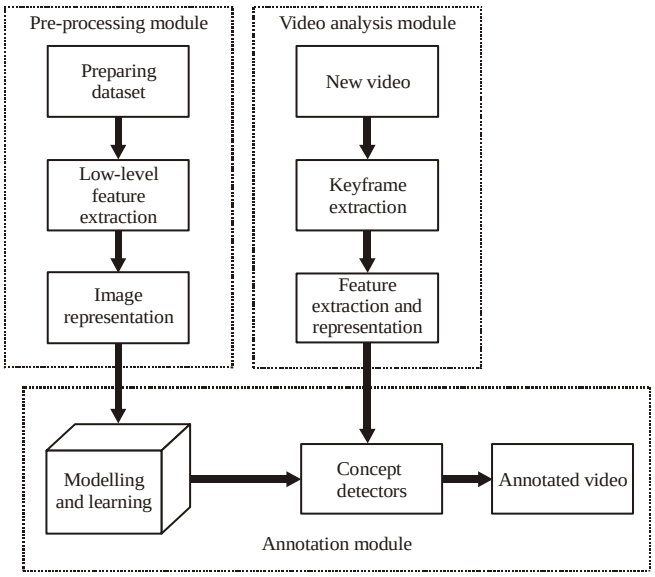

Fig. 1: AVAuCD framework

Then these extracted features are passed into the annotation module in order to assign the annotations/concepts to them. Hence, in the first step of annotation module, the features that provided in the pre-processing module are used as training samples to train concept detecors for each concept of Columbia374. Once the models are constructed then the annotation for the keyframe's features of video analysis module is predicted using this model.

Implementation of AVAuCD framework

Pre-processing module: The two main objectives of the pre-processing module are extracting the important features from the training sample and constructing associated feature vectors. The SIFT and GIST features are used to get the local details and holistic descriptions of each keyframes correspondingly. To make these raw features usable for further processes some encoding and pooling need to be performed.

Preparing dataset: This study used TRECVID 2005 dataset and Columbia374 to train its concept detectors.

Low-level feature extraction: Feature extraction is a process to obtain a set of features which reveal a compact representation of keyframes that cover their visual property. These features include local and global representation of keyframes. While the global features cover the overall attribute (such as color, shape, texture) of the keyframe, the local features contain the visual property like pixel's intensity and color. The best appropriate feature sets in this framework are Scale Invariant Feature Transform (SIFT) to extract local shape features and GIST to present spatial envelope of a scene from a given keyframe.

This study used SIFT features to find the keypoints or local characteristic of the keyframes, since it has successfully been used in many retrieval and recognition tasks. The SIFT developed by Lowe (2004) is considered as the fastest and most popular method. That is invariant to rescaling, rotation, translation, illumination changes and affine transformations.

The SIFT detects keypoints (interest points or salient keyframe regions) using Difference of Gaussians (DOG) at different location and scales, then the SIFT descriptors (compact description of the keyframes appearance) are computed based on affine covariant region surrounding the keypoints. In other word, every SIFT descriptor is composed of edge direction histogram of the keyframe at different locations and it is computed by rotating the region of keyframe in accordance with the main region intensity direction. Then, this region is split in some equal sub regions followed by computing the orientation histograms of all these sub regions. The K-means clustering techniques is applied to all these SIFT descriptors to quantize them and provide keyframe patch.

Figure 2 illustrates different steps of constructing SIFT features. The first processing step of SIFT algorithm is to detect candidate keypoitns using DOG on keyframe to check the scale space exterma by comparing each point with its 8 neighbours (pixels) in the same scale as well as its 9 neighbours (pixels) in one scale upper and lower.

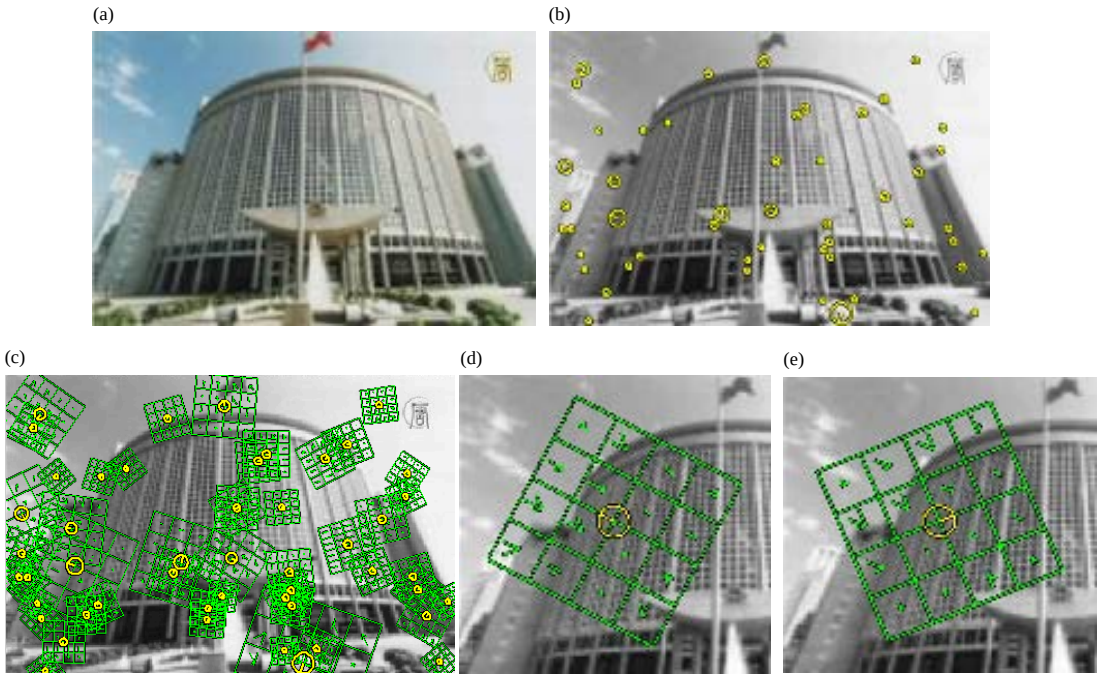

Fig. 2(a-e): (a) Input keyframe, (b) Detected SIFT features, (c) Detected SIFT features and their descriptors and (d, e) Custom keypoints with different orientations

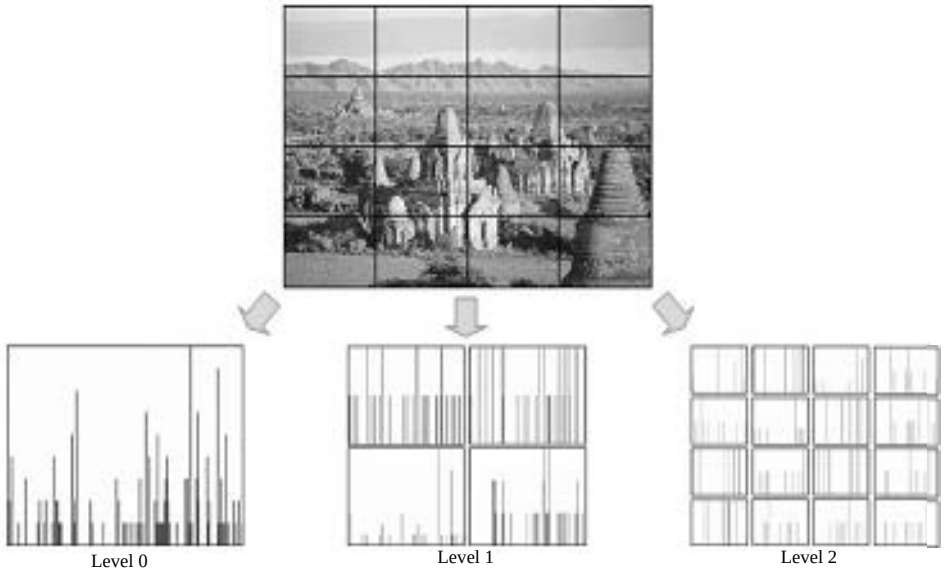

Fig. 3: Schematic illustration of SPM (Dickinson *et al*., 2009)

In the second step some insignificant interest points or keypoints are eliminated for example those which lies along the edges.

Then, to make keyframe invariance to rotation, an orientation needs to be assigned to every keypoint. Considering these orientation and scale a rectangular of 4×4 grid centered at the keypoint estimate dominant orientation by calculating the average weighted of the gradient magnitude and direction at this grid. The local histograms computed for 8 quantized directions (bins) at all the 4×4 grid neghibour pixels which lead to a keyframe descriptor in 128 dimensions (4×4×8) for every keypoints. The result is a feature vector including 128 elements known as the SIFT descriptor.

The GIST descriptor is extracted based on Oliva and Torralba method due to its low dimensionality description of the scene to quantify high level semantic attribute and its efficiency in scene classification (Oliva and Torralba, 2001). The main idea of GIST descriptor is to find global representation of the scene (relationship among the properties and outlines of the scene) without performing any segmentation. They proposed five set of perceptual properties such as: Openness, expansion, naturalness, ruggedness and roughness to represent the spatial structure of the scene which are also meaningful to human perception. They showed the reliability estimation of these dimension by means of localized and spectral information. The keyframe's orientation histograms extract from a 4×4 grid size. Then, Gabor filters is used to compute the response of each cell of the grid and the final GIST feature vector are computed by concatenating these results.

Image representation: The next step of pre-processing module is vector quantization or construction of a codebook (set of vocabularies or visual words) as a representative feature descriptor from several similar raw extracted features. Different methods have been developed to generate these codebooks from all feature vectors.

Clustering method such as K-means is one of these techniques. Although the efficiency of K-means algorithm is satisfactory, its low discriminative ability and distortion error are some drawbacks of their generated codebooks (Shabou and LeBorgne, 2012). Sparse coding is an appropriate supervised method to construct codebooks with optimal sparse representation of local feature descriptors. Though, this method makes acceleration in the process, it is expensive in term of computational task.

Feature coding and feature pooling are two steps after generating codebooks. In feature coding some important codebooks are activated for each feature descriptor and the result is coding vector with the length equal to the codebooks number. Various coding algorithms act differently in the way of codebooks activation. The output of pooling stage is pooling vector which is the final representation of given keyframe by integrating all responses on each codebook. Among the above steps, the feature coding is the most important steps since it links the feature extraction and pooling.

Among different coding algorithm Locality-constrained Linear Coding (LLC) which was developed by Wang *et al*. (2010) is considered as the best choice due to its speed in coding and its accuracy in classification. The LLC encodes local features by conserving locality constraints in the feature space as well as the spatial domain.

The code vectors computed in the former stage need to be normalized because there are still lots of code vector; therefore using pooling techniques such as Spatial Pyramid Matching (SPM) is unavoidable. The SPM is one of the renowned and successful methods in image and scene classification as well as spatial pooling (Lazebnik *et al*., 2006). Figure 3 shows SPM partitions an image in $2^l \times 2^l$ (finer resolution) and works in

different scale l = 0, 1, 2, hence, in level 0 there is only one grid, in level 1 and 2 there are 4 and 16 grids of equal size, respectively. Then the histogram of feature within each grid is computed for all scale then weights each spatial histogram and finally concatenates them. Hence, the final representation of image includes:

$$\sum_{i=0}^{l} 2^i \times n$$

vector, where, n is the codebook length.

Video analysis module: The processes of video analysis module start with loading a new video and segmenting its related shots (an unbroken and continuous sequence of frames captured by one camera in short period of time) then extracting a set of representative frames called keyframes, from each shot. Once the keyframe extraction is completed the SIFT and GIST features are extracted from these keyframes. Then, the extracted features need to be encoded and pooled using LLC and SPM, finally these refined feature vectors are passed to the annotation module for further process.

Keyframe extraction: Keyframe extraction is the first step of video annotation processes. Keyframes are the significant frames with salient content and information that needs to be analyzed in the later steps. So, instead of processing all video frames only the extracted keyframes are processed for assigning annotations. A good keyframe extraction algorithm avoids reducing the numbers of frames to a scope that vital information could be lost.

There are different approaches for extracting keyframes, the simplest way is to select the first, middle, or the last frame of a given shot as keyframe. Han *et al.* (2000) proposed a method to use adaptive temporal sampling for extracting keyframes after detecting shot boundary by performing low pass filtering of histogram. Keyframes can be extracted by computing differences of RGB channels histogram for each successive frame and comparing with calculated threshold (Ahmed *et al.*, 1999). The other approach is to compute a motion metric by calculating optical flow for all frames.

In this study the required keyframes are extracted based on modified version of Khurana and Chandak (2013) algorithm. The edge differences among three consecutive frames are computed and the keyframes are selected comparing threshold.

Feature extraction and representation: Figure 4 summarizes the required steps to provide feature vectors for further processing. These processes are the same as those performed in pre-processing module. Therefore, same SIFT and GIST features are extracted then codebooks are generated and coded using LLC and then pooled with SPM.

Annotation module: The feature vectors, obtained in pre-processing step, are fed to a set of classifiers to construct concept detectors by training them. Consequently, annotation

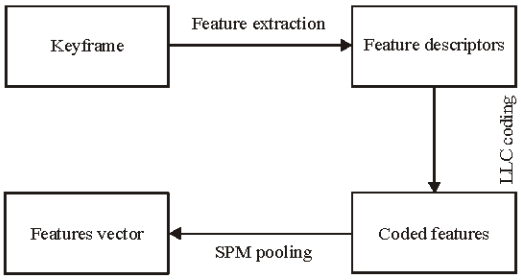

Fig. 4: Feature extraction and representation on keyframe

for the extracted keyframes of new video can be predicted using the provided feature vectors from video analysis module and these concept detectors.

This study used a very effective and renowned supervised classifier, Support Vector Machine (SVM) due to its generalization performance and superiority even in high dimensional space. The binary SVM performs classification by constructing a hyperplane with the largest distance to the nearest training samples of either class. Hence, the risk of misclassification of new samples (test samples) is minimized as much as this margin increases.

Here, in this study multi-class problem needs to be solved to predict annotation among various existing concepts (Columbia374 concepts) for new video. Mutli-class SVM problem is solved by reducing to multiple binary SVM with deploying One Against One (OAO) or One Against All (OAA) strategies.

RESULTS AND DISCUSSION

The effectiveness and efficiency of the proposed video annotation framework, AVAuCD, is evaluated for video annotation task on various Clipcanvas's video. The widely used video dataset, TRECVID 2005, along with concepts of Columbia374 are used for training concept detectors. The TREC Video Retrieval Evaluation (TRECVID) 2005, established by the National Institute of Standards and Technology (NIST), consists of 169 h of broadcast programming in English, Arabic and Chinese. Here, the development set of TRECVID 2005 (DEV) comprises of 137 multilingual broadcast news in 80 h with 61901 shots is used as training set in this study.

In the first step of training stage, the 374 annotated semantic concepts of Columbia374 are employed for annotating DEV's keyframes. Each keyframes extracted from shots in DEV based on presence or absence of each Columbia374's concept.

The SIFT and GIST features extracted from each keyframe for either training set or new videos. The dense local features of size 128-dimensional vectors are extracted using VL_Feat library with setting patch size to 16×16 pixels and dense grid to 6 pixels (Vedaldi and Fulkerson, 2010). In addition, the 512 dimensional GIST features are extracted from each keyframe for 4×4 spatial resolution in 8 orientations and 4 scales of Gabor filters.

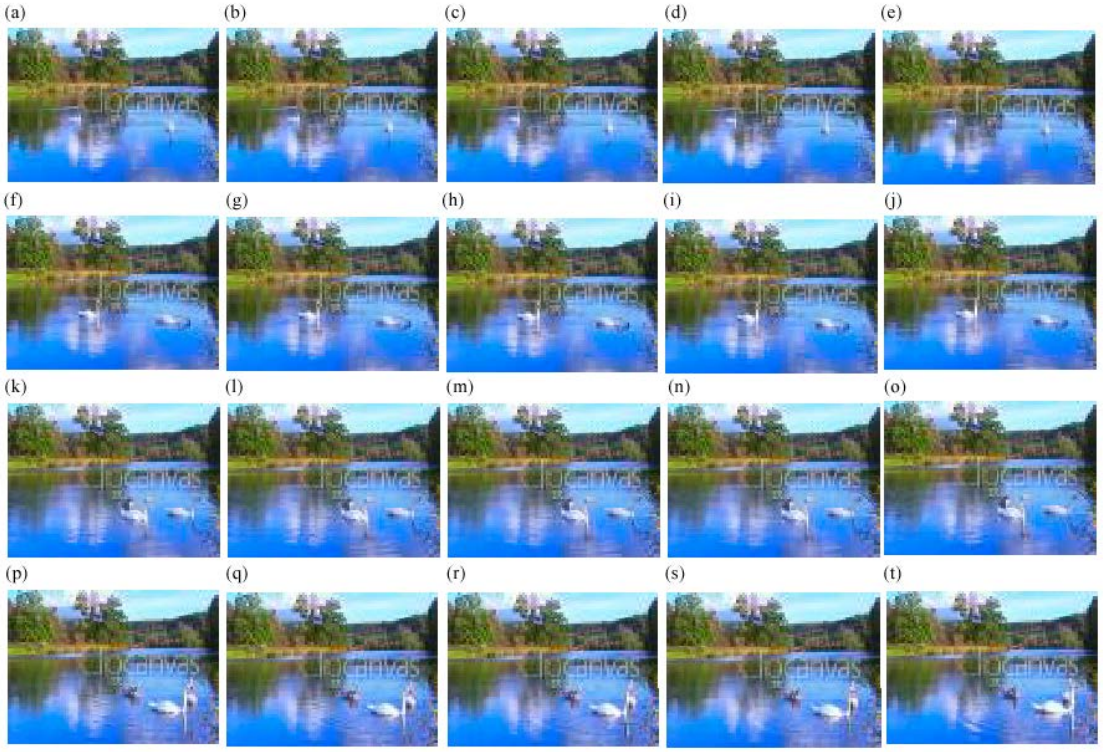

Fig. 5(a-t): Some frames (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, (f) 192, (g) 193, (h) 194, (i) 195, (j) 196, (k) 521, (l) 522, (m) 523, (n) 524, (o) 525, (p) 695, (q) 696, (r) 697, (s) 698 and (t) 699 of incoming video bird

The codebook of size (visual word) 1024 is generated and encoded by applying LLC considering numbers of neighbours equal to 5. The 3 levels SPM is performed on these encoded feature vectors and the final representation of keyframes are provided by concatenating the same weight vector computed from pooled vectors. LIBLINEAR library is used to train the concept detectors due to its efficiency on large-scale feature vectors (Fan *et al*., 2008). Here, the OAA strategy is selected during training step in order to support multi-class problem for assigning each 374 concepts to keyframes. Using this model, annotation can be predicted for every incoming (new) video.

To extract keyframes from video, the edge differences are used to compute the differences between current, former and consecutive frames. Those frames which their differences exceed the thresholds are considered as the representative frames or keyframes. Once a new video is given all its frames are read and converted in gray scale to compute their edge differences by utilizing Canny edge detector. The edge difference is chosen because of its dependency on content. The thresholds are computed using means and standard deviations of frame differences according to Eq. 1:

$$\text{Threshold} = \text{Mean} + a \times \text{standard deviation} \quad (1)$$

Khurana and Chandak (2013) chose a = 2 by various examinations. After all, those frames which their differences exceed these computed thresholds are considered as keyframes with significant variation in their content compare with the former and consecutive frames. Whilst, they computed only the differences among consecutive frames, here keyframes are extracted based on three frames differences.

We downloaded 5 various video types (videos: Bird, family, sport, traffic and airplane from wildlife, lifestyle, sport, traffic, transportation categories respectively) from 42 available video categories in Clipcavas. While the first downloaded video containing birds had 732 frames, as some of them are illustrated in Fig. 5, only very few of them are extracted as keyframes. Figure 6 shows these keyframes along with their frame numbers as well as the predicted annotation which are assigned to these keyframes after extracting their feature vectors using the same process as the pre-processing module and tested with trained concept detectors.

Performance of AVAuCD is evaluated by measuring recall and precision ratio as defined in Eq. 2 and 3:

Precision is the fraction of annotated keyframes that are relevant to actual annotation:

$$\text{Precision} = \frac{\text{Relevant annotated keyframes}}{\text{Relevant annotated keyframes} + \text{Irrelevant annotated keyframes}} \quad (2)$$

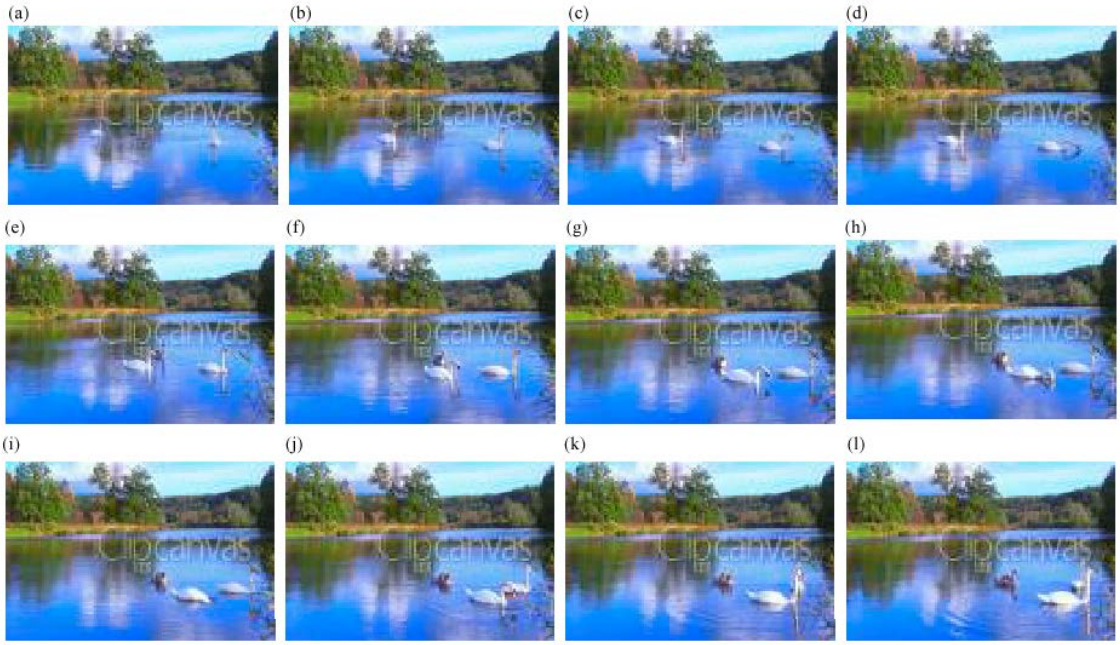

Fig. 6(a-l): Extracted keyframes from bird's video and their predicted annotation, (a) Frame 53, (b) 153, (c) 185, (d) 191, (e) 349, (f) 492, (g) 575, (h) 588, (i) 600, (j) 665, (k) 703 and (l) 732. Predicted annotation: Sky, birds, waterways, clouds, lakes, animal, trees

Table 1: Evaluation result of AVAuCD

Video	Video birds	Video airplane	Video sport	Video traffic	Video family
#Frames	732.00	767.00	747.00	322.00	585.0
#Keyframes	12.00	15.00	13.00	6.00	9.0
Recall (%)	100.00	100.00	100.00	100.00	100.0
Precision (%)	91.70	86.70	92.30	100.00	77.7
Processing time (sec)	16.82	18.01	17.35	14.65	15.4

Recall also known as sensitivity is the fraction of the annotated keyframes which are successfully annotated:

$$\text{Recall} = \frac{\text{Relevant annotated keyframes}}{\text{Relevant annotated keyframes} + \text{Not annotated keyframes}} \quad (3)$$

To define the relevancy of the predicted annotation of each video, we made the ground truth manually by specifying which predicted annotations for given video are relevant considering 374 concepts and samples from the training set. Moreover, the processing time is the time spent to predict annotation for new video and it is computed on a 2.6 GHz Intel processor in second.

Table 1 shows the ratio of annotation precision and recall over total 374 concepts. The result clearly shows that the sensitivity of AVAuCD is 100% but the precision depend on video types. As a result AVAuCD achieved average accuracy of 89.7% in predicting appropriate annotation.

CONCLUSION

This study presented a novel and efficient framework for automatic video annotation. The main contributions of this works are firstly, to extracted keyframes by computing edge differences among 3 consecutive keyfrmes; secondly, to train AVAuCD's concept detectors using 374 concepts of Columbia 374 over TRECVID dataset and lastly, to use the learned concept detectors for annotating new videos. The experimental result reveals that the proposed framework is sufficiently effective and promising for the video annotation tasks; nevertheless, the processing time need to be further reduced by employing faster supervised technique in annotation module or other feature extraction methods.

There are two directions for the future work: First, the performance of the proposed framework shall be evaluated by comparing the result of AVAuCD with other video annotation methods; second, the proposed framework can be integrated with the work of Memar *et al.* (2013) as one of the existing Content-Based Video Retrieval system to enhance the usability of AVAuCD.

ACKNOWLEDGMENT

This study is supported by the Prototype Development Research Grant Scheme (PRGS 552900) from the Ministry of Higher Education, Malaysia.

REFERENCES

Ahmed, M., A. Karmouch and S. Abu-Hakima, 1999. Key frame extraction and indexing for multimedia databases. Proceedings of the Visual Interface 99th Conference, May 19-21, 1999, Quebec Canada, pp: 506-511.

Bagdanov, A.D., M. Bertini, A. Del Bimbo, G. Serra and C. Torniai, 2007. Semantic annotation and retrieval of video events using multimedia ontologies. Proceedings of the International Conference on Semantic Computing, September 17-19, 2007, Irvine, CA., pp: 713-720.

Datta, R., D. Joshi, J. Li and J.Z. Wang, 2008. Image retrieval: Ideas, influences and trends of the new age. ACM Comput. Surv., Vol. 40. 10.1145/1348246.1348248

Dickinson, S.J., A. Leonardis, B. Schiele and M.J. Tarr, 2009. Object Categorization: Computer and Human Vision Perspectives. Cambridge University Press, New York, ISBN-13: 978-0521887380, Pages: 522.

Ding, G. and K. Qin, 2010. Semantic classifier based on compressed sensing for image and video annotation. Electron. Lett., 46: 417-419.

Dorado, A., J. Calic and E. Izquierdo, 2004. A rule-based video annotation system. IEEE Trans. Circuits Syst. Video Technol., 14: 622-633.

Fan, R.E., K.W. Chang, C.J. Hsieh, X.R. Wang and C.J. Lin, 2008. LIBLINEAR: A library for large linear classification. J. Mach. Learn. Res., 9: 1871-1874.

Han, S.H., K.J. Yoon and I.S. Kweono, 2000. A new technique for shot detection and key frames selection in histogram space. Proceedings of the 12th Workshop on Image Processing and Image Understanding, January 27-28, 2000, Jeju Island, South Korea, pp: 217-220.

Khurana, K. and M.B. Chandak, 2013. Key frame extraction methodology for video annotation. Int. J. Comput. Eng. Technol., 4: 221-228.

Lazebnik, S., C. Schmid and J. Ponce, 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, June 17-22, 2006, New York, USA., pp: 2169-2178.

Li, J. and J.Z. Wang, 2008. Real-time computerized annotation of pictures. IEEE Trans. Pattern Anal. Mach. Intell., 30: 985-1002.

Li, Y., J. Lu, Y. Zhang, R. Li and B. Zhou, 2009. A novel video annotation framework based on video object. Proceedings of the International Joint Conference on Artificial Intelligence, April 25-26, 2009, Hainan, China.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 60: 91-110.

Memar, S., L.S. Affendey, N. Mustapha, S.C. Doraisamy and M. Ektefa, 2013. An integrated semantic-based approach in concept based video retrieval. Multimedia Tools Applic., 64: 77-95.

Naphade, M.R. and J.R. Smith, 2004. Active learning for simultaneous annotation of multiple binary semantic concepts [video content analysis]. Proceedings of the IEEE International Conference on Multimedia and Expo, Volume 1, June 27-30, 2004, IEEE, pp: 77-80.

Oliva, A. and A. Torralba, 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision, 42: 145-175.

Pan, J.Y., H.J. Yang, P. Duygulu and C. Faloutsos, 2004. Automatic image captioning. Proceedings of the IEEE International Conference on Multimedia and Expo, Volume 3, June 27-30, 2004, IEEE, pp: 1987-1990.

Qiu, Y., G. Guan, Z. Wang and D. Feng, 2010. Improving news video annotation with semantic context. Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, December 1-30, 2010, Sydney, NSW., pp: 214-219.

Settles, B., M. Craven and L. Friedland, 2008. Active learning with real annotation costs. Proceedings of the NIPS Workshop on Cost-Sensitive Learning, pp: 1-10. http://burrsettles.com/pub/settles.nips08ws.pdf

Shabou, A. and H. LeBorgne, 2012. Locality-constrained and spatially regularized coding for scene categorization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, Providence, RI., pp: 3618-3625.

Tang, J., X.S. Hua, G.J. Qi, Z. Gu and X. Wu, 2007. Beyond accuracy: Typicality ranking for video annotation. Proceedings of the IEEE International Conference on Multimedia and Expo, July 2-5, 2007, Beijing, pp: 647-650.

Tao, D., D. Xu and X. Li, 2009. Semantic Mining Technologies for Multimedia Databases. 1st Edn., Information Science Reference, New York, ISBN-13: 978-1605661889, Pages: 550.

Vedaldi, A. and B. Fulkerson, 2010. VLFeat: An open and portable library of computer vision algorithms. Proceedings of the International Conference on Multimedia, October 25-29, 2010, Firenze, Italy, pp: 1469-1472.

Wang, J., J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, 2010. Locality-constrained linear coding for image classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA., pp: 3360-3367.

Wang, M., X.S. Hua, J. Tang and R. Hong, 2009. Beyond distance measurement: Constructing neighborhood similarity for video annotation. IEEE Trans. Multimedia, 11: 465-476.

Yanagawa, A., S.F. Chang, L. Kennedy and W. Hsu, 2007. Columbia University's baseline detectors for 374 lscom semantic visual concepts. ADVENT Technical Report, Columbia University, March 2007.

Zhang, H., 2003. Content-Based Video Analysis, Retrieval and Browsing. In: Multimedia Information Retrieval and Management, Feng, D., W.C. Siu and H.J. Zhang (Eds.). Springer, Berlin, Heidelberg, pp: 27-56.