Video Segments Retrieval Using A Hierarchy Built Over Semantic Attributes

No Author Given

No Institute Given

Abstract. There are a lot of multimedia contents like images and videos in today's digital world. To make good use of this content we need an efficient retrieval engine. An image captures one scene and can be described by a few words. However, a video on the other hand can contain images describing different concepts which may not be related at times. For example, a news video has various segments such as sports, politics and business. In this paper, we present an application to efficiently retrieve video segments from a set of videos, relevant to textual search query. To achieve this, key frames of each video are obtained and the video is segmented based on these frames. The key frames are annotated with concepts based on their visual and semantic properties. In order to build a scalable system, the tags are used to build a hierarchy by grouping common tags at each level. Given a search query, the hierarchy is traversed to find the relevant frames and the corresponding video segments. We present an experimental evaluation of the proposed system by using a set of test videos.

Keywords: Key frames extraction, video segment retrieval, multimedia retrieval, content based retrieval, hierarchy

1 Introduction

Searching videos from a large database is a challenging task. Conventional video search uses meta-data to retrieve relevant videos. This approach has several drawbacks. The meta-data is manually generated and may not completely describe the contents of the video. They may also fail to capture the semantics of the video. Most of these videos have heterogeneous contents. The relevant content might be a part of the whole video. The search would be more accurate if only these relevant video segments would be retrieved.

In this work, we have developed an application to retrieve relevant video segments from large videos. The primary step to achieve this involves understanding the semantics of the videos. The semantics can be represented as keywords or tags. There are two ways of obtaining these keywords. First approach is to use the images along with their tags available on popular image sharing sites like Flickr, Pinterest. But, it is expensive to obtain large amount of labeled data as

it demands a significant human labor. Another approach involves using unsupervised learning methods to understand the contents of unlabeled images. Deep learning techniques have proven to be better at this task.

For content-based retrieval of images, Deng et al.[1] show that hierarchical relationships can significantly improve the retrieval process. Incorporating hierarchical relationships is becoming important as datasets grow larger. In our application, a hierarchy is built exploiting the frequency of tags.

The rest of the paper is organized as follows. Section 2 presents related work in the area and in section 3, we discuss the proposed approach. Section 4 presents a discussion on the results obtained by applying the proposed method followed by conclusion and future work in section 5.

2 Related Work

In this section we briefly explain the related work on video annotation and video segment retrieval methods. As videos can be represented as a set of key frames or images, work on image retrieval are also explored.

Several attempts have been made to understand the content of images/videos and organize them in different ways to make retrieval fast and efficient. Understanding the semantics of the images is really important for retrieval. Feng *et al* [2] proposed a multiple Bernoulli relevance model for automatically annotating images. The relevance model is a joint probability distribution of the word annotations and the image feature vectors and is computed using the training set. The word probabilities are estimated using a multiple Bernoulli model and the image feature probabilities using a non-parametric kernel density estimate. In this work, the deep learning technique of convolutional neural networks is used through an API called Clarifai [10]. The drawback of Bernoulli relevance method is that it works only for a single query whereas the proposed hierarchical structure can handle multiple word queries as well.

There are many ways in which the annotated data is organized. Work has been done on hierarchical relationships between classes. It has been found to have an impact on recognition performance and the potential to improve recognition accuracy. This was demonstrated by putting existing datasets into hierarchies by Griffin *et al* [5]. Two types of hierarchies have recently been explored in computer vision. Language-based hierarchy which includes semantic organization of images such as in [4] and low-level visual feature based hierarchy which consider visual information that connects images together such as in [3]. A method to automatically discover the *semantivisual* image hierarchy by incorporating both image and tag information was proposed by Li *et al* [6]. Santhana Krishnamachari *et al* [7] designed a clustering based indexing technique, where the images in the database are grouped into clusters of images with similar color content using a hierarchical clustering algorithm. Experiments with different database sizes showed that the number of similarity comparisons required to achieve a given retrieval accuracy does not increase linearly with the size of the database thus

making the algorithm scalable to large databases. We take inspiration from these content based methods to develop a search mechanism using a hierarchy built over it.

3 Proposed Work

In this section, we describe the working of the proposed application and the different steps involved in achieving the results.

3.1 Pre-processing

The preliminary step is to extract key/representative frames and use them to segment the videos. A random set of videos is accumulated from various video sources like YouTube and Netflix. Key frames are extracted using the absolute difference method. Key frames extraction involves quantifying the difference between consecutive frames in a video. Some of the techniques used include color histogram, frame correlations, edge histogram, Flexible Rectangles algorithm, Adaptive Sampling algorithm and Shot Reconstruction Degree Interpolation. In this work, *absolute difference* method is utilized. In this method, the absolute difference between consecutive frames is calculated using the pixel values of the frames. A linear combination of the mean and standard deviation of these differences are computed and are used to set a manual threshold.

If the difference between the frames is lesser than the threshold, the frames are assumed to be similar. If not, there is a significant change in the content of the frames. The video is segmented at this location and the key frame associated with this segment is saved. After an entire video is analyzed, a group of segments and key frames are obtained.

3.2 Annotation of key frames

The next objective is to understand the contents of the key frames of the videos. These are annotated with concepts or tags to add semantics by using the Clarifai API[10]. Clarifai is the ILSVRC-2013 winning submission[8] with an error of 11.2% with outside training data and 11.7% without it. It uses Convolutional Neural Networks (CNN), a Deep Learning architecture. CNN are hierarchical machine learning models which learn a complex representation of images using vast amounts of data. They are inspired by the human visual system and learn multiple layers of transformations, which extract a progressively more sophisticated representation of the input. The API takes an image as input and returns a set of tags as output along with the probabilities of their relevance to the content of the image. Every key frame is annotated using the API.

3.3 Construction of hierarchy

Each key frame has a set of tags associated with it. Given a textual query, a simple way to retrieve the relevant frames is to linearly search through the tags of every frame. But, many of these frames will have common tags. This can be used to improve the efficiency of the retrieval. We propose a methodology to construct a hierarchy which clusters frames which have common tags, iteratively i.e. the hierarchy will be of multiple levels. Algorithm 1 presented below is used for construction of hierarchy. The hierarchy is constructed in a bottom-up manner.

Algorithm 1 Hierarchy construction

Input: Set of frame numbers $F(1)...F(n)$ and the tags associated with them
Output: A hierarchical structure
Let L be the level of the hierarchy. Initialize $L = 1$
for each pair of frames $F(i)F(j)$ with set of tags $T(i) = \{t_{i_1}, t_{i_2}....t_{i_x}\}$ and $T(j) = \{t_{j_1}, t_{j_2}....t_{j_y}\}$ **do**
    if there are some common tags **then**
        Create a new node N_k in level 1 with
        Frames, $F(N_k)$=$F(i) \cup F(j)$ and
        Tags,$T(N_k)$=$T(i) \cap T(j)$
    end if
end for
while some common tags are present among nodes at level L **do**
    for each pair of nodes $< N_i, N_j >$ in level L **do**
        if $T(N_i) = T(N_j)$ **then**
            Merge N_i and N_j with tags=$T(N_i)$ and
            frames=$F(N_i) \cup F(N_j)$
        end if
    end for
    for each pair of nodes $< N_i, N_j >$ in level L **do**
        if If there are come common tags **then**
            Create a new node $N(k)$ in level $L + 1$ with
            $F(N_k)$=$F(N_i) \cup F(N_j)$ and $T(N_k)$=$T(i) \cap T(j)$
        end if
    end for
    $L = L + 1$
end while

Pairs of frames are compared and new clusters are formed containing the largest subset of common tags. If two clusters at the same level have the same set of tags, they are merged to form a single cluster. As the level of the hierarchy increases, fewer common tags will be encountered. The top level will have clusters whose intersection of tags is a null set. This structure will also give us the most common tags on top of the hierarchy and the relationship between the tags as we move down the hierarchy. To accommodate large number of frames multiple hierarchies are constructed.

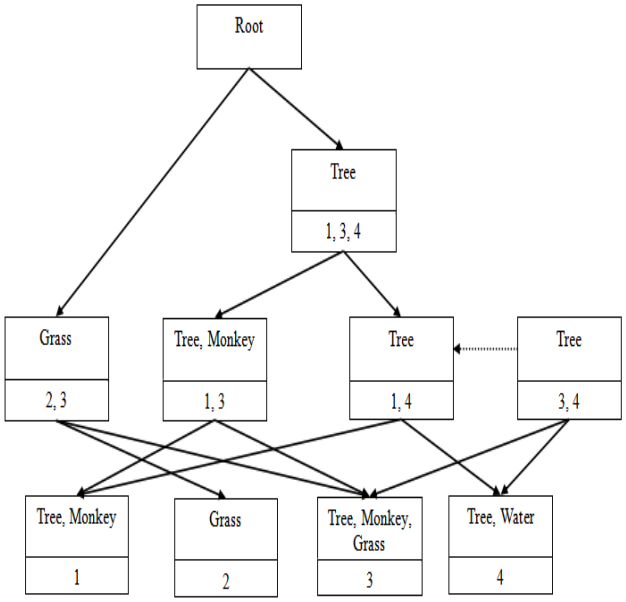

Fig. 1. A sample hierarchy to demonstrate the idea

Fig. 1 shows an example where there are 4 frames, numbered *1, 2, 3* and *4*. Their corresponding tags are *[Tree, Monkey]*, *[Grass]*, *[Tree, Monkey, Grass]* and *[Tree,Water]*. These 4 frames will be the leaves of the tree. At every level of the hierarchy, we compare every pair of frames to find common tags and cluster these tags in the next level.

When frames 1 and 4 are compared, the common tag is *Tree*. A new cluster containing *Tree* as the tag and frame numbers 1 and 4 is formed at the next level. When frames 1 and 3 are compared, the largest subset of common tags is *Tree, Monkey* which forms a new cluster. Similarly, all pairs of frames are compared and the next level is formed. In level 2, there are two clusters with the same tag *Tree* but with frame numbers *[1,4]* and *[3,4]*. These two clusters are merged into a single cluster. The same procedure is followed at every level.

At each level from the second onward, clusters which have no tags in common with any other cluster are added to the root. In the example above, the cluster *Grass* in level 2 has no tags in common with any other cluster and is hence added to the root. The hierarchy is complete when all the clusters at a level are mutually exclusive as shown in level 3 of the example.

The hierarchies are constructed only once for a set of frames. Hence, they need to be stored permanently. One of the simple, easy to use and understandable formats is the Newick tree format [9]. This format is used to represent the hierarchies as text strings.

3.4 Searching through the hierarchy

Search algorithm Construction of the hierarchy is bottom-up whereas search through the hierarchy is top-down. When a textual query is given, the hierarchy constructed is exploited to retrieve the most relevant video segments. Once a query is given, a breadth-first search is performed till a node containing one or more query term is found. After finding this node, breadth-first search is performed on its sub-tree to find the remaining set of query terms. This procedure is repeated until all the terms are found or till the end of the hierarchy. This

Algorithm 2 Searching through the hierarchy

Input: The hierarchy and query terms $Q_1, Q_2....Q_x$
Let Q be the set of query words, and let $\mathbb{P}_{>1}(Q)$ be the power set of Q
Output: $R = F(N_i) \mid \mathbb{P}_{>1}(Q) \in F(N_i)$, where R is the resultant set of frames
Starting search from root, current node $N = root$
procedure BFS(N)
    for each node N_i the descendant of N, **do**
        if $Q_k \in F(N_i) \forall Q_k \in Q$ **then**
            Add $F(N_i)$ to R
            Remove query word Q_k from Q
            if Q is empty **then**
                return R
            else
                return BFS(N_i)
            end if
        end if
    end for
end procedure

algorithm works by reducing the search space when each query term is found. Search is carried forward on the subtree with reduced query set. For example, if query is *Tree, Monkey* using the hierarchy shown in Fig. 1 *Tree* is found first, the subtree with *Tree* as the root is the new search space. BFS with query as *Monkey*, then returns frames 1,3.

Ranking results Once the frames for a query are retrieved, they are *ranked* before being displayed. The frames with larger subset of query terms are ranked higher in the results. For the frames pertaining to subsets of the same length, the probabilities of query terms in each result frame are used. This probability signifies the relevance of the query terms to the contents of the frame and is obtained with the output of Clarifai API. When multiple frames are retrieved, the average probability corresponding to the query terms are used for ranking.

4 Results and Analysis

Performance evaluation has long been a difficult problem in content-based retrieval. This is primarily due to lack of relevant quantitative measures for evaluation. In content-based retrieval, precision and recall measures have been frequently used to evaluate the performance of retrieval algorithms. In our case, there is no need to evaluate the subjective quality of the retrieval, but it is only necessary to compare the retrieval with clustering against the linear(exhaustive) search.

The metric used to measure the efficiency of the proposed method is the number of comparisons that occur during the search process. We prepared a dataset of **702** video segments extracted from a set of test videos obtained from YouTube and Netflix. Table 1 shows the number of comparisons that were made to retrieve video segments for certain query words using the hierarchy and the performance gain when compared to linear search. The performance gain is measured using equation (1) where L is the number of comparisons using linear search and H is the number of comparisons using the hierarchy.

$$PerformanceGain = \frac{L}{H} \tag{1}$$

Table 1. Efficiency of the retrieval engine

Query	Hierarchy	Performance Gain
Landscape	87	8x
Portrait	93	7.5x
Water	132	5.3x
Nature, Sky	156	4.5x
Landscape, Sky, Water	323	2.1x

There does not exist a standard video test set to measure retrieval performance nor standard benchmarks to measure system performance. Thus we compare our method with the brute-force method of linear search where the number of node comparisons would be the total number of keyframes/video segments (702, in this case) as every node has to be processed. Whereas using the hierarchy constructed, we can significantly reduce the number of comparisons as shown above in Table 1. The retrieval is efficient in situations where the tags are very common in the dataset. If there are multiple query words, the search uses very less comparison when the query words occur together frequently. Hence, the efficiency of retrieval depends on the query.

5 Conclusions

We developed a hierarchy for efficient retrieval of video segments for a given textual query. The hierarchy is built on tags annotated to the key frames extracted from the videos. The key frames are annotated using their visual and semantic properties. We observed that searching using the hierarchy is most efficient when the query terms occur frequently and are related. Currently, due to resource constraints, we are forced to construct multiple hierarchies to accommodate large video sets.

Faster retrieval can be achieved if a single hierarchy is constructed for the entire dataset. The work can also be improvised by parallelizing the search through the multiple hierarchies. The search can be made robust by using the ontology of query words.

Bibliography

[1] Deng, Jia and Berg, Alexander C and Fei-Fei, Li, *Hierarchical semantic indexing for large scale image retrieval.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 785-792. IEEE, 2011.

[2] Feng, SL and Manmatha, Raghavan and Lavrenko, Victor *Multiple bernoulli relevance models for image and video annotation.* Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR) , pages II–1002. IEEE, 2004.

[3] Bart, Evgeniy and Porteous, Ian and Perona, Pietro and Welling, Max, *Unsupervised learning of visual taxonomies.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8. IEEE, 2008.

[4] Jin, Yohan and Khan, Latifur and Wang, Lei and Awad, Mamoun, *Image annotations by combining multiple evidence & wordnet.* Proceedings of the 13th annual ACM international conference on Multimedia, pages 706–715. ACM, 2005.

[5] Griffin, Gregory and Perona, Pietro, *Learning and using taxonomies for fast visual categorization.* Computer Vision and Pattern Recognition (CVPR), pages 1–8. IEEE, 2008.

[6] Li, Li-Jia and Wang, Chong and Lim, Yongwhan and Blei, David M and Fei-Fei, Li, *Building and using a semantivisual image hierarchy.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3336–3343. IEEE, 2010.

[7] Krishnamachari, Santhana and Abdel-Mottaleb, Mohamed, *Hierarchical clustering algorithm for fast image retrieval.* Electronic Imaging'99, pages 427–435. International Society for Optics and Photonics, 1998.

[8] Stanford Vision Lab, *Large Scale Visual Recognition Challenge 2013 (ILSVRC 2013).* http://www.image-net.org/challenges/LSVRC/2013/results.php#cls

[9] Archie, James and Day, William HE and Felsenstein, Joseph and Maddison, Wayne and Meacham, Christopher and Rohlf, F James and Swofford, David, *The Newick tree format.* http://evolution.genetics.washington.edu/phylip/newicktree.html , 1986

[10] Matthew Zeiler, *Clarifai: Amplifying Intelligence.* http://clarifai.com.