A Region Thesaurus Approach for High-Level Concept Detection in the Natural Disaster Domain

Evaggelos Spyrou and Yannis Avrithis

Image, Video and Multimedia Systems Laboratory,
School of Electrical and Computer Engineering
National Technical University of Athens
9 Iroon Polytechniou Str., 157 80 Athens, Greece,
espyrou@image.ece.ntua.gr,
WWW home page: http://www.image.ece.ntua.gr/∼espyrou/

Abstract. This paper presents an approach on high-level feature detection using a region thesaurus. MPEG-7 features are locally extracted from segmented regions and for a large set of images. A hierarchical clustering approach is applied and a relatively small number of region types is selected. This set of region types defines the region thesaurus. Using this thesaurus, low-level features are mapped to high-level concepts as model vectors. This representation is then used to train support vector machine-based feature detectors. As a next step, latent semantic analysis is applied on the model vectors, to further improve the analysis performance. High-level concepts detected derive from the natural disaster domain.

1 Introduction

High-level concept detection in both image and video documents remains still an unsolved problem. However, due to the continuously growing volume of audiovisual content, this problem attracts a lot of interest within the multimedia research community. Its two main and most interesting aspects appear the selection of the low-level features that will be extracted and the method that will be used for assigning these low-level descriptions to high-level concepts. The latter problem is commonly referred to as the "Semantic Gap". Many approaches have been proposed that share the target of bridging this semantic gap, in other words facilitating the mapping between low and high level features within multimedia documents.

There have been a lot of approaches to fulfil the need both for low-level feature extraction and for using them effectively in order to use them for detecting high-level concepts. The first have led to the development of many descriptors that aim to capture the audio, color, texture, shape and motion characteristics of audiovisual documents. The latter have investigated the application of many machine learning techniques such as neural networks, fuzzy systems, support

vector machines and so on. These approaches aim to exploit the similarities that occur between different instances of the same high-level concept.

Another aspect of the motivation of this work has been the increasing number of available "global" annotations. It is generally easier for a user to annotated an image globally, that is defining the existence of a high-level concept within it, rather than locally, with defining the region wherein lies the concept. One example of such an annotation is the LSCOM workshop annotation [1], where a huge number of shots of news bulletins are globally annotated for a large number of concepts. On the other hand, annotated data sets per region are very rare. Special note should be given to an effort to effectively evaluate and benchmark various approaches in the field of information retrieval, by the TREC conference series, during the last few years. Within this series the TRECVID [2] evaluation attracts many organizations and research interested in comparing their research in tasks such as automatic segmentation, indexing, and content-based retrieval of digital video. For the high-level feature detection task of TRECVID, global annotations have been offered by different organizations and a huge database of video keyframes has been available to active participants.

One of the most famous systems for multimedia analysis and retrieval is presented in [3]. This prototype system uses multi-modal machine learning techniques in order to model semantic concepts in video, from automatically extracted multimedia content. Moreover, in [4], a region-based approach in content retrieval that uses Latent Semantic Indexing (LSI) techniques is presented. It also appears crucial for good analysis results the choice of global or local visual features. In order to exploit the spatial content of a keyframe, the extraction of low-level concepts is performed after the image is modeled using grids, thus color and texture features are selected locally [5]. A similar approach is presented in [6]. Here, the features are extracted by regions of an image that resulted using a mean shift algorithm. Also, in [7], a region-based approach is presented, that uses knowledge encoded in the form of an ontology. MPEG-7 visual features are extracted and combined and high-level concepts are detected.

Moreover, a hybrid thesaurus approach is presented in [8], where, semantic object recognition and identification within video news archives is achieved, with emphasis to face detection and TV channel logos. Another form of a thesaurus, a lexicon, is used in an approach for an interactive video retrieval system is presented in [9], where the core is the automatic detection of an unprecedented lexicon of 101 concepts. Another lexicon design for semantic indexing in media databases is also presented in [10]. In the same context, [11] presents an approach for texture and object recognition that uses scale- or affine-invariant local image features in combination with a discriminative classifier.Support vector machines have been used for image classification based on their histogram as in [12] and for the detection of semantic concepts such as goal, yellow card and substitution in the soccer domain [13]. A Self-Organized map that uses MPEG-7 features is presented in [14]. Within this work, content-based image and information retrieval is achieved in large non-annotated image databases.

In this work, the problem of concept detection in video is approached in the following way: First, a segmentation algorithm is applied on the image. The algorithm is tuned to produce a coarse segmentation, with a small number of regions. For each region, several color and texture MPEG-7 descriptors are extracted. Fusing all low-level features, a feature vector for each image is formed. Using a significantly large number of images and applying a hierarchical clustering method, a region thesaurus is constructed that contains all the commonly occurred region types. These region types may or may not represent the concepts that are chosen to be detected. Each region type of the region thesaurus contains the appropriate merged color and texture description. By measuring the distances of the regions of an image to the region types, a model vector is formed that captures the semantics of an image in terms of the region thesaurus. A support vector machine-based feature detector is then trained to detect each high-level semantic concept, based on the values of the model vectors. Finally, in an effort to exploit the co-occurrences of the region types to the images, latent semantic indexing is applied. Feature detectors are trained again based on the projections of the model vector to the concept space.

This paper is organized as follows: Section 2 presents the method used for the extraction of the color and texture features of a given keyframe. The method for the construction of the region thesaurus containing the most frequent region types within the training set is presented in section 3, followed by the construction of the model vectors that include the semantic image features in section 4. The application of Latent Semantic Indexing techniques is presented in section 5. Section 6 presents SVM-based high-level feature (concept) detectors, followed by experimental results in section 7. Finally, conclusions are drawn in section 8 accompanied by plans for future work.

2 Low-Level Feature Extraction

For the representation of the low-level features of a given image, descriptors from the ISO/IEC MPEG-7 standard [15] were used. These descriptors are designed to capture a standardized and non-textual description of multimedia content, defining the syntax and the semantics of an elementary feature, aiming to provide interoperability among applications that use audio-visual content descriptions. This section presents the descriptor extraction procedure followed in this approach.

2.1 Image Segmentation

For the extraction of the low-level features of an image, there are generally two categories of approaches:

- Extract the desired descriptors *globally* (from the entire image)
- Extract the desired descriptors *locally* (from regions of interest)

Fig. 1. An input image and its coarse segmentation.

While global descriptor extraction appears a trivial task, since MPEG-7 descriptors already have such an option, extracting descriptors locally may turn out a more complex task, since there does not exist neither a standardized way of defining a given image to regions, from which the features are to be extracted, nor a predefined method to combine and use those features. There have been some approaches on the extraction within orthogonal grids of the images. Some of them split the image in a fixed number of regions and consider each one as an entire image. Others use a moving block and extract features from overlapping orthogonal regions. Moreover, there also exist approaches that apply a clustering method on the pixels of the image and extract descriptors from each cluster as in our previous work [16]. In the presented approach, a color segmentation algorithm is first applied on a given image as a pre-processing step. The algorithm is a multiresolution implementation [17] of the well-known RSST method [18] tuned to produce a coarse segmentation. This way, the produced segmentation can intuitively provide a qualitative description of the image. To explain this, an input image along with its coarse segmentation is depicted in figure 1

In this example, one could easily describe the input image as a set of regions. This is subjective to the user as it is also subjective to the segmentation. i.e. here a user could see "a light blue region" (sky), "two green regions" (vegetation), "an orange region" (fire) etc, in a same sense as to the output of a coarse segmentation. This idea leads to the definition of a set of images that will contain all possible types of regions and will act as the means to bridge the gap between the low-level descriptors and the high-level features.

After splitting the image in a small number of regions of interest, low-level descriptors of color and texture features are extracted from each region separately, as presented in subsections 2.2 and 2.3.

2.2 Color Features

For the representation of the color features of the image regions, three MPEG-7 color descriptors are used: The *Color Layout Descriptor*, the *Scalable Color*

Descriptor and the *Color Structure Descriptor*. For the extraction of the aforementioned descriptors, the eXperimentation Model (XM)[19] of the MPEG-7 is used. More specifically:

Color Layout Descriptor (CLD) is a compact MPEG-7 visual descriptor designed to represent the spatial distribution of color in the YCbCr color space. It can be used globally in an image or locally in an arbitrary-shaped region of interest. The given picture or region of interest is divided into $8 \times 8 = 64$ blocks and the average color of each block is calculated as its representative color. A discrete cosine transformation is then performed into the series of the average colors and a few low-frequency coefficients are selected using zigzag scanning. The CLD is formed after quantization of the the extracted coefficients. In conclusion, the CLD appears an effective descriptor in applications such as sketch-based image retrieval, content filtering using image indexing and visualization.

The Color Layout Descriptor is formed as:

$$CLD = [\{DY_{DC}, DY_{AC_i}\}, \{DCr_{DC}, DCr_{AC_j}\}, \{DCb_{DC}, DCb_{AC_k}\}] \quad (1)$$

where i, j and k denote the number of AC coefficients and can its possible values are 3, 6, 10, 15, 21, 28 and 64. Within the presented approach, the number of Y AC coefficients is fixed to 5, while the number of Cb and Cr AC coefficients to 3.

Scalable Color Descriptor (SCD) is a Haar-transform based encoding scheme that measures color distribution over an entire image or region of interest. The color space used is the HSV, quantized uniformly to 256 bins using its largest scale. To sufficiently reduce the large size of this representation, the histograms are encoded using a Haar transform allowing also the desired scalability. Experimental results prove that a reasonable performance can be achieved using even a small number of bits.

The Scalable Color Descriptor is a histogram descriptor and is formed as:

$$SCD = [C_1, C_2, \ldots, C_N] \quad (2)$$

where N denotes the number of the bins of the histogram. In this approach $N = 64$.

Color Structure Descriptor (CSD) captures both the global color features of an images and the local spatial structure of the color. The latter feature of the CSD provides the descriptor the ability to discriminate between images that have the same global color features but different structure, thus a single global color histogram would fail. An 8×8 structuring element scans the image and the number of times a certain color is found within it is counted. This way, the local color structure of an image is expressed in the form of a "color structure histogram". This histogram is identical in form to a color histogram, but is semantically different. The color representation is given in the HMMD color space. The CSD is defined using four color space quantization operating points: 256, 128, 64, and 32 bins, to allow scalability while the size of the structuring element is kept fixed.

The Color Structure Descriptor is formed as:

$$CSD = [\overline{h}_s(m)], m \in 1, \ldots, M \tag{3}$$

where M denotes the number of bins and s the size of the structuring element. The value in each bin represents the number of occurrences of structuring elements as they scan the image, that contain at least one pixel with color c_M. In this case $M = 256$ and $s = 8$.

2.3 Texture Features

To efficiently capture the texture features of an image, the MPEG-7 Homogeneous Texture Descriptor (HTD) [20] is applied, since it provides a quintative characterization of texture and comprises a robust and easy to compute descriptor. The image is first filtered with orientation and scale sensitive filters. The mean and standard deviation of the filtered outputs are computed in the frequency domain. The frequency space is divided in 30 or 60 channels, as described in [19], and the energy and energy deviation of each channel are computed and logarithmically scaled.

The texture description for a region of an image is then formed as follows:

$$HTD = [f_{DC}, f_{SD}, e_1, e_2, ..., e_N, d_1, d_2, ..., d_N] \tag{4}$$

Where f_{DC} and f_{SD} denote the mean and the standard deviation of the image texture respectively and N denotes the number of channels, which in this case is $N = 30$. The energy deviation of each channel is discarded, in order to simplify the description since it generally does not affect the quality of the results.

3 Region Thesaurus Construction

3.1 Some Initial Observations

Given the entire training set of images and their extracted low-level features as described in section 2, one can easily observe that those regions that belong to similar semantic concepts, also have similar low-level descriptions and also those images that contain the same high-level concepts are consisted of similar regions.

As it becomes obvious, this region similarity can be exploited as region coexistences often characterize the concepts that exist. As a result of the aforementioned observations, clustering is performed on all the different regions that are encountered within the images of the given training set. The goal of this clustering approach is to identify the more often regions that are encountered within the training set, and use them to build a knowledge base. These regions will be referred to as "region types".

3.2 Hierarchical Clustering

As it appears rather obvious, one cannot have a priori knowledge for the exact number of the required classes, in order to effectively cluster the heterogeneous types of image regions. Thus, a K-means or a Fuzzy C-means clustering approach does not appear useful enough, since in the case that more clusters are needed the algorithm should run from the beginning and their results are always different, as they begin with a random guess of the cluster centroids and the number of the clusters remains the same during every step of the algorithm.

In an attempt to overrun this problem, *Hierarchical clustering* [21] is the method chosen to be applied on the low-level description set, since after the clustering, one can easily select the number of clusters to keep and easily modify it without re-applying the algorithm.

In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to N clusters each containing a single object. Hierarchical clustering is subdivided into agglomerative methods, which proceed by series of fusions of the N objects into groups, and divisive methods, which separate N objects successively into finer groupings. In this approach an agglomerative approach has been adopted.

After the separation of the regions of the training set into clusters, an initial observation is that each cluster may or may not contain regions from the same high-level feature and regions from the same high-level feature may be encountered in more than one clusters. For example, the high-level concept *vegetation* can have more than one instances differing in i.e. the color of the leaves of trees. Each instance will be represented by the centroid of a cluster and will be referred to as region type. Moreover, in a cluster that contains instances from the semantic entity i.e. *sea*, these instances could be mixed up with parts from i.e. *sky*, since a typical sky region is almost identical to certain sea regions.

An example from the application of hierarchical clustering on a small number of images and for different numbers of clusters is depicted on figure 2. The clustering process starts with 9 different regions and groups them into pairs. The binary tree that results facilitates the determination of the number of the clusters and allows easy modification of this choice without re-application of the method on the data. In this case, the choice of the number of the clusters is set to 4 and the region types that occur are depicted.

3.3 Region Thesaurus

Generally, a thesaurus combines a list of every term in a given domain of knowledge and a set of related terms for each term in the list which are the synonyms of the current term. In our approach, the constructed "Region Thesaurus" contains all the "Region Types" that are encountered in the training set. These region types are the centroids of the clusters and all the other feature vectors of a cluster are their synonyms. By using a significantly large training set of keyframes, the

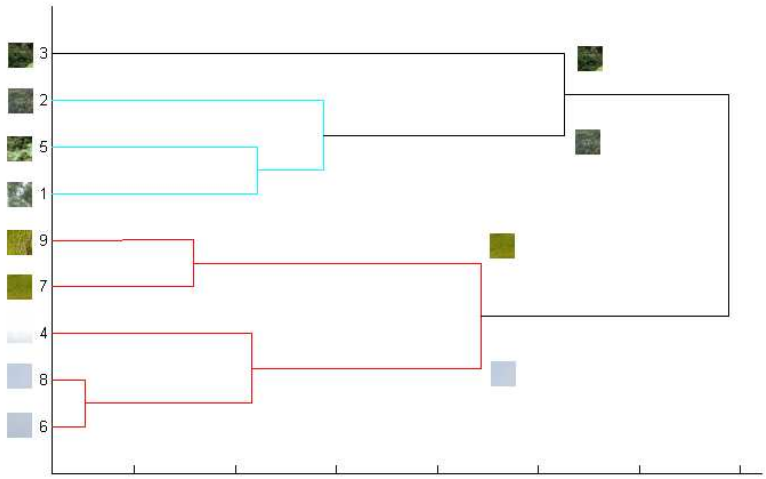

Fig. 2. A dendrogram showing region type selection using hierarchical clustering

thesaurus is constructed. Its purpose is to formalize a conceptualization between the low and the high-level features and facilitate their association.

Each region type is represented as a feature vector that contains all the extracted low-level information for it. As it is obvious, a low-level descriptor does not carry any semantic information. It only constitutes a formal representation of the extracted visual features of the region. On the other hand, a high-level concept carries only semantic information. A region type lies in-between those features. It contains the necessary information to formally describe the color and texture features, but can also be described with a "lower" description than the high-level concepts. I.e., one can describe a region type as "a green region with a coarse texture".

4 Model Vectors for representing a given image

This section presents the basic distance measures used for comparing two low-level descriptors and a simple algorithm for describing an image segmented coarsely in regions, with the aid of the constructed visual thesaurus.

4.1 Dissimilarity Functions for MPEG-7 Visual Descriptors

The distance between 2 *Homogeneous Texture Descriptors* is computed as:

$$D(HTD_1, HTD_2) = \sum_{k=1}^{N} \left| \frac{HTD_1(k) - HTD_2(k)}{d_{norm}^{HTD}} \right| \tag{5}$$

where $a(k)$ is the standard deviation of $TD_{database}(k)$ for a given database. However, the MPEG-7 standard does not strictly define the normalization factor to be used within this distance function. In the presented approach the normalization factor used is:

$$d_{norm}^{HTD} = maxD(HTD_1, HTD_2) = 7680 \tag{6}$$

For the *Color Layout Descriptor*, MPEG-7 suggests the following distance measure:

$$D(CLD_1, CLD_2) = \frac{1}{d_{norm}^{CLD}} \sqrt{\sum_i w_{yi}(DY_i^1 - DY_i^2)^2 + \sum_j w_{rj}(DCr_j^1 - DCr_j^2)^2 + \sum_k w_{bk}(DCb_k^1 - DCb_k^2)^2} \tag{7}$$

where DY_i, DCb_i, DCr_i represent the i-th DCT coefficients of the respective color components. The normalization factor applied on this distance function is chosen to be:

$$d_{norm}^{CLD} = maxD(CLD_1, CLD_2) = 96 \tag{8}$$

Finally, the remaining color descriptors, Color Structure and Scalable Color are histograms. The well-known L2 distance which is applicable very often on histogram comparisons is applied on them and given by:

$$D(CSD_1, CSD_2) = \frac{1}{d_{norm}^{CSD}} \sum_{i=1}^{N} |\overline{h}_s^1(i) - \overline{h}_s^1(i)| \tag{9}$$

and

$$D(SCD_1, SCD_2) = \frac{1}{d_{norm}^{SCD}} \sum_{i=1}^{N} |C_1(i) - C_2(i)| \tag{10}$$

respectively. The MPEG-7 standard does not propose any normalization factor for these aforementioned descriptors. Driven by some experimental results, this approaches chooses:

$$d_{norm}^{CSD} = d_{norm}^{SCD} = 8192 \tag{11}$$

respectively. At this level of experiments all visual descriptors are given the same weight when they are combined in order to compare two different image regions. However, in equation **??** this combination is presented as a weighted sum.

$$d_{all} = w_{HTD}D_{HTD} + w_{CLD}DCLD + w_{SCD}DSCD + w_{CSD}DCSD \tag{12}$$

However we should notice that the MPEG-7 standard does not strictly define the distance functions to be used, thus leaving the developers the flexibility to develop their own dissimilarity/distance functions and to exploit other well-known similarity functions such as the Euclidean, the Minkowski or the City-Block Distance. Also, there has not been a standard method to combine these distances. Some ideas on this problem have been presented in [22].

4.2 From Low-Level features to Model Vectors

Having calculated the distance of each region (cluster) of the image to all the words of the constructed thesaurus, the model vector that semantically describes the visual content of the image is formed by keeping the smaller distance for each high-level concept. More specifically, let: $d_i^1, d_i^2, ..., d_i^j, i = 1, 2, 3, 4$ and $j = N_C$, where N_C denotes the number of words of the lexicon and d_i^j is the distance of the i-th region of the clustered image to the j-th region type. Then, the model vector D_m is the one depicted in equation 13.

$$D_m = [min\{d_i^1\}, min\{d_i^2\}, ..., min\{d_i^{N_C}\}], i = 1, 2, 3, 4 \tag{13}$$

4.3 Principal Component Analysis

Since the number of the region types can be very large depending on the complexity of the training set and the variations of the regions types, the dimensionality of the model vector may become very high. It is then possible that the extracted region types may carry redundant information. It is, thus, possible that two region types may be strongly correlated although they may appear different. To avoid this, principal component analysis (PCA) is applied in order to reduce the dimensionality and facilitate the performance of the high-level feature detectors which are presented in section 6.

5 Latent Semantic Analysis

Apart from the obvious next step of simply training classifiers using the aforementioned model vectors as the means of representing the extracted features of the given keyframe, we also perform some experiments using a Latent Semantic Analysis [23](LSA) approach. LSA is a technique in natural language processing, which exploits the relationships between a set of documents and the terms they contain more often by producing a set of concepts related to the documents and terms. More specifically:

Let $\mathbf{X}$ be a matrix whose (i, j) element describes the occurrence of *term* i in *document* j. In this approach, term corresponds to a region type of the region thesaurus and document to a given image. Thus, an image is considered to be a document whose words are the corresponding region types to its region.

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,i} & \dots & x_{m,n} \end{pmatrix} \tag{14}$$

where a line $t_i^T = (x_{i,1}, \dots, x_{i,n})$ denotes a vector that describes the relation of a term(region type) to every document(image). Apart from that, each column

of $\mathbf{X}$, denoted by $d_j = \begin{pmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{pmatrix}$ is a vector that corresponds to a single document(image) and describes its relation to every term(region type). This relation is the confidence of the existence of this region type to the given image. It is then obvious that d_j is the model vector of this image, as depicted in equation 13.

The dot product $t_i^T t_p$ between two term vectors gives their correlation and the matrix $\mathbf{X}\mathbf{X}^T$ contains all these dot products. It is obvious that $t_p^T t_i$. Apart from that, the matrix $\mathbf{X}^T\mathbf{X}$ contains all the dot products between the document vectors $d_j^T d_q$, describing their correlation over the terms. Assuming that there exists a decomposition of $\mathbf{X}$ described by $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{\Sigma}$ is diagonal. This decomposition is the well known Singular Value Decomposition (SVD) and is formally and analytically depicted in equation 15

$$\mathbf{X} = \begin{pmatrix} \begin{pmatrix} \\ \mathbf{u_1} \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{u_l} \\ \\ \end{pmatrix} \end{pmatrix} \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{pmatrix} \begin{pmatrix} (\mathbf{v_1}) \\ \vdots \\ (\mathbf{v_l}) \end{pmatrix} \tag{15}$$

where in equation σ_i denote the singular values and $\mathbf{u}_i$, $\mathbf{v}_i$ the left and right singular values, respectively.

Now, the k largest singular values may be kept from $\mathbf{\Sigma}$ along with their corresponding columns of $\mathbf{U}$ and rows of $\mathbf{V}$. Now, an approximation of $\mathbf{X}$ is formed and denoted in equation 16.

$$\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \tag{16}$$

This way, a translation of term and document vectors into a "concept" space is achieved. Now, let $\hat{t}_i$ and $\hat{d}_j$ denote the vectors that contain the occurrences of terms in concepts and relations between documents and concepts, respectively. The transformation of a document vector to the concept space is achieved using the matrix of the singular values $\mathbf{\Sigma}$ and the left singular value matrix $\mathbf{U}$, under the transform:

$$\hat{\mathbf{d}}_j = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{d} \tag{17}$$

In our approach as a further step, the model vectors of the training set are extracted with the use of the region thesaurus, as explained in the previous sections. Then, LSA is applied and matrices $\mathbf{\Sigma}$ and $\mathbf{U}$ are determined. After some experiments, the value of k is selected. Then each input model vector is driven to the concept space. This way the training and the test sets of the model vectors are transformed and used to train and test the high-level concept detectors.

Table 1. Accuracy for all 6 concepts.

Concept	35 Without LSA	With LSA
Fire	76.0%	84.0%
Rocks	69.5%	73.9%
Smoke	60.0%	64.0%
Snow	80.9%	90.5%
Trees	94.7%	89.4%
Water	65.3 %	72.0%

6 SVM Feature Detector Training

Support Vector Machines [24] are feed-forward networks that are frequently used for tasks such as pattern classification. Their main idea is to construct a hyperplane that acts as a decision space in such a way that the margin of separation between positive and negative examples is maximized. This hyperplane is not constructed in the input space, where the problem may not be linearly solvable, but in the feature space where the problem is driven. Despite the fact that a support vector machine does not incorporate domain-specific knowledge, it provides a good generalization performance.

For each semantic concept, a separate support vector machine is trained, thus solving a binary problem, of the existence or not of the concept in question. The input of the SVM is the model vector D_m described in section 4. The well known polynomial support vector machine, described in equation 18 is selected in our framework.

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^p \tag{18}$$

For the experiments of this work, the well-known LIBSVM library [25] has been used.

7 Experimental Results

For the evaluation of the presented framework a dataset[1] from various images collected from the world wide web. This set consists of approximately 600 images from the following semantic classes: *fire, rocks, smoke, snow, trees, water.* A separate detector was trained for each concept. Results are shown in table 1, before and after the application of Latent Semantic Analysis.

8 Conclusions - Future Work

The experimental results indicate that the extraction of the aforementioned low-level features is appropriate for semantic indexing. The selected concepts can be successfully detected when a given image is represented by a model vector that contains the distances to all the semantic entities of a constructed lexicon containing unlabeled semantic features. Plans for future work include the extraction of more visual features, exploitation of the spatial context of a keyframe and extension of this method for applications such as shot/image classification.

[1] Special thanks to Javier Molina for sharing his collection.

9 Acknowledgements

The work presented in this paper was partially supported by the European Commission under contracts FP6-027026 K-Space and FP6-027685 MESH. Evaggelos Spyrou is funded by PENED 2003 Project Ontomedia 03ED475.

References

1. Naphade, M.R., Kennedy, L., Kender, J.R., Chang, S.F., Smith, J.R., Over, P., Hauptmann, A.: A light scale concept ontology for multimedia understanding for trecvid 2005. (IBM Research Technical Report, 2005)
2. TREC: - video retrieval evaluation. (http://www-nlpir.nist.gov/projects/t01v/)
3. IBM: (Marvel: Multimedia analysis and retrieval system)
4. Souvannavong, F., Mérialdo, B., Huet, B.: Region-based video content indexing and retrieval. In: CBMI 2005, Fourth International Workshop on Content-Based Multimedia Indexing, June 21-23, 2005, Riga, Latvia. (2005)
5. Aksoy, S., Avci, A., Balcuk, E., Cavus, O., Duygulu, P., Karaman, Z., Kavak, P., Kaynak, C., Kucukayvaz, E., Ocalan, C., Yildiz, P.: Bilkent university at trecvid 2005. (2005)
6. Saux, B., G.Amato: Image classifiers for scene analysis. In: International Conference on Computer Vision and Graphics. (2004)
7. Voisine, N., Dasiopoulou, S., Mezaris, V., Spyrou, E., Athanasiadis, T., Kompatsiaris, I., Avrithis, Y., Strintzis, M.G.: Knowledge-assisted video analysis using a genetic algorithm. In: 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005). (April 13-15, 2005)
8. N. Boujemaa, F. Fleuret, V.G., Sahbi, H.: Visual content extraction for automatic semantic annotation of video news. In: IS&T/SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia, part of Electronic Imaging symposium. (2004)
9. Cees G.M. Snoek, Marcel Worring, D.C.K., Smeulders, A.W.: Learned lexicon-driven interactive video retrieval. (2006)
10. A. Natsev, M.N., Smith, J.: Lexicon design for semantic indexing in media databases. In: International Conference on Communication Technologies and Programming. (2003)
11. Lazebnik, S., Schmid, C., Ponce, J.: A discriminative framework for texture and object recognition using local image features. In: Towards category-level object recognition. Springer (2006) to appear.
12. O.Chapelle, P.Haffner, V.N.Vapnik: Support vector machines for histogram-based image classification. IEEE Transactions on Neural Networks **10**(5) (1999) 1055–1064
13. Snoek, C.G.M., Worring, M.: Time interval based modelling and classification of events in soccer video. In: Proceedings of the 9th Annual Conference of the advanced School for Computing and Imaging (ASCI). (2003)
14. Laaksonen, J., Koskela, M., Oja, E.: Picsom, self-organizing image retrieval with mpeg-7 content descriptors. (2002)
15. Chang, S.F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE trans. on Circuits and Systems for Video Technology **11**(6) (2001) 688–695
16. Spyrou, E., Koumoulos, G., Avrithis, Y., Kollias, S.: Using local region semantics for concept detection in video, 1st International Conference on Semantics And digital Media Technology (SAMT 2006), Athens, Greece (2006)

17. Avrithis, Y., Doulamis, A., Doulamis, N., Kollias, S.: A stochastic framework for optimal key frame extraction from mpeg video databases. (1999)
18. Morris, O.J., Lee, M.J., Constantinides, A.G.: Graph theory for image analysis: An approach based on the shortest spanning tree. (1986)
19. MPEG-7: Visual experimentation model (xm) version 10.0. ISO/IEC/ JTC1/SC29/WG11, Doc. N4062 (2001)
20. Manjunath, B., Ohm, J., Vasudevan, V., Yamada, A.: Color and texture descriptors. IEEE trans. on Circuits and Systems for Video Technology **11**(6) (2001) 703–715
21. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2 edn. Wiley Interscience (2000)
22. E.Spyrou, H.LeBorgne, T.Mailis, E.Cooke, Y.Avrithis, N.O'Connor: Fusing mpeg-7 visual descriptors for image classification. In: International Conference on Artificial Neural Networks (ICANN). (2005)
23. Deerwester, S., Dumais, S., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the Society for Information Science **41**(6) (1990) 391–407
24. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons (1998)
25. Chang, C.C., Lin, C.J.: Libsvm : a library for support vector machines (2001)