TÜBİTAK UZAY at TRECVID 2009: High-Level Feature Extraction and Content-Based Copy Detection

Ahmet Saracoğlu[1,2], Ersin Esen[1,2], Medeni Soysal[1,2], Tuğrul K. Ateş[1,2], Berker Loğoğlu[1], Mashar Tekin[1],
Talha Karadeniz[1], Müge Sevinç[1], Hakan Sevimli[1], Banu Oskay Acar[1], Ünal Zubari[1], Ezgi C. Ozan[1,2],
Egemen Özalp[1], Duygu Oskay Onur[1], Sezin Selçuk[1],
A. Aydın Alatan[2], Tolga Çiloğlu[2]
[1]TÜBİTAK Space Technologies Research Institute
[2]Department of Electrical and Electronics Engineering, M.E.T.U.
{ahmet.saracoglu, ersin.esen, medeni.soysal, tugrul.ates, berker.logoglu, mashar.tekin,
talha.karadeniz, muge.sevinc, hakan.sevimli, banu.oskay, unal.zubari, ezgican.ozan, duygu.oskay,
sezin.selcuk, egemen.ozalp}@uzay.tubitak.gov.tr
{alatan,ciloglu}@eee.metu.edu.tr

Abstract

In this notebook paper, we discuss and give an overview of our participation to the High-Level Feature Extraction (HLFE) and Content-Based Copy Detection (CBCD) tasks of TRECVID 2009. In our HLFE system both visual and audio concept detection has been implemented and also complimentary standalone detectors have been incorporated to the system. For the visual concept detection, a generalized visual feature extraction method based on codebook approach is employed. On the other hand, our audio system is a hierarchic system with continuous static spectral features and Gaussian Mixture Model classifiers. Furthermore, for the CBCD task our group has submitted to video-only, audio-only and audio + video subtasks. Our video-based CBCD system utilizes local interest points and bag-of-words concept and for the audio copy detection voting-based audio fingerprint matching method has been utilized.

1 High-Level Feature Extraction Task

Our HLFE system comprises four major building blocks; Aural Feature Extraction, Generalized Visual Feature Extraction, Gender Classification with Face Detection and Decision Fusion Modules. A basic block diagram of the system can be seen in the following figure. In addition, a dedicated detector for the *Hand* feature has been incorporated into the system. It should be noted that our HLFE system has been designed as a two class machine that is a decision about a shot is given as a feature is present in the shot or not and this classification is carried out for each feature of TRECVID 2009 high-level features.

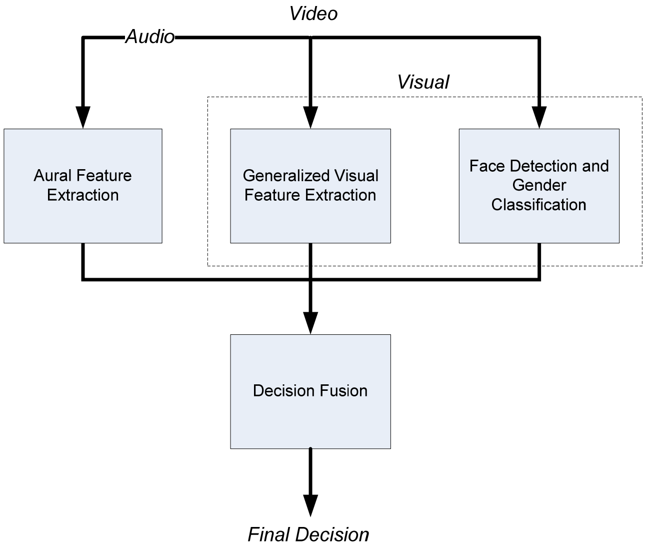
Video
Audio
Visual
Aural Feature Extraction
Generalized Visual Feature Extraction
Face Detection and Gender Classification
Decision Fusion
Final Decision

Figure 1: HLFE system design.

1.1 Generalized Visual Feature Extraction

Essentially, this sub-module presents the generic framework for the extraction of visual features such as crowd, *Cityscape*, *Classroom* and etc. System is based on the very-well known bag-of-words and codebook [1], [2], [3] approaches. Block-diagram of the system can be seen in the following figure.

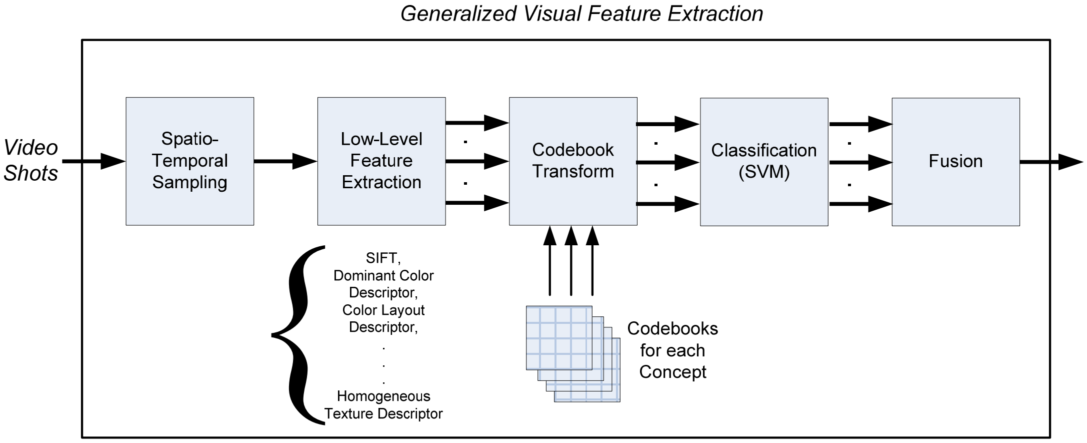
Generalized Visual Feature Extraction
Video Shots
Spatio-Temporal Sampling
Low-Level Feature Extraction
Codebook Transform
Classification (SVM)
Fusion
SIFT, Dominant Color Descriptor, Color Layout Descriptor, . . . Homogeneous Texture Descriptor
Codebooks for each Concept

Figure 2: Generalized Visual Feature Extraction block diagram.

At the first stage of the method, video shots are sampled in the temporal domain. Aforementioned sampling in our system has been performed by selecting five keyframes with equal intervals for the sake of decreasing the complexity of the system. Further sampling of keyframes are realized in the spatial domain in order to prepare keyframes for the low-level feature extraction stage.

The spatial sampling of the keyframes rely on three different grid structures by which keyframes are divided into non-overlapping regions. In our application grid structures have been selected as *1x1* in which full frame is processed, *2x2* which divides frames into four segments and *3x3*.

1.1.1 Low-Level Feature Extraction

In the feature extraction step visual descriptors of the keyframe regions are extracted by using 5 different methods. Four of the low-level feature extraction methods are selected from MPEG-7 descriptors [4]. These are Homogeneous Texture Descriptor (HTD), Edge Histogram Descriptor (EHD), Color Structure (CSD) and Color Layout Descriptor (CLD). And lastly Scale Invariant Feature Transform (SIFT) [5] has been selected as the fifth low-level feature extraction method. In the following paragraphs these methods are briefly described.

i. **Homogeneous Texture Descriptor:** captures the texture content of a given image by using the Gabor filters. Particularly Gabor filters can be seen as selective kernels in a given orientation and scale. The mean energy and deviation of energy are calculated for different sub-bands in the frequency domain.

ii. **Edge Histogram Descriptor:** A useful texture descriptor for image retrieval and similarity matching which represents the local and global edge distribution of an image [6]. The edge distribution of image is extracted by using predefined five types of edges (vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional).

iii. **Color Structure Descriptor:** A very compact and resolution-invariant representation of color, which captures both the color distribution and the spatial layout of the color. It can be used for wide variety of similarity based retrieval. It is also useful for spatial structure based applications.

iv. **Color Layout Descriptor:** A useful representation of an image by its color distribution and local spatial color structure. It gives additional information about the structure of the colors in the image, which cannot be obtained by the color histogram.

v. **Scale Invariant Feature Transform:** In this very well known method interest points are detected by utilizing the extremas of the Difference of Gaussians in different scales. And descriptor vector is computed by using the local gradients around the interest points.

Low-level visual features in our system are categorized as *global*, *semi-global* and *sparse*. Global features are extracted from the 1x1 grid that is the whole frame by using the MPEG-7 based descriptor methods. On the other hand, semi-global features are extracted from 2x2 and 3x3 grid elements by again using MPEG-7 based descriptors. And finally sparse features are the ones that are obtained by using only-SIFT on the whole frame.

1.1.2 Codebook Transform

At this stage of the system, all of the low-level features extracted from the given shot are transformed into a single feature vector/fingerprint thus the number of feature vectors is dramatically decreased and furthermore a compact representation of a shot is obtained. This transformation is carried out by using the codebook approach. In the transformation process, low-level feature vectors are first assigned to the nearest codeword in the codebook and afterwards distribution of codeword assignments is obtained as the final feature vector of the shot. Aforementioned codebooks are constructed by partitioning the visual feature space, which is achieved by employing k-Means clustering on the training set. The cluster centroids are associated with codewords, whose assembly constitutes the codebook. This said, aforementioned codebooks are constructed for each concept and on each low-level visual feature space, individually. That is to say, system contains for each high-level feature 57 different feature codebooks (4 global, 52 semi-global and 1 sparse). With this method, it should be noted that noisy feature vectors are eliminated more effectively thus robustness is increased.

1.1.3 Classification

For the classification of the codebook transformed feature vectors Support Vector Machine (SVM) classifier is utilized. In the learning stage, different kernels have been experimented and K-fold cross-validation is used for the training of classifiers. From our empirical analysis, Radial Basis Functions (RBF) has performed better compared to sigmoid and polynomial kernels. As it has been mentioned before, for each high-level feature and for each low-level feature domain SVM classifier is trained and utilized for classification.

1.1.4 Fusion

At this stage, results from bank of classifiers for a given high-level feature are combined in order to reach our final decision. Although there are a lot of methods in the literature for decision fusion, we have employed heuristic rules such as logical AND/OR operations and weighting scheme because of the simplicity of these methods.

1.2 Face Detection and Gender Classification

Block-diagram of the gender classification module used in our system can be seen in the following figure. Our approach is based on the method proposed by Moghaddam and Yang [7], [8] and [9] which utilizes SVM. In the first step of the method, a face detection algorithm is executed. For the face detection, very well known method of Viola and Jones [10] has been utilized.

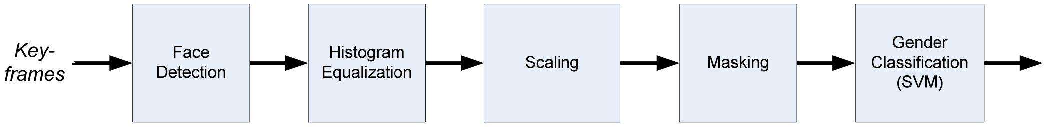

Figure 3: Gender classification block from face image diagram.

After the detection of face regions on the keyframe, histogram equalization is performed in order to reduce the lighting problems such as illumination variations. Afterwards, face regions are scaled to 24x24 dimensions in order to reduce the dimensionality of the data. And before the classification hair and remaining background regions are masked out and this masked image is used as the input to the classifier. Finally, SVM is employed for the classification of this two-class problem.

For the training and validation FERET [11] and [12] face database has been used which contains male and female face images taken from different angles. In our work, 940 female and 1600 male frontal images are used for training. Among the RBF, sigmoid and polynomial kernels RBF kernel performed best with 91.2% recall performance in our experiments.

1.3 Hand Detection

Human hands in images stand-out by three main properties, which are color, texture and shape. Utilizing hand color for detection can be achieved by employing a skin color detector. In [13] a high performance skin color detection method based on Gaussian mixture models in RGB color representation space is proposed. Furthermore, hand texture and shape can be recognized by region characteristics or frequency content [14].

Our approach to detect close up views of human hands in video content is focused on the skin response offered by [13] of the video. If total skin response of a shot is within a determined range, bare human parts are assumed to be in focus of the shot. Also, our observations showed that human parts other than hands and faces rarely appear in close-up views in common video footage. Also, wide-shot scenes which result in high skin response due to human appearance observed to contain a clear view of human heads. Thus, we apply two Viola-Jones face detectors [10], one for frontal and one for profile faces, in order to discriminate scenes that contain face and thus discriminate scenes that do not contain hand close-up hand(s).

To reduce false alarms from misclassification of skin pixels, we multiply the skin color response of frames with their motion responses calculated from the motion vectors. This allows us to eliminate static background in skin colors but results in misses from static hands shots, which are observed to be rare in the used dataset.

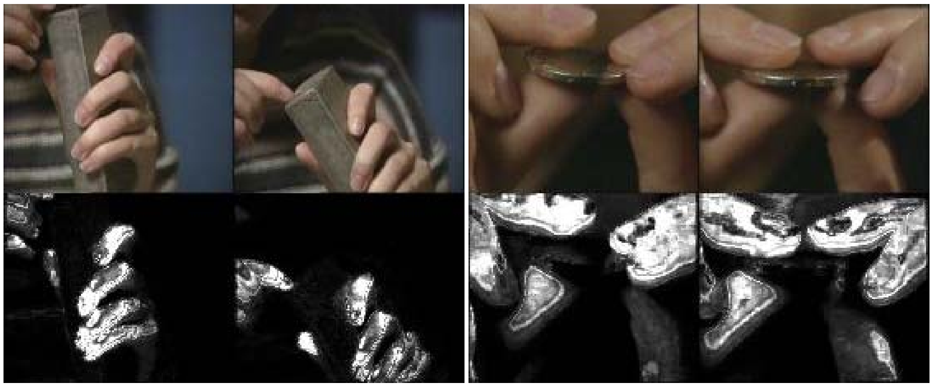

Figure 4: Example keyframes from the hand detection results, second row displays skin color responses.

1.4 Aural Feature Extraction

In TRECVID 2009 HLFE task, we have utilized aural feature extraction for two of the high level features, *singing* and *female face*, to be classified by audiovisual features. Due to the requirements on these features, audio-only detection is not applicable thus in general aural feature extraction is to reduce the false alarms caused by the generalized visual feature extraction and female face detection.

Audio event detection is performed in a hierarchical manner. The first step is silence and not-silence classification. In the next step, data classified as not silence is further processed. Afterwards, audio is classified as one of the four classes; *speech*, *music*, *singing* and *others*. And in the next level, speech is further categorized into female speech or male speech.

1.4.1 Silence Detection

Silence detection is performed by setting an energy threshold on single-second length windows. The threshold is determined by training a 2 mixture GMM with silence and not-silence data and using the frame energy as feature.

1.4.2 Speech, Music, Singing and Other Classification

Mel Frequency Cepstral Coefficients (MFCC) are known to be good features for Speech/Music classification [15]. In addition, delta MFCC's are also known as useful for singing detection [16]. Six MFCC and six delta-MFCC coefficients obtained from 0-3000Hz band are used in our system. And we have constructed the *other* group mostly from non-harmonic data. Since classes singing, speech and music are all harmonic, we selected the harmonicity as a discriminative feature between the *other* and speech, music, singing. The spectral entropy (SE) feature [17] used as an auxiliary descriptor for the existence of formant structure in the spectrum. The SE value is low for a flat spectrum and high for a curved spectrum. Finally, combining these features we end up with a 14 dimensional feature vector; 6 (MFCC) + 6 (delta MFCC) + 1 (Harmonicity) + 1 (SE). The features extracted every 10 millisecond on a 25 millisecond length window. For each class a 12 mixture GMM has been trained. The classification performed on non overlapping single-second length windows, each window containing 100 frames. And the class having the maximum count in 100 frames has been taken as the decision class.

1.4.3 Male and Female Classification

Male/Female voice classification performed on the data classified as speech. 12 dimensional Perceptual Linear Prediction (PLP) coefficients found to give the best result in our experiments. PLP's are extracted every 10ms on a 25ms window length. Two 12 mixture GMM trained, one for male and one for female voice. The classification performed on non overlapping 1 second length windows, each window containing 100 frames. And as before, the class having the maximum count in 100 frames has been taken as the decision class.

1.5 Discussion

This was the first participation of our team to the TRECVID 2009 initiative and HLFE task and although our performance has been subpar we had invaluable experiences and observations. First of all, the importance of the number of training samples has been apparent more than ever. For some of the high-level features, there were less than 50 annotations in our training data. Furthermore, the distribution of positive and negative samples in the training set poses an important problem. On the other hand, number of codewords in the codebooks should be higher than couple of hundred vectors as this was in our case.

2 Content-Based Copy Detection Task

2.1 Visual Copy Detection

Mainline approaches for content description for copy detection utilize global or local descriptors from video and comparing these descriptors for similarity. In the literature [19], it has been shown that local features perform better in terms of robustness on the other hand global features are computationally simpler. Local features for content description can be extracted around pixels returned by interest point detectors [20]. Thus, an interest point detector followed by a feature extractor is enough for describing most local aspects of a video scene.

Our approach to video CBCD is based on the clustering of SIFT descriptors and comparing video scenes by their memberships to these clusters. A codebook *C*, which holds the information for SIFT clusters, is created. SIFT descriptors obtained from luminance channels of sample videos are clustered with k-means algorithm and resulting cluster centers are stored in this codebook. Further extracted SIFT descriptors are assigned a code, which is the index of the nearest cluster center to the descriptor, from this codebook.

A reference database *R*, inside which queries will be searched, is created. This reference database holds the codes of every interest point from selected frames of reference videos. Query clips (*Q*), whose copies will searched inside the database, are summarized with the exact method of obtaining codes from interest points. Query codes are searched inside the reference database and matching reference video locations are voted for being a copy. Finally, top voted locations in reference video database are returned as results to copy detection system.

Reference database consists of *N* tables, C_1 to C_N, where N is the code count in *C*, which lists all the interest points with the corresponding code in reference videos. These tables hold several information about the interest points they summarize. Reference video ID, $id_{R,j}$ stores an integer id of the reference video the interest point is located. Interest point time, $t_{R,j}$ is the appearance time of the interest point with respect to the beginning of the video in seconds. A simple global characteristics of video frames are stored in $s_{R,j}$, is the number of total interest point encountered in along with the current interest point in its video frame and is a measure of complexity of the scene.

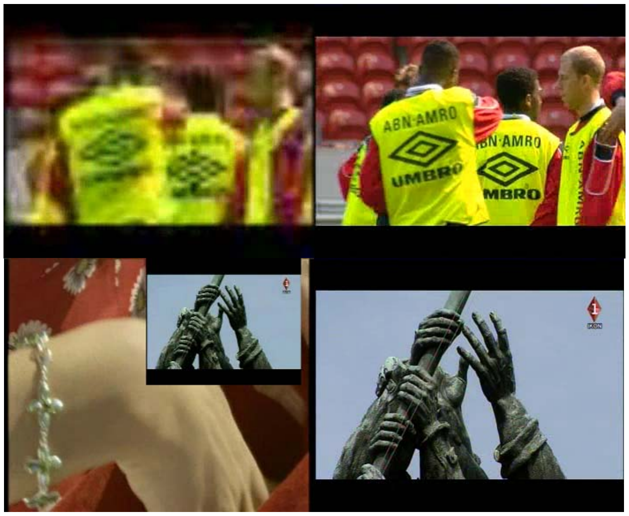

Figure 5: Example shots from the queries (on the left) and corresponding reference shots (on the right).

A query clip is represented with the codes of its interest points $Q_{t,i}$, where ***t*** is the temporal location in query clip in seconds and *i* is index of the interest point within this location. Frame complexity, $s_{Q,i}$ is also utilized similarly. Total query duration is represented by d_Q.

When a query is asked to the system, all interest points from selected frames in the query is extracted and their corresponding codebook groups are found. A vote table *V* is initialized and filled with the votes of code matches. For every code in *Q*, corresponding table in reference database is found and a vote is calculated and added to the vote table for every interest point in reference table as follows:

$$V_{id_{R,j}} += \frac{\min(s_{R,j}, s_{Q,i}) / \max(s_{R,j}, s_{Q,i})}{d_Q} \qquad (1)$$

This equation favors reference frames with similar complexity as the query frame in search. It also scales the global score for current query according to the query time, thus enabling the use of a universal threshold for this copy detection system. When the voting step is over, results are returned from the reference locations with the most scores. These results can later be eliminated by their scores to suit the needs of the application.

According to the video-only CBCD run results of TRECVID 2009, current scheme of our copy detection method is observed respond to most attacks including quality decrease and picture in picture. However due to the utilized local descriptor, SIFT; the system is prone to flip attacks. The problem can be overcome by searching flipped versions of queries in the reference database or local converting descriptors to represent actual flipped images. Results show that the system achieves much better precision compared to our efforts in [18] and recall rate shows a minor increase over the last year. Finally as a future venture, augmentation of local and *global* descriptors from chrominance channels of videos is expected make an increase in terms of retrieval performance.

2.2 Audio Copy Detection

Our audio copy detection method has been realized from the work in [18]. In this section we summarize the method. In the first step of our audio copy detection method *fingerprints/hash values* are extracted. The fingerprints are extracted in the form of 15 bits. The hash values are calculated using the power spectra of 25 ms frames separated by 10ms. Each signal frame is multiplied by Hamming window before its Fourier Transform is computed. The spectrum over 300Hz – 3000Hz is divided into 16 sub-bands according to the Bark scale. The energy differences between these sub-bands are used to calculate the hash values according to (2).

$$F(n,m) = \begin{cases} 1, & EB(n,m) - EB(n,m+1) > 0 \\ 0, & EB(n,m) - EB(n,m+1) \leq 0 \end{cases} \qquad (2)$$

In (2), *EB(n, m)* represents the energy value of the n^{th} frame at the m^{th} sub-band. In the following figure, a detailed diagram of audio fingerprint extraction method is provided.

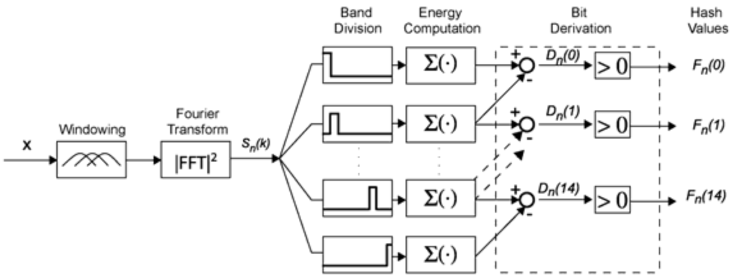

Figure 6: Fingerprint extraction method.

And for the audio fingerprint matching, a hash database containing all possible hash values is created to reach out quickly to the exact match points. This hash database contains 2^{15} different hash values, each holding linked lists, pointing to the locations of these hash values in the reference audio files inserted to the database. The hash values of the query are matched with the hash values of the reference data using the previously formed hash database without any sequence scan. For every query file, a voting table is created. This voting table holds a vote that is calculated by counting the number of the equal time differences between the matching points of query and reference data. For example, the 3^{rd} and 5^{th} hash values of the query, matches exactly with the 10^{th} and 12^{th} hash values of the reference file. So the voting table holds a value of 2 for the difference 7. Then, if there are more exact matches with the same difference of 7, this value of 2 is increased. So, the sequential exact match points are searched within the reference data to locate the query. The voting table also holds the first and last time indices of the corresponding difference value. This shows where the query data located within the reference file. The voting function, *V* that calculates the value obtained for the time differences between the query and the reference file is given in the following equation

$$V(\tau) = \sum_{\{r,q\}\epsilon R} \delta(\tau - |r - q|)$$

In (3), q and r show the time indices of the matching locations of the query and reference fingerprints whereas, τ is the difference between the time indices. The similarity for every difference value τ is calculated by dividing $V(\tau)$ by the difference of the first and last time index of the corresponding difference in seconds. The point with the highest similarity gives the most similar area for the reference and query data. In other words, similarity is calculated as the number of exact matches per second.

2.3 Fusion

At the decision fusion stage for the audio-video runs, individual matching results obtained from previously explained audio and video processing stages are combined. Combination rule is to choose the best matching result in terms of confidences obtained from separate audio and visual content matching. For each query a single best matching temporal segment from the reference database is returned, if the resultant confidence value exceeds a certain threshold.

3 References

[1] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. IEEE Trans. PAMI, 28(10):1678–1689, 2006.

[2] S. Chang, J. He, Y. Jiang, A. Yanagawa, and E. Zavesky, E. El Khoury, C. Ngo, "Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search," in TRECVID Proceedings, 2008.

[3] C. G. M. Snoek, K. E. A. van deSande, O. de Rooij, B. Huurnink, J. C. van Gemert, J.R.R. Uijlings, J. He, X. Li, I. Everts, V. Nedovic, M. van Liempt, R. van Balen, F. Yan, M.A. Tahir, K. Mikolajczyk , J. Kittler, T. Gevers, D. C. Koelma, A. W. M. Smeulders, and M.Worring. The MediaMill TRECVID2008 semantic video search engine, 2008.

[4] B. S. Manjunath, Philippe Salembier, Thomas Sikora, "Introduction to MPEG-7 Multimedia Content Description Interface".

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int'l J. Computer Vision, Volume 60, Number 2, pp. 91–110, Nov. 2004.

[6] Park, D. K., Jeon, Y. S., and Won, C. S. 2000. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM Workshops on Multimedia*, New York, NY, 51-54.

[7] Ming-Hsuan Yang; Moghaddam, B., "Gender classification using support vector machines," Image Processing, 2000. Proceedings. 2000 International Conference on , vol.2, no., pp.471-474 vol.2, 2000.

[8] Gregory Shakhnarovich, Paul A. Viola, Baback Moghaddam, "A Unified Learning Framework for Real Time Face Detection and Classification," Automatic Face and Gesture Recognition, IEEE International Conference on, pp. 0016, Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FG'02), 2002.

[9] B. Moghaddam, M.H. Yang, "Learning Gender with Support Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 707-711, May, 2002.

[10] Paul Viola, Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, pp. 511, 2001.

[11] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss,"The FERET database and evaluation procedure for face recognition algorithms," Image and Vision Computing J, Vol. 16, No. 5, pp. 295-306, 1998.

[12] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, pp. 1090-1104, 2000.

[13] M.J. Jones and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1999.

[14] M. Kölsch and M. Turk, "Robust Hand Detection," Proc. IEEE International Conference on Automatic Face and Gesture Recognition, IEEE CS Press, 2004, pp. 614–619.

[15] Logan B., "Mel Frequency Cepstral Coefficients for Music Modeling", Proceeding of the International Symposium on Music Information Retrieval (ISMIR) 2000, Plymouth, USA, October 2000.

[16] Wu Chou and Liang Gu, "Robust Singing Detection in Speech/Music Discriminator Design," International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), pp.865-868, Salt Lake City, Utah, USA, May 2001.

[17] A. M. Toh, R. Togneri, and S. Nordholm, "Spectral entropy as speech features for speech recognition," in Proceedings of PEECS, 2005, pp. 22–25.

[18] Saracoğlu, A.; Esen, E.; Ateş, T.K.; Oskay Acar, B.; Zubari, Ü.; Özalp, E.; Alatan, A.A.; Çiloğlu, T., "Content based copy detection with coarse audio-visual fingerprints," Seventh International Workshop on Content-Based Multimedia Indexing, CBMI 2009, pp.213-218, 3-5 June 2009, Chania, Crete, Greece.

[19] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, F. Stentiford, "Video copy detection: A comparative study," in ACM International Conference on Image and Video Retrieval (CIVR'07), July 9–11, 2007, Amsterdam, The Netherlands, pp. 371–378.

[20] A. Joly, O. Buisson,C. Frelicot. "Content-based Copy Retrieval Using Distortion-based Probabilistic Similarity Search," IEEE Trans. on MM, vol. 9, no. 2, Feb. 2007.