A NOVEL APPROACH FOR CONTENT BASED VIDEO RETRIEVAL

Mr. Prashant B. Jawade
Department of Information Technology
Government College Of Engineering,
Karad, Maharashtra, India.
prashantjawade1234@gmail.com

Miss. Sneha D. Patil
Department of Information Technology
Government College Of Engineering,
Karad, Maharashtra, India.
sneha.patil13sp@gmail.com

Mr. Ganesh S. Gutte
Department of Information Technology
Government College Of Engineering,
Karad, Maharashtra, India.
ganeshgutte777@gmail.com

Abstract— **Content based video retrieval is an approach for browsing videos over the World Wide Web using an image or a video clip as an input instead of semantic information. Video contains several types of audio and visual information which are difficult to extract, combine or trade-off in common video information retrieval. Therefore, in this paper we propose a scheme for retrieving videos basically by face detection and feature extraction. The video retrieval system includes various steps: Video Framing, Face Detection, Face Recognition, SURF Feature Extraction, image matching is done using extracted facial features and finally providing the video of the respective person to the end user.**

Keywords- Video Framing, Face Detection, SURF Feature Extraction, CBVR, etc.

I. INTRODUCTION

There is amazing growth in the amount of digital video data in recent years. Inexpensive storage, ubiquitous broadband Internet access, low cost digital cameras, and nimble video editing tools result in a flood of unorganized video content. Most of the multimedia search systems rely on available metadata or contextual information in text form. Even if there is a full textual description of the content available, it often cannot be found directly using conventional text queries. Video contains several types of audio and visual information which are difficult to extract. [4]A video may have an auditory channel as well as a visual channel. The available information from videos includes the following: 1) video metadata, which are tagged texts embedded in videos, usually including title, summary, date, actors, producer, broadcast duration, file size, video format, copyright, etc.; 2) audio information from the auditory channel; 3) transcripts: Speech transcripts can be obtained by speech recognition and caption texts can be read using optical character recognition techniques; 4) visual information contained in the images themselves from the visual channel. If the video is included in a webpage, there are usually webpage texts associated with the video. Content-based video retrieval (CBVR), also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) is the application of computer vision to the video retrieval problem, that is, the problem of searching for video in large databases. "Content-based" means that the search will analyze the actual content of the video. The retrieval is based on the content of the video object. [2] "Content" in this context might refer to colors, shapes, textures or any other information that can be derived from the image itself. Without the ability to examine video content, searches must rely on metadata such as captions or keywords, which may be laborious or expensive to produce. Modeling of semantic information with ontologies for managing and querying data enables a targeted search for content. The occurrences of different faces provide rich information for browsing, navigating and retrieval of huge amount of video/image data. Current state-of-the-art face detectors that can reliably and quickly detect frontal faces with different sizes and locations in complex background images can be used to extract faces from videos. In this paper we present a Content Based Video Retrieval (CBVR) System which includes various steps: (1) Video Framing: - Videos are converted into frames, also called as still images. (2) Face Detection: - Faces of persons in the video are detected using Viola-Jones face-detection algorithm. (3) Feature Extraction: - SURF features are extracted and stored for the detected faces in the video frames. (4) Similarity Matching: - The extracted features of the input image are compared with the stored extracted features of the persons. For retrieving the video from database, the retrieval subsystem processes the presented query, performs similarity matching operations and finally displays the result to end user. Content-based video retrieval have a wide range of applications such as quick browsing of video folders, analysis of visual electronic commerce (such as analysis of interest trends of user's elections and orderings, analysis of correlations between advertisements and their effects), remote instruction, digital museums, news event analysis, intelligent management of web videos (useful video search and harmful video tracing) and video surveillance. Video retrieval continues to be one of the most exciting and fastest growing research areas in the field of multimedia technology.

II. RELATED WORK

Quite significant work has been done with regard to content based video retrieval systems. Similarity matching technique is used in [1], taking image as a query. Feature extraction and indexing is used in the CBVR system in [2]. In [3], we find work done on video indexing and video retrieval. Ample amount of efforts have been taken in [4] for surveying on visual content based video indexing and retrieval. [5] Deals with local binary patterns and TRECVID, an efficient method

978-1-4673-8855-9/17/$31.00 ©2017 IEEE

for face retrieval from large datasets. [6] Focuses on CBVR system with face recognition and retrieval using face detection and face tracking techniques. Semantic web-based methods and video annotation techniques are used in [7]. Good work is done on Content-based Video Retrieval and Summarization using MPEG-7 in [8]. A detailed literature survey on Face Recognition is provided in [9]. [10] Provides an approach for facial feature extraction and verification for Omni-face detection in videos. [11] Provides an automated CBVR system. Video indexing technique is used for face detection and verification in [12]. In [13] we get a brief description of all the techniques and systems for image as well as video retrieval.

We propose a novel idea of creating a framework for multimedia content extraction and retrieval using facial feature extraction. Using Viola-Jones face detection algorithm we firstly detect the faces in the video frames. The detected face of the person gets cropped and image enhancement is done for the same. We found that SURF Feature Extraction for the enhanced image provides more accuracy for comparison. In our paper we present a generic ontology which mainly concentrates on the representation of extracted features of a video.

III. PROPOSED SYSTEM

Content Based Video Retrieval (CBVR) is a multimedia retrieval technique used basically for retrieving videos, images, etc using contents of the media rather than any textual information associated with it. The proposed system is a CBVR system based on face detection and SURF feature extraction. Proposed Video Storage and Retrieval System, stores and manages a large number of video data and allows users to retrieve videos from the database efficiently. Proposed System provides different functionality for two main clients-which are Administrator and User. Administrator is responsible for controlling the entire database including security and adding, updating and deleting videos to and from database. User can only retrieve videos based on submitted query based on content as well on metadata. Fig. 1 gives the basic block diagram of the proposed system.

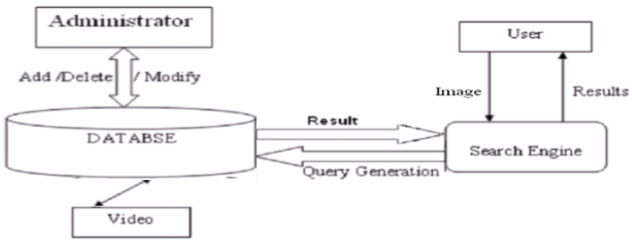

Fig. 1 Block Diagram of the proposed system

The two main clients are Administrator and End User. Administrator is responsible for maintaining the database. Database consists of a collection of large number of video datasets. We have focused on the domain of video lectures. Thus here we consider a video dataset to be a collection of video lectures of various people. Administrator will have an authority to add videos to the database delete videos from the database and perform necessary modifications to the database. For this purpose we have created a standalone application for Administrator which can be accessed only by an authorized user, i.e., Administrator. Another client is the End User of the system. User will provide an image of a person to the system in order to get all of the videos of that particular person. This image is referred as a query to the system. In Fig. 1, the end user provides an image input to the system named as "Search Engine" which does query generation for finding output. For the end user, we have created another standalone application which is the basic interface to the user for interaction with the system.

Initially database will be consisting of video dataset having video lectures of various people. For storing a video into the database, administrator will provide name of the video and video location where it is to be stored. The respective video will be stored at that particular location. At the same time, video will be converted into frames (still images). Key frames will be selected from these set of frames. Key-frames are still images extracted from original video data that best represent the content of shots in an abstract manner. Once key frames are extracted next step is to detect faces of people in the frames. For face detection we have used Viola Jones face detection algorithm. After the faces have been detected, the detected face is cropped for further process. Image enhancement is done for this cropped image. We have converted the cropped image into a grayscale image. Further we do SURF feature extraction for the cropped grayscale image. Thus we store these extracted facial features of every person in the database.

In the similar manner, same procedure is performed on the input image provided by the user. Face detection method is used to perform face detection on the input image and the detected face is being cropped. The cropped face image is converted into grayscale image and feature extraction is done for the same. Now these extracted facial features of the input image will be compared with the stored facial features of all the persons in the database. When we get the person with same facial features, all the videos having that particular person will be retrieved and provided as output to the end user.

Fig. 2 gives the internal logic of the system.

A. Video Framing

A video file consists of frames. These frames when appear before us in a rate more than our perception of vision, gives a sensation of an object moving before us, by looking just at the screen on which frames are appearing at high rate. Thus one can say that frames are the fundamental entity of a video file. A frame is an electronically coded still image in video technology. When we shoot video, we are actually taking pictures – 30 times per second. In video parlance, we shoot at 30 frames per second (fps). There are other frame rates that are used as well. Films have generally been shot at 24 fps.

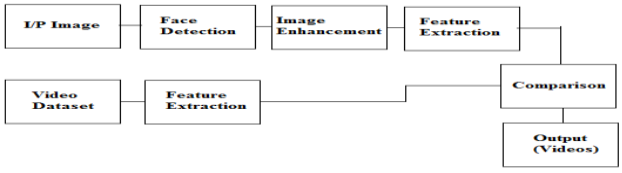

Fig. 2 Internal Logic of the system

We firstly convert videos into frames. As we got a frame rate of 20 frames per second (fps) we extracted key-frames from them. A key frame in animation and filmmaking is a drawing that defines the starting and ending points of any smooth transition. The drawings are called "frames" because their position in time is measured in frames on a strip of film. A sequence of key frames defines which movement the viewer will see, whereas the position of the key frames on the film, video, or animation defines the timing of the movement. Because only two or three key frames over the span of a second do not create the illusion of movement, the remaining frames are filled with in-betweens. Key-frames have been frequently used to supplement the text of a video log, though they were selected manually in the past. Key-frames, if extracted properly, are a very effective visual abstract of video contents and are very useful for fast video browsing. Once you extract the correct key frames the further process gets easy and amount of processing gets drastically reduced increasing efficiency.

B. Face Detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process. Face Detection consists of two types: (1) feature-based method; and (2) classification-based method. The feature based methods search for different facial features and use their spatial relationship to locate faces. The classification-based methods detect faces by classifying all possible sub-images of a given image as face or non-face sun-images [10]. Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

There are many face detection algorithms to locate a human face in a scene – easier and harder ones. Some of them are: Viola Jones Face Detection, Real-Time Face Detection using Edge-Orientation Matching, Robust Face Detection using the Hausdorff Distance, etc. We have used Viola Jones face detection algorithm in our proposed model. The real-time face detection scheme proposed by Viola and Jones is arguably the most commonly employed front face detector, which consists of a cascade of classifiers trained by AdaBoost employing Harr-wavelet features. AdaBoost is one of the most successful machine learning techniques applied in computer vision, which provides a simple yet effective approach for stagewise learning of a nonlinear classification function. Later their approach was extended with rotated Harr like features and different boosting algorithms [6].

Fig. 3 gives a configuration of a generic face identification and verification system [9]. For identification of a person we first do face detection and then feature extraction for the detected face. These extracted features help in person identification/verification. In our system, it is used for identifying videos of the person provided in input image.

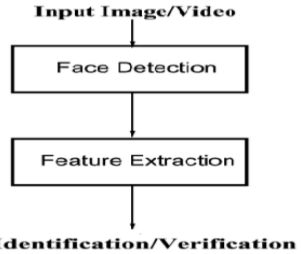

Fig. 3 Configuration of a generic face identification and verification system

C. Face Detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate The principal objective of enhancement is to process a given image so that the result is more suitable than the original image for a specific application. In Image Processing image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis. For example, you can remove noise, sharpen, or brighten an image, making it easier to identify key features. Here are some useful examples and methods of image enhancement: Filtering with morphological operators, histogram equalization, noise removal using a Wiener filter, linear contrast adjustment, median filtering, unsharp mask filtering, contrast-limited adaptive histogram equalization (CLAHE), de-correlation stretch.

In image enhancement we have performed two tasks: (1) image cropping; and (2) conversion to grayscale image. Our first step was face detection using Viola Jones algorithm. After the face is being detected in the image, the detected face is being cropped in the first step of image enhancement. Here we have cropped the image because removing the unnecessary background and keeping only face image helps in better feature extraction, thus improving accuracy. Then in the second step of image enhancement the cropped face image is converted into a grayscale image. As we have used SURF feature extraction, it requires grayscale image for better feature detection and extraction. A grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only two colors, black and white (also called bi-level or binary images).

IV. ALGORITHM DESCRIPTION

The algorithms that we have used in our system are as follows: - Viola Jones algorithm (Face Detection), SURF Feature Extraction (Feature Extraction).

A. Viola Jones Face Detection Algorithm

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. There are many face detection algorithms to locate a human face in a scene – easier and harder ones. Some of them are: Viola Jones Face Detection, Real-Time Face Detection using Edge-Orientation Matching, Robust Face Detection using the Hausdorff Distance, etc. We have used Viola Jones face detection algorithm in our proposed model.

The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. The problem to be solved is detection of faces in an image. A human can do this easily, but a computer needs precise instructions and constraints. To make the task more manageable, Viola–Jones requires full view frontal upright faces. Thus in order to be detected, the entire face must point towards the camera and should not be tilted to either side. While it seems these constraints could diminish the algorithm's utility somewhat, because the detection step is most often followed by a recognition step, in practice these limits on pose are quite acceptable. The algorithm has four stages.

1) Haar Feature Selection
2) Creating an Integral Image
3) Adaboost Training
4) Cascading Classifiers.

1) Haar Features

All human faces share some similar properties. These regularities may be matched using Haar Features. A few properties common to human faces:

a) The eye region is darker than the upper-cheeks.

b) The nose bridge region is brighter than the eyes.

Composition of properties forming matchable facial features:

a) Location and size: eyes, mouth, bridge of nose

b) Value: oriented gradients of pixel intensities

The four features matched by this algorithm are then sought in the image of a face (shown at left).

Rectangle features:

c) Value = Σ (pixels in black area) - Σ (pixels in white area)

d) Three types: two-, three-, four-rectangles, Viola & Jones used two-rectangle features

e) For example: the difference in brightness between the white &black rectangles over a specific area

f) Each feature is related to a special location in the sub-window

2) An image representation called the integral image evaluates rectangular features in constant time, which gives them a considerable speed advantage over more sophisticated alternative features. Because each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in nine.

3) Learning algorithm

The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there are a total of M=162,336 possible features, and it would be prohibitively expensive to evaluate them all when testing an image. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them. This algorithm constructs a "strong" classifier as a linear combination of weighted simple "weak" classifiers.

$$h(x) = sign\left(\sum_{j=1}^{M}\left(a_j h_j(x)\right)\right)$$

Each weak classifier is a threshold function based on the feature

$$h_j(x) = f(x) = \begin{cases} -s_j, & if\ f_j < \theta_j \\ s_j, & otherwise \end{cases}$$

The threshold value and the polarity $\epsilon \pm 1$ are determined in the training, as well as the coefficients .

Here a simplified version of the learning algorithm is reported:

Input: Set of ***N*** positive and negative training images with their labels. If image ***i*** is a face. If not

1. Initialization: assign a weight to each image ***i.***
2. For each feature with ***j=1 , … , M***

1. Renormalize the weights such that they sum to one.

2. Apply the feature to each image in the training set, then find the optimal threshold and polarity , that minimizes the weighted classification error. That is

$$\theta_j, s_j = \arg\min_{\theta,s} \sum_{i=1}^{N} w_j^i e_j^i \text{ where } e_j^i = \begin{cases} 0, if\ y^i = h_j(x^i, \theta_j, s_j) \\ 1, otherwise \end{cases}$$

3. Assign a weight to that is inversely proportional to the error rate. In this way best classifiers are considered more.

4. The weights for the next iteration, i.e. , are reduced for the images ***i*** that were correctly classified.

3. Set the final classifier to

$$h(x) = sign\left(\sum_{j=1}^{M}\left(a_j h_j(x)\right)\right)$$

4) Cascade architecture

- On average only 0.01% of all sub-windows are positive (faces)

- Equal computation time is spent on all sub-windows
- Must spend most time only on potentially positive sub-windows.
- A simple 2-feature classifier can achieve almost 100% detection rate with 50% FP rate.
- That classifier can act as a 1st layer of a series to filter out most negative windows
- 2nd layer with 10 features can tackle "harder" negative-windows which survived the 1st layer, and so on…

- A cascade of gradually more complex classifiers achieves even better detection rates. The evaluation of the strong classifiers generated by the learning process can be done quickly, but it isn't fast enough to run in real-time. For this reason, the strong classifiers are arranged in a cascade in order of complexity, where each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continue on searching the next sub-window. The cascade therefore has the form of a degenerate tree. In the case of faces, the first classifier in the cascade – called the attentional operator – uses only two features to achieve a false negative rate of approximately 0% and a false positive rate of 40%.[6] The effect of this single classifier is to reduce by roughly half the number of times the entire cascade is evaluated.

In cascading, each stage consists of a strong classifier. So all the features are grouped into several stages where each stage has certain number of features.

The job of each stage is to determine whether a given sub-window is definitely not a face or may be a face. A given sub-window is immediately discarded as not a face if it fails in any of the stages.

A simple framework for cascade training is given below: -

F(0) = 1.0; D(0) = 1.0; i = 0
while F(i) > Ftarget
i++
n(i) = 0; F(i)= F(i-1)
while F(I) > f x F(i-1)
n(i) ++
use P and N to train a classifier with n(I) features
Evaluate current cascaded classifier on validation set to determine F(i) & D(i)
decrease threshold for the ith classifier
until the current cascaded classifier has a detection rate of at least d x D(i-1) (this also affects F(i))
N = ∅
if F(i) > Ftarget **then**
evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N.

The cascade architecture has interesting implications for the performance of the individual classifiers. Because the activation of each classifier depends entirely on the behavior of its predecessor, the false positive rate for an entire cascade is:

$$F = \prod_{i=1}^{k} f_i$$

Similarly, the detection rate is:

$$D = \prod_{i=1}^{k} d_i$$

B. SURF Feature Extraction

Feature detection is the process where we automatically examine an image to extract features, that are unique to the objects in the image, in such a manner that we are able to detect an object based on its features in different images. Speeded up robust features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image.

The SURF algorithm is based on the same principles and steps as SIFT; but details in each step are different. The algorithm has three main parts: interest point detection, local neighborhood description and matching.

1) Detection

SURF uses square-shaped filters as an approximation of Gaussian smoothing. (The SIFT approach uses cascaded filters to detect scale-invariant characteristic points, where the difference of Gaussians (DoG) is calculated on rescaled images progressively.) Filtering the image with a square is much faster if the integral image is used:

$$S(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j)$$

The sum of the original image within a rectangle can be evaluated quickly using the integral image, requiring evaluations at the rectangle's four corners.

SURF uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix is used as a measure of local change around the point and points are chosen where this determinant is maximal. In contrast to the Hessian-Laplacian detector by Mikolajczyk and Schmid, SURF also uses the determinant of the Hessian for selecting

the scale, as is also done by Lindeberg. Given a point p=(x, y) in an image I, the Hessian matrix H(p, σ) at point p and scale σ, is:

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix}$$

where $L_{xx}(p,\sigma)$ etc. are the second-order derivatives of the

grayscale image.

The box filter of size 9×9 is an approximation of a Gaussian with σ=1.2 and represents the lowest level (highest spatial resolution) for blob-response maps.

C. Scale-space representation and location of points of interest: -

Interest points can be found at different scales, partly because the search for correspondences often requires comparison images where they are seen at different scales. In other feature detection algorithms, the scale space is usually realized as an image pyramid. Images are repeatedly smoothed with a Gaussian filter, then they are subsampled to get the next higher level of the pyramid. Therefore, several floors or stairs with various measures of the masks are calculated:

$$\sigma_{approx} = Current\ filter\ size * \left(\frac{Base\ Filter\ Scale}{Base\ Filter\ Size}\right)$$

The scale space is divided into a number of octaves, where an octave refers to a series of response maps of covering a doubling of scale. In SURF, the lowest level of the scale space is obtained from the output of the 9×9 filters.

Hence, unlike previous methods, scale spaces in SURF are implemented by applying box filters of different sizes. Accordingly, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size. The output of the above 9×9 filter is considered as the initial scale layer at scale s=1.2 (corresponding to Gaussian derivatives with σ=1.2). The following layers are obtained by filtering the image with gradually bigger masks, taking into account the discrete nature of integral images and the specific filter structure. This results in filters of size 9×9, 15×15, 21×21, 27×27,.... Non-maximum suppression in a 3×3×3 neighborhood is applied to localize interest points in the image and over scales. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Brown, et al. Scale space interpolation is especially important in this case, as the difference in scale between the first layers of every octave is relatively large.

D. Descriptor

The goal of a descriptor is to provide a unique and robust description of an image feature, e.g., by describing the intensity distribution of the pixels within the neighbourhood of the point of interest. Most descriptors are thus computed in a local manner, hence a description is obtained for every point of interest identified previously.

The dimensionality of the descriptor has direct impact on both its computational complexity and point-matching robustness/accuracy. A short descriptor may be more robust against appearance variations, but may not offer sufficient discrimination and thus give too many false positives.

The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then we construct a square region aligned to the selected orientation, and extract the SURF descriptor from it.

E. Orientation assignment

In order to achieve rotational invariance, the orientation of the point of interest needs to be found. The Haar wavelet responses in both x- and y-directions within a circular neighbourhood of radius 6 s {\displaystyle 6s} 6s around the point of interest are computed, where s {\displaystyle s} 8 is the scale at which the point of interest was detected. The obtained responses are weighted by a Gaussian function centered at the point of interest, then plotted as points in a two-dimensional space, with the horizontal response in the abscissa and the vertical response in the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of size π/3. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector. The longest such vector overall defines the orientation of the point of interest. The size of the sliding window is a parameter that has to be chosen carefully to achieve a desired balance between robustness and angular resolution.

F. Descriptor based on the sum of Haar wavelet responses

To describe the region around the point, a square region is extracted, centered on the interest point and oriented along the orientation as selected above. The size of this window is 20s.
The interest region is split into smaller 4x4 square sub-regions, and for each one, the Haar wavelet responses are extracted at 5x5 regularly spaced sample points. The responses are weighted with a Gaussian (to offer more robustness for deformations, noise and translation).

G. Matching

By comparing the descriptors obtained from different images, matching pairs can be found.

V. EXPERIMENTAL RESULTS

In this section, a set of experimental results are demonstrated to verify the effectiveness and efficiency of the proposed strategy.

We used the video datasets consisting of video lectures of different people. These datasets have about 9 hours of video broadcasts on diff erent topics such as English, Quantitative Aptitude, General Knowledge, Technical, Non-Technical, History, Geography, Personality Development, etc. The total number of frames that we processed was about 6,25,200 with 10 face tracks. For result analysis we have considered 10 persons from the database.

Frame rate as per the frame extraction algorithm that we have used is approximately 20 fps (frames per second).

In SURF feature extraction we have counted the number of features extracted and the total number of features matched with the query image.

Following table shows the list of images (frames) of people with features stored in database and the total number of features matched with the query image: -

PEOPLE WITH FEATURES STORED IN DATABASE	NUMBER OF MATCHED FEATURES
1.jpg (same image)	434
11.jpg (same person's different image)	18
2.jpg	8
3.jpg	5
4.jpg	4
5.jpg	5
6.jpg	3
7.jpg	2
8.jpg	2
9.jpg	1
10.jpg	4

TABLE I. NUMBER OF MATCHED FEATURES

Here we have taken 1.jpg as the query image. It is successively compared with the images 1.jpg, i.e. the same image from the frames extracted. As both the images are same, the numbers of features matched are also high, viz. 434 features. Then the query image is compared with the image 11.jpg which is a different image of the same person in the query image. Though the images to be compared are different, both of them have same person hence the number of features matched are comparatively high, i.e. 18 features. When the query image is compared with rest of the images of different people, the numbers of features matched are low.

For comparison of the number of features matched, we have taken criteria of more than 90% of feature matching. That is, if more than 90% of the features of a person stored in database are matched with the features of query image then the videos of that particular person are retrieved as output.

Time Analysis for similarity matching is done based on the amount of time required for retrieving the video from the database and the percentage match of the query video with video files stored in database.

Following table shows the video length and retrieved time: -

QUERY VIDEO	VIDEO DURATION (sec)	RETRIEVE TIME (sec)
1.mp4	340 sec	6 sec
2.mp4	14 sec	2 sec
3.mp4	20 sec	3 sec
4.mp4	662 sec	10 sec
5.mp4	21 sec	4 sec
6.mp4	600 sec	10 sec
7.mp4	442 sec	7 sec
8.mp4	720 sec	11 sec
9.mp4	332 sec	5 sec
10.mp4	431 sec	7 sec

TABLE II. TABLE TYPE STYLES

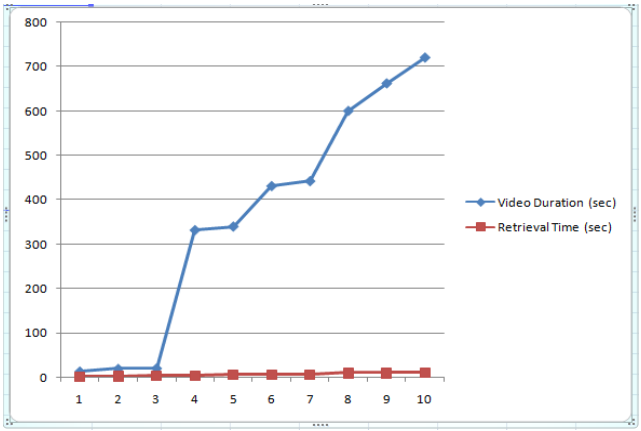

Fig. 4 Video Duration versus Retrieval Time

The graph represents plotting of video duration versus video retrieval time in seconds. The retrieval time increases very slightly with increasing video duration. For 14 seconds video we got retrieval time 2 seconds. For 720 seconds video we got retrieval time 11 seconds. Thus, though we are dealing with videos of large duration we get minimal retrieve time with the usage of SURF feature extraction.

VI. CONCLUSION

This Paper has been envisioned for the purpose of retrieving videos from the Video Database by using efficient algorithms to increase the performance of the system which is difficult in traditional video retrieving system. Ours is a Content Based Video Retrieval system (CBVR). We have presented the working of the system and algorithms used. Experimental results and analysis is presented which shows reliability of the system. Experimental results show that integration of extracted features improves video indexing and retrieval. The fine-tuning of image processing algorithms lead

to more appropriate results and the retrieval time got substantially reduced.

REFERENCES

[1] Madhav Gitte, Harshal Bawaskar, Sourabh Sethi, Ajinkya Shinde — CONTENT BASED VIDEO RETRIEVAL SYSTEM ‖ IJRET Volume: 03 Issue: 06 | Jun-2014

[2] B. V. Patel and B. B. Meshram — CONTENT BASED VIDEO RETRIEVAL ‖ IJMA Vol.4, No.5, October 2012

[3] B. V. Patel and B. B. Meshram — CONTENT BASED VIDEO RETRIEVAL SYSTEMS ‖ IJU Vol.3, No.2, April 2012

[4] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng and Stephen Maybank — A Survey on Visual Content-Based Video Indexing and Retrieval ‖ IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS – PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 6, NOVEMBER 2011

[5] Thao Ngoc Nguyen, Thanh Duc Ngo, Duy-Dinh Le, Shin'ichi Satoh, Bac Hoai Le, Duc Anh Duong — An Efficient Method for Face Retrieval from Large Video Datasets ‖ ACM CIVR '10, July 5-7, Xi'an China

[6] Caifeng Shan — Face Recognition and Retrieval in Video ‖ Springer-Verlag Berlin Heidelberg 2010

[7] Sancho C Sebastine, Bhavani Thuraisingham, Balakrishnan Prabhakaran — Semantic Web for Content Based Video Retrieval ‖ 2009 IEEE International Conference on Semantic Computing

[8] Werner Bailer, Harald Mayer, Helmut Neuschmied, Werner Haas, Mathias Lux, Werner Klieber — Content-based Video Retrieval and Summarization using MPEG-7 ‖ 2004 SPIE and IS&T

[9] W. ZHAO, R. CHELLAPPA, P. J. PHILLIPS and A. ROSENFELD — Face Recognition: A Literature Survey ‖ ACM Computing Surveys, Vol. 35, No. 4, December 2003, pp. 399–458.

[10] Xingquan Zhu, Jianping Fan, Ahmed K. Elmagarmid — Towards facial feature extraction and verification of omni face ‖ IEEE ICIP 2002

[11] Srdan Zagorac, Ainhoa Llorente, Suzanne Little, Haiming Liu, Stefan Ruger — Automated Content Based Video Retrieval ‖ Knowledge Media Institute, The Open University Walton Hall, Milton Keynes, MK7 6AA, UK

[12] Stefan Eickeler, Stefan Muller, Gerhard Rigoll — Video Indexing Using Face Detection and Face Recognition Methods ‖ Gerhard-Mercator-University Duisburg, Department of Computer Science Faculty of Electrical Engineering, 47057 Duisburg, Germany

[13] Y. Alp Aslandogan and Clement T. Yu — Techniques and Systems for Image and Video Retrieval ‖ IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.11, NO.1, JANUARY/FEBRUARY 1999