ISSN: 2321-7782 (Online)
Volume 1, Issue 7, December 2013
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Face Tracking and Recognition in Videos: HMM Vs KNN

Madhumita R. Baviskar
Assistant Professor
Department of Computer Engineering
MIT College of Engineering (Pune University)
Pune - India

Abstract: The problem of tracking and recognizing faces in real-world, noisy videos is addressed. While traditional face recognition is typically based on still images, face recognition from video sequences has become popular recently. This work describes a new method to perform face recognition from video sequences. Faces are detected, tracked and recognized in a video sequence using Hidden Markov Model and K-Nearest Neighbor. Feature extraction for tracking and recognition is performed by Principal Component Analysis. This process also allows locating and extracting facial feature regions around the eyes, nose and mouth. Identity of the tracked subject is established by fusing pose-discriminant and person-discriminant features over the duration of a video sequence. This leads to a robust video-based face recognizer with the state-of-the-art recognition performance. The quality of tracking and face recognition on real-world noisy videos as well as the standard Honda/UCSD database is tested.

Keywords: Face Detection; Face Tracking; Face Recognition; Hidden Markov Model (HMM); K-Nearest Neighbor (KNN); Principal Component Analysis (PCA).

I. INTRODUCTION

Face detection and recognition has emerged as an active area of research in fields such as security system, videoconferencing and identification. The methods for face recognition can be broadly grouped in two classes, i.e. static and dynamic approaches. The former concerns face recognition in single (static) images, while the latter concentrates on people possibly moving in video sequences. The static approach has developed many interesting and powerful techniques, succeeding in different applications. Nevertheless, as far as recognition in video sequences is concerned, much work still remains to be done.

Face recognition in video sequences often involves four important steps as shown in Fig 1.: (1) face detection; (2) feature extraction; (3) feature matching; and (4) recognition.

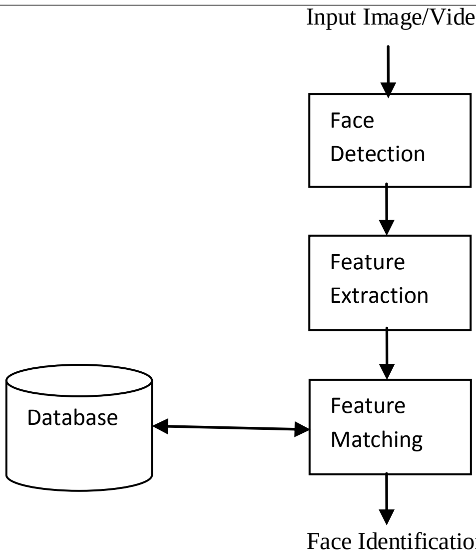

Fig. 1 Face Recognition System

Face detection is defined as the process of extracting faces from scenes. So, the system positively identifies a certain image region as a face. This procedure has many applications like face tracking, pose estimation or compression. The next step - feature extraction- involves obtaining relevant facial features from the data. These features could be certain face regions, variations, angles or measures, which can be human relevant (e.g. eyes spacing) or not. This phase has other applications like facial feature tracking or emotion recognition. Finally, the system does recognize the face. In an identification task, the system would report an identity from a database. This phase involves a comparison method, a classification algorithm and an accuracy measure.

It should be clear that the large amount of data involved in video sequences represent a challenge for real-time implementation of these four steps. In order to circumvent this problem, not all face regions will be taken into account by the foreseen system. Instead, the system is based on extracting information from important facial feature regions (FFRs) defined around specific landmarks, i.e. the eyes, the nose and the mouth.

In this paper, a new method for face tracking and recognition in video is presented that successfully circumvents the difficulties and leverages the benefits of the video setting and is capable of dealing with unconstrained real-world videos. Effectively solving the video-based face recognition problem depends on two tasks: accurate face tracking and interpretation/classification of the tracked data. Face tracking is a critical prior step that localizes the region of the face in video frames, from which a relevant feature set can be extracted and subsequently served as input to the face recognizer. As such, the accuracy of tracking directly impacts the ability to recognize subjects in video.

Visual tracking of objects of interest, such as faces, has received significant attention in the vision community. Accurate tracking is made difficult by the changing appearance of targets due to their non-rigid structure, 3D motion, interaction with other objects (e.g., occlusions) and changes in the environment, such as illumination.

Most existing video-based recognition systems attempt the following: the face is first detected and then tracked over time. Only when a frame satisfying certain criteria (size and pose) is acquired, recognition is performed using still-to-still recognition technique. For this, the face part is cropped from the frame and transformed or registered using appropriate transformations. This tracking-then-recognition approach attempts to resolve uncertainties in tracking and recognition sequentially and separately.

There are several unresolved issues in the tracking-then-recognition approach: criteria for selecting good frames and estimation of parameters for registration. Also, still-to-still recognition does not effectively exploit temporal information. A common strategy that selects several good frames, performs recognition on each frame and then votes on these recognition results for a final solution might be ad hoc.

To overcome these difficulties, a tracking-and-recognition approach is proposed, which attempts to resolve uncertainties in tracking and recognition simultaneously in a unified probabilistic framework. The rest of the paper is organized as follows. A brief review of current video-based recognition approaches is presented in Section 2. Then a new constrained adaptive face tracker is described and shows its basic advantages. Section 6 introduces the video-based recognition framework that relies on the tracker in Section 4. Finally demonstrates, in an extensive set of experiments, the performance of the coupled tracking-recognition framework on Honda/UCSD data. The approach produces successful face tracking results on a large fraction of the videos without instance-specific parameter tuning, while achieving high recognition rates.

II. APPROACHES

Numerous approaches have been developed to recognize faces. While the main focus was on image-based methods in the beginning [4,5,6], it shifted more and more towards video-based approaches in the last years. These are developed in order to overcome shortcomings of image-based recognizers like sensitivity to low resolution, pose variations and partial occlusion.

Zhou et al. [7] use sequence importance sampling (SIS) to propagate a joint posterior probability distribution of identity and motion over time to do tracking and recognition of a person simultaneously. To overcome continuous changes of head pose and facial expressions, Lee et al. represent the appearance of a person by the means of pose manifolds which are connected by transition probabilities. In order to model person-specific appearance and dynamics, Liu and Chen [15] train individual hidden Markov models (HMM) on eigenface image sequences. Confident classifications are used to adapt these models.

A large variety of head pose and illumination variations, as well as occlusion, is encountered in feature films. Arandjelovic and Zisserman built a system to retrieve all faces from a film that match one or multiple query images. The appearance-based approach uses a modified Euclidean distance for classification. Instead of doing frame-based retrieval, Sivic et al. [8] group all face views of a person within the same shot into a face-track, represented as a histogram. Given a query image in one of the scenes, the corresponding face-track is determined. All matching face-tracks are retrieved from the whole film by means of a chi-square goodness-of-fit test..

In this section, the review of recent approaches to tracking and video-based face recognition is done. While many different approaches have been proposed in the past here briefly focus is given on those most related to the approach.

Robustness of tracking and adaptation to changing target appearance and scene conditions are critical properties a tracker should satisfy. Numerous approaches to target modeling have attempted to tackle these issues using view-based appearance models ,contour models ,3D models , mixture models, and kernel representations among others. Direct use of object detectors in discriminative tracking has been proposed more recently, c.f .,.For instance, tackled the challenging case of the low frame rate videos by integrating tracking and detection in a cascade fashion, where the lifespan and feature sets of observation models change during the tracking process to increase efficiency and robustness of the tracker. An adaptive graph-based discriminative tracker combines foreground templates with updating background model. However, learning a model for discrimination from the full background is usually difficult. Furthermore, the problem of small drifts from the face, i.e., the “goodness of the crop”, critical for the recognition stage, is typically not addressed in these approaches.

Coupling of face tracking and recognition has attracted interest in the vision community. For instance, a state-space model was proposed in for classifying videos where state dynamics separates different subjects. assumed that the appearance of faces lies on a probabilistic manifold specific to a subject’s identity, approximated by a set of pose-specific linear subspaces. Subject identification in a video sequence is accomplished by finding the closest manifold (identity) where distance is computed using a temporal fusion in a Bayesian fashion. However, this approach suffers from the need for off-line trained subject-specific trackers, which increases the number of model parameters that need to be set preventing scalability of such an approach.

This approach was extended to simultaneously deal with tracking and recognition using an initial generic appearance manifold that is adapted, in the course of tracking, to a person-specific manifold. The on-line adaptation process, however, relies on training using synthesized face images specific to the currently predicted identity which may limit this approach to situations where such models are available. Very recently, the face recognition problem in real-world videos, containing uncontrolled variations in facial appearance was considered. They accomplish this by assigning confidence scores from local classifiers to the face images in each frame, and then obtaining the sequence level prediction using heuristic weighting of frame-based confidences. Despite promising results, the need for significant parameter tuning and heuristic integration schemes may limit the generalization of this approach.

III. Mathematical Framework

A. *Motivations*

It has been shown that the set of images of an object under all viewing conditions can be considered as a low-dimensional manifold in the image space. For video face recognition, the foremost important image variation that needs to be adequately modeled is due to pose variation, the relative orientation between the camera and the object, and the work is limited to this. Other important image variations such as shape changes (e.g., expression variation) and partial occlusions are not directly modeled in this work. Although such variations are likely to occur in video sequences, their occurrences will be considered to be episodic, and tracking and recognition under these episodic circumstances will be tackled in the underlying framework with the aid of a probabilistic method detailed later.

If the appearance manifold of a face is known, tracking and recognition become straightforward. Suppose there is a set of *N* faces (indexed by *k*) that we wish to track and recognize. Let M_k denote the appearance manifold of person *k*, and $\{F_1, . . . ,F_l\}$ denotes a video sequence of l frames. For each frame, the tracking/recognition system produces an estimate of the face's location in the image and also its identity. In this work, the location of a face in an image is specified by a rectangular region that contains the face, and the rectangular region is represented by a set u of five parameters, specifying the rectangular region's center (in image coordinates), its width and height as well as its orientation. If $f(u,F_t)$ denotes the cropping function (*f* returns the subimage *I* of F_t enclosed in the rectangular region specified by *u*), below tracking and recognition algorithm can be succinctly summarized by the following optimization problem [1]

$$(u_t^*, k_t^*) = \arg\min_{u,k} d(f(u,F_t), M_k) \tag{1}$$

Where *d (I,Mk)* denotes the usual L^2 distance between an image and manifold *Mk*. The pair (u_t^*, k_t^*) is the tracking/recognition result for frame *t.*

The domain of the optimization (**u**,*k)*, can be very large. Optimization techniques for continuous objective functions are not available. One possible solution is to discretize the domain and solve the optimization problem on the discretized domain, by drawing a large number of samples of u, and finding the minimum among the samples. Note that *k* indexes a discrete variable not a continuous one; therefore, the actual number of samples is the product of the number of samples for **u** and the number *N* of individuals to be recognized. If the number of samples for **u** is large (which is usually the case), even a small *N* would have generated a great quantity of samples for the algorithm to process and hence invariably limit its performance. To reduce the number of necessary samples, each variable in Eq. (1) is independently minimized, i.e., minimize **u** with fixed *k* and vice versa:

$$u_t^* = \arg\min_{u} d(f(u,F_t), M_{k_{t-1}^*}) \tag{2}$$

$$k_t^* = \arg\min_{k} d(f(u_t^*,F_t), M_k) \tag{3}$$

The two sub optimization problems correspond exactly to the tracking and recognition problems, respectively. In Eq. (2) a tracking problem is solved with appearance model provided by *Mk,* whereas Eq. (3) is a recognition problem using the tracking result **u** as the input. Therefore, within this framework, the recognizer uses the tracker's result as input, and it updates the internal appearance model used by the tracker through the identity variable k. The tight coupling between the tracking and recognition components is achieved via the shared appearance models $M_1, . . . ,M_N$. Another difficulty of solving Eq. (1) directly is related to the definition of the L^2 distance $d(I,Mk)$ between an image I and a manifold Mk in the image space. By definition, $d(I,Mk) = d(I,x^*)$ with x^* is a point on Mk having minimal L^2 distance to I (See Fig. 2).

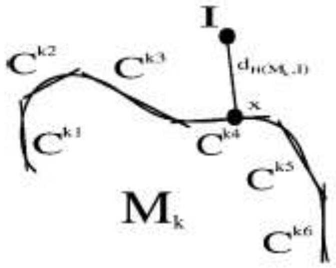

Fig. 2 Appearance manifold. A complex and nonlinear manifold M_k can be approximated as the union of several simpler submanifolds; here, each submanifold C^{ki} is represented by a PCA plane.

B. Face Tracking and Recognition

In this section, tracking/recognition algorithm is outlined. The tracker and recognizer compute Eqs. (2 and 3) respectively. Given the current frame F_t from a video sequence and assuming that the tracking result for the previous frame is u^*_{t-1}, the tracker samples a collection of subimages specified by different **u** based on a Gaussian distribution centered at u^*_{t-1}. Eq. (4) is evaluated by the tracker (with f as the cropping function)

$$u_t^* = \arg\min_u d(f(u,F_t), C^{ki}_{t-1}) \tag{4}$$

The tracker determines a subimage $I_t = f(u_t^*, F_t)$ which has the shortest distance to the submanifold C^{ki}_{t-1} determined in the previous frame. Next, the recognizer uses the subimage I_t returned by the tracker to compute the distance $d\ (I_t,M_k)$ for each person k.

IV. FACE TRACKING SYSTEM

In a probabilistic framework, tracking can be seen as (online) temporal filtering that estimates:[10]

$$P(u_t, F_{0:t}),\ \text{for } t = 1,2,\ldots, \tag{5}$$

Where F_t is the input image frame and u_t is the tracking state at time t. The initial state u0 is assumed to be known. In this paper, similarity transformation parameters $u_t = [c_x, c_y,\rho,\varphi]^T$ are used, where the first two elements are the center position of the square tracking box, ρ is the scale w.r.t. the standard image size (48 × 48), and φ is the inplane rotation angle from the horizontal axis (Fig. 3). The tracker is required to localize the face in space (c_x,c_y) as well as in size and orientation. Accurate estimation of all four parameters is crucial for subsequent use of the detected image in the face recognition phase which is typically sensitive to alignment. Given F_t, the tracking state u_t determines the cropped face image I_t by the warping function $I_t = \omega(u_t, F_t)$, as illustrated in Fig. 3.

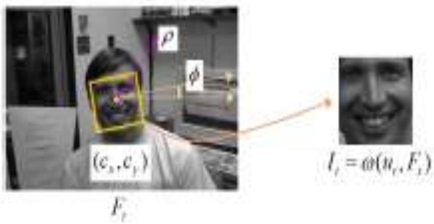

Fig. 3 The wrapping function from the tracking state $u_t= [c_x , c_y , \rho , \varphi]^T$ (solid box) to a cropped image I_t.

The energy $E(I_t)$ plays a crucial role as it estimates the confidence of a candidate particle in terms of its quality. The simplest first-frame (or a two-frame) tracker has a fixed target model as the initial track I_0 (or the previous track I_{t-1}), which defines the energy as a distance to this template, namely, $E(I_t) = d(I_t, I_0)$ (or $E(I_t) = d(I_t, I_{t-1})$), where $d(\cdot, \cdot)$ is a distance measure in the image space. These simplistic energy functions typically make the tracker either too inflexible or too susceptible to appearance variations due to changes in 3D face orientation, scene lighting, occlusions, etc.

V. Adaptive Tracking with Visual Constraints

Since work is restricted to the class of human faces, one can define a reasonable set of visual constraints that serves as an indicator for detection or correction of drifting. In this work propose two such constraints; one for facial pose and the other for the alignment of the cropped faces.

A. *Pose constraints*

To constrain the appearance across different (out-of-plane) poses, a set of pose subspaces is constructed. A set of linear subspaces encapsulated in the model $M_p = \{(\mu_i, B_i)\}_{i=pose}$ is considered. Fig. 4(a) illustrates one such model. Then defining the energy related to this pose constraint as a minimum distance [10] among the pose subspaces, namely, $d(I_t, M_p) = min_i\, d(I_t, (\mu_i, B_i))$ is done.

The pose subspace model can be estimated from a dataset of differently oriented face images. Here the face data from the Honda/UCSD video database in [12] is used. Detection of faces and manually aligning them is taken place, also obtaining face images of different poses with different people and varying illumination conditions. The poses are roughly categorized into 5 clusters (frontal, left/right 45-deg, left/right profile), and data from each pose cluster is used to train a PCA subspace, forming a set of pose subspaces.

B. *Alignment constraints*

The alignment constraint determines whether or not the candidate image contains a well-aligned and cropped face. Fig. 4(b) depicts some examples of correctly and incorrectly cropped faces. Determining how well an image is cropped can be accomplished using a confidence score of a classifier that discriminates well-cropped face images from the drifted images or, possibly, non-faces. In this case the face data (for all poses and subjects) used in pose subspace learning become positive examples for learning a classifier.

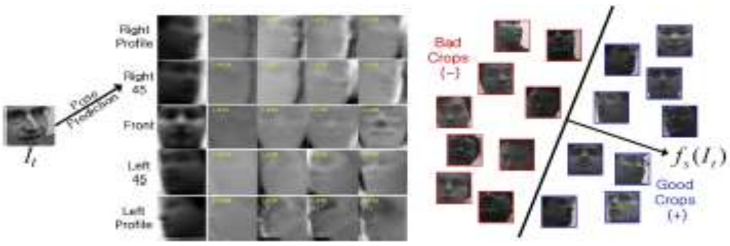

Fig. 4 Two visual constraints. (a) Each row represents the PCA subspace learned for each pose. The distances from I_t to the subspaces are computed, and the minimum is selected as a predicted pose for I_t. (b) classifier that discriminates well-cropped face images (+, in blue) against drifted or shifted images (−, in red).

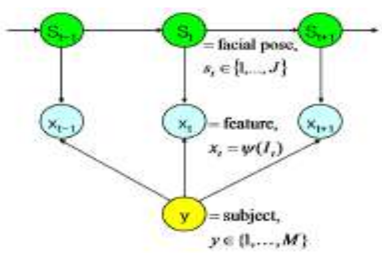

Fig. 5 A graphical model for face recognition

VI. VIDEO BASED FACE RECOGNITION

Tracking provides well cropped/aligned face images $I_{1,...,T}$ for face recognition. The task of the face recognition phase is to label an arbitrary video sequence with the identity of the person in the video clip. Assumption of a single subject in each clip is made. However, in the case of multiple people, recognition is not affected if multiple face tracks can be provided.

Facial pose presents an appealing choice for the hidden state. The pose space may allow the use of well-defined discriminative pose features in the face recognition HMM. In particular, s_t represents a particular pose among J possible poses, $s_t \in \{1, . . . , J\}$, and x_t denotes a pose-discriminant feature vector described below. The appearance sequence of length T is then modeled by the generative model (HMM) for each subject y:[10]

$$P_y(\mathbf{x}, \mathbf{s}) = P(s_1) \left[\prod_{t=2}^{T} P(s_t \mid s_{t-1})\right] \left[\prod_{t=1}^{T} P_y(x_t \mid s_t)\right] \tag{6}$$

The pose discriminating features are obtained using the PCA (Principal Component Analysis). Each image I_t is projected onto a discriminant subspace ψ trained by PCA on pose labeled data to yield $x_t = \psi (I_t)$.

The subject-specific observation density, needed by each HMM, is modeled as a Gaussian distribution, namely, $P_y(x_t \mid s_t = j) = \mathcal{N}(x_t; m_j^y, V_j^y)$ where m_j^y and V_j^y are the mean and the covariance for pose j of subject y, respectively. We let the pose dynamics be shared across all y's. This is a reasonable assumption, implying that the way poses change is generic and independent of any one particular person. The model is trained by the EM algorithm with the subject labeled face sequence data.

A. *Face Recognition using K-Nearest Neighbor (KNN)*

The simplest classification scheme is a nearest neighbor classification in the image space. Under this scheme an image in the test set is recognized by assigning to it the label of the closest point in the learning set, where distance are measured in image space. If all images have been normalized to be zero mean and have unit variance, then this procedure is equivalent to choosing the image in learning set that best correlates with the test image. Because of normalization process, the result is independent of light source intensity and the effects of a video camera's automatic gain control.

The Euclidean distance metric [11] is often chosen to determine the closeness between the data points in KNN. A distance is assigned between all pixels in a dataset. Distance is defined as the Euclidean distance between two pixels. The Euclidean metric is the function d: R^n X $R^n \rightarrow R$ that assigns to any two vectors in

Euclidean n-space $X=(x_1,\ldots, x_n)$ and $Y=(y_1,\ldots, y_n)$ the number,

$$d(x,y) = \sqrt{((x_1 - y_1)^2 + \ldots + (x_n - y_n)^2)} \tag{7}$$

This gives the "standard" distance between any two vectors in R^n. From these distances, a distance matrix is constructed between all possible pairings of points (x, y).

The k closest neighbors S_i, $i = 1, 2, . . . , k$ of a test vector x are selected with score $s_i = d(x, S_i)$. Because the distances and, thus, the resulting scores can differ largely between frames, they need to be normalized. This is achieved with linear *min-max normalization* [12],

$$s_i' = 1 - \frac{s_i - s_{min}}{s_{max} - s_{min}} \qquad i=1,2,\ldots,k \tag{8}$$

Which maps the scores to [0, 1]. Of course, among the k closest representatives, there can be several ones from the same class. Since some people have far fewer representatives than others, care must be taken that their scores are not dominated by those. Individual scores are selected by a simple max-rule, which only selects the maximum score for each class.

1) KNN Algorithm

- Each data pixel value within the data set has a class label in the set, Class = $\{c_1,...,c_n\}$.
- The data points', k-closest neighbors (k being the number of neighbors) are then found by analyzing the distance matrix.
- The k-closest data points are then analyzed to determine which class label is the most common among the set.
- The most common class label is then assigned to the data point being analyzed.

VII. Experiments and Results

In this section, description of experimental evaluations of the discussed tracking/recognition algorithm is given. The work is implemented in Matlab. Honda/UCSD dataset is used for testing and training purposes. The database contains around 500 grayscale images in JPG format of 10 individuals. There are 50 images per subject in the database.

A. Data preparation and training process

Each video sequence is recorded an indoor environment at 15 frames per second, and each lasted for at least 20s. The resolution of each video sequence is 640x480. Every individual is recorded in at least two video sequences. Since we believe that pose variation provides the greatest challenge to recognition, all the video sequences contain significant 2-D (in-plane) and 3-D (out-of-plane) head rotations. In each video, the person rotates and turns his/her head in his/her own preferred order and speed, and typically in about 15 s, the individual is able to provide a wide range of different poses. In addition, some of these sequences contain difficult events which a real-world tracker/recognizer would likely encounter, such as partial occlusion, face partly leaving the field of view, and large scale changes, etc.

The main part of the training procedure is to compute the local linear approximations of each person’s appearance manifold as well as the connectivity between these local approximations. For this, a simple face tracker (a variant of the EigenTracker) was applied to each training sequence. The tracker returns a cropped face image for each frame, and these cropped images are the training images used to compute the approximation of the appearance manifold for each individual. All of the cropped images produced by the tracker are visually inspected. This manual intervention during the training process is inevitable and necessary because the simple EigenTracker used here is prone to loose the target, and it needs to be re-initialized after each failure. The cropped images are down-sampled to a standard size of 19x19 pixels because the tracking windows from different frames are generally of different sizes. Fig. 6 displays some of the cropped and normalized images used as training images.

The cropped and normalized images from each individual are grouped into 7 clusters. A 7-dimensional subspace is computed from the images in each cluster using PCA.

B. Tracking experiments

Fig. 7 displays the tracking results for five key frames from different video sequences.

Fig. 6 Samples of the training videos used in the experiments. All sequences contain significant pose variation.

Fig. 7 Tracking results for different video sequences. Each row displays a set of five key frames from a video sequence.

C. Recognition results

The detected face image is given as input to the Eigen face recognition system which obtains the PCA projected values of reduced dimension, which is used to recognize the face image by preserving the Euclidean structure of a person. The KNN and HMM classifier are used to classify the test image based on their Euclidean structure obtained using PCA.

The graph in Fig. 8 explains that KNN classifier is best suited for classifying persons based on their images due to its lesser execution time and better accuracy than other commonly used methods which include Hidden Markov Model.

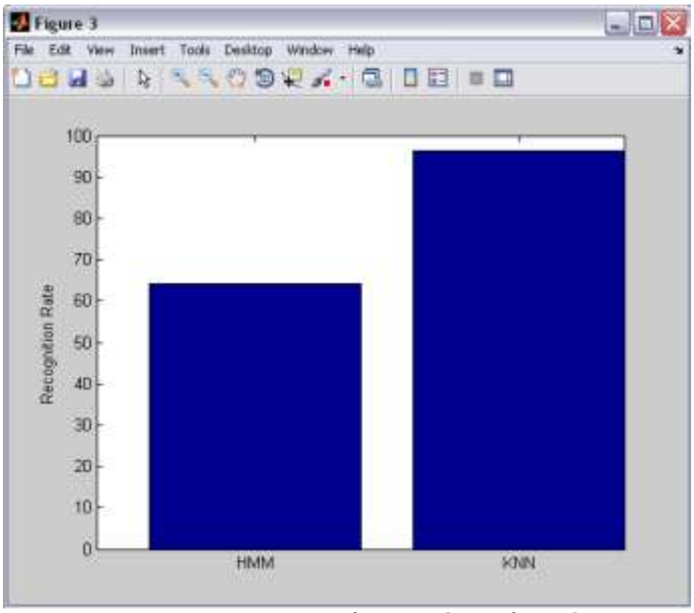

Fig. 8 Average recognition rate of HMM and KNN for 9 subjects

VIII. APPLICATIONS

There are numerous application areas in which face recognition can be exploited for these two purposes, a few of which are outlined below.[2]

- Security (access control to buildings, airports/seaports, ATM machines and border checkpoints computer/network security, email authentication on multimedia workstations).
- Surveillance (a large number of CCTVs can be monitored to look for known criminals, drug offenders, etc. and authorities can be notified when one is located.
- General identity verification (electoral registration, banking, electronic commerce, identifying newborns, national IDs, passports, drivers' licenses, employee IDs).
- Criminal justice systems (mug-shot/booking systems, post-event analysis, forensics).
- Image database investigations (searching image databases of licensed drivers, benefit recipients, missing children, immigrants and police bookings).
- "Smart Card" applications (in lieu of maintaining a database of facial images, the face-print can be stored in a smart card, bar code or magnetic stripe, authentication of which is performed by matching the live image and the stored template) .
- Multi-media environments with adaptive human computer interfaces (part of ubiquitous or context aware systems, behavior monitoring at childcare or old people's centers, recognizing a customer and assessing his needs).
- Video indexing (labeling faces in video).
- Witness face reconstruction.

In addition to these applications, the underlying techniques in the current face recognition technology have also been modified and used for related applications such as gender classification , expression recognition and facial feature recognition and tracking, each of these has its utility in various domains: for instance, expression recognition can be utilized in the field of medicine for intensive care monitoring while facial feature recognition and detection can be exploited for tracking a vehicle driver's eyes and thus monitoring his fatigue, as well as for stress detection.

IX. Conclusion

The tracking algorithm localizes the face from the given input image using the feature detection method where face is located using template matching. The detected face image is projected using Eigen face analysis and classified using the K nearest neighborhood (KNN) classifier and Hidden Markov Model (HMM) classifier. This comparison shows that the KNN algorithm is efficient than HMM.

References

1. Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang And David Kriegman, Visual Tracking And Recognition Using Probabilistic Appearance Manifolds, Computer Vision And Image Understanding, 99, 2005, 303-331.
2. Navin Prakash, Dr. Yash Pal Singh, Dr. D. Rai, Emerging Trends of Face Recognition : A Review, World Applied Programming Journal, Vol(2), No(4), April 2012, 242-250.
3. Shujaat Ali, Novel fast and Efficient Face Recognition Technique, International Journal of Emerging trends in Engineering and Development, Issue 2, Vol.6, September 2012, 564-570.
4. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class-specific linear projection, IEEE Trans. PAMI, 19(7):711–720, July 1997.
5. A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. PAMI, 23(6):643–660, 2001.
6. M. Turk and A. Pentland. Eigenfaces for recognition. J. Cogn. Neurosci., 3(1):71–86, 1991.
7. S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. CVIU, 91(1-2):214–245, Feb. 2003.
8. J. Sivic, M. Everingham, and A. Zisserman. Person spotting: Video shot retrieval for face sets. In CIVR, volume 3568, pages 226–236. Springer, July 2005.

Author(s) Profile

Ms. Madhumita Baviskar, is working as an Assistant Professor in the Department of Computer Engineering, MIT College of Engineering, Pune. It is affiliated to Pune University, Maharashtra, India. She received the B.E. degree in Information Technology from P.E.S. College of Engineering, Aurangabad (MS) (Affiliated to Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS)) in 2007 and M.E. degree in Computer Science and Engineering from Government Engineering College, Aurangabad (MS) (Autonomous) in 2013.