Facial Emotion Recognition in Videos Using Hmm

Rani R. Pagariya[1,] Mahip M. Bartere[2]

[1] M.E (2nd sem. CSE), GHRCEM, Amravati,
[2] Lecturer in CSE Dept. at GHRCEM, Amravati

ABSTRACT:

Human computer interaction is an emerging field in computer science. It is said that for a computer to be intelligent it must interact with human the way human and human interact. Human mainly interact through speech along with that it interact through physical gestures and postures which mainly include facial expressions. This paper discusses what the facial expressions are, the needs of recognizing the facial expression, and how to recognize the facial expression? Two such methods of recognizing the facial expressions using HMM are provided. One is Emotion Recognition from Facial Expressions using Multilevel HMM and another one compute a derivative of features with histogram differencing and derivative of Gaussians and model the changes with a hidden Markov model.

Keywords: *Derivatives, Emotion, Facial expression, Features, Hidden Markov Models, State sequence, ML classifier.*

I. INTRODUCTION

Emotion plays an important role in human life. At different movements of time human faces reflects differently with different intensity, which reflects there mood. Facial features and expressions are critical to everyday communication. Besides speaker recognition, face assists a number of cognitive tasks: for example, the shape and motion of lips forming visemes can contribute greatly to speech comprehension in a noisy environment. While intuition may imply otherwise, social psychology research has shown that conveying messages in meaningful conversations can be dominated by facial expressions, and not spoken words. This result has led to renewed interest in detecting and analyzing facial expressions in not just extreme situations, but also in everyday human–human discourse. [1] There are different six facial expressions considered over here: happy, angry, surprise, disgust, fear, sad. And evaluate using HMM.

II. LITERATURE SURVEY:

The origins of facial expression analysis go back into the 19th century, when Darwin originally proposed the concept of universal facial expressions in man and animals. Since the early 1970s, Ekman and Friesen (1975) have performed extensive studies of human facial expressions, providing evidence to support this universality theory. These ‘universal facial expressions’ are those representing happiness, sadness, anger, fear, surprise, and disgust. To prove this, they provide results from studying facial expressions in different cultures, even primitive or isolated ones. These studies show that the processes of expression and recognition of emotions on the face are common enough, despite differences imposed by social rules. Ekman and Friesen used FACS to manually describe facial expressions, using still images of, and usually extreme, facial expressions. This work inspired researchers to analyze facial expressions by tracking prominent facial features or measuring the amount of facial movement, usually relying on the ‘universal expressions’ or a defined subset of them. In the 1990s, automatic facial expression analysis research gained much interest, mainly thanks to progress, in the related fields such as image processing (face detection, tracking and recognition) and the increasing availability of relatively cheap computational power.

In one of the ground-breaking and most publicized works, Mase and Pentland (1990) used measurements of optical flow to recognize facial expressions. In the following, Lanitis et al. used a flexible shape and appearance model for face identification, pose recovery and facial expression recognition. Black and Yacoob (1997) proposed local parameterized models of image motion to recover non-rigid facial motion, which was used as input to a rule-based basic expression classifier; Yacoob and Davis (1996) also worked in the same framework, this time using optical flow as input to the rules. Local optical flow was also the basis of Rosenblum’s work, utilizing a radial basis function network for expression classification. Otsuka and Ohya utilized the 2D Fourier transform coefficients of the optical flow as feature vectors for a hidden Markov model(HMM).

Regarding feature-based techniques, Donato, Bartlett, Hager, Ekman, and Sejnowski (1999) tested different features for recognizing facial AUs and inferring the facial expression in the frame. Oliver et al. tracked the lower face to extract mouth shape information and fed them to an HMM, recognizing again only universal expressions [1].

III. HIDDEN MARKOV MODEL

The Hidden Markov Model (HMM) is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing, and in particular speech processing, but have also been applied with success to low level NLP tasks such as part-of-speech tagging, phrase chunking, and extracting target information from documents. Andrei Markov gave his name to the mathematical theory of Markov processes in the early twentieth century, but it was Baum and his colleagues that developed the theory of HMMs in the 1960[2]. A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states. An HMM can be considered as the simplest dynamic Bayesian network.

Markov Processes fig1 depicts an example of a Markov process. The model presented describes a simple model for a stock market index. The model has three states, Bull, Bear and Even, and three index observations up, down, unchanged. The model is a finite state automaton, with probabilistic transitions between states. Given a sequence of observations, example: up-down-down we can easily verify that the state sequence that produced those observations was: Bull-Bear-Bear, and the probability of the sequence is simply the product of the transitions, in this case 0.2 ×0.3 × 0.3 [2].

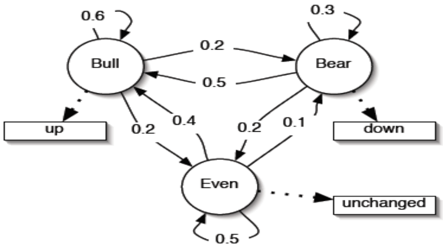

Figure 1: Markov process example[1]

The Markov property guarantees that the future evolution of the process depends only on its present state, and not on its past history [3].Hidden Markov Models Diagram 2 shows an example of how the previous model can be extended into a HMM. The new model now allows all observation symbols to be emitted from each state with a finite probability. This change makes the model much more expressive and able to better represent our intuition, in this case, that a bull market would have both good days and bad days, but there would be more good ones.

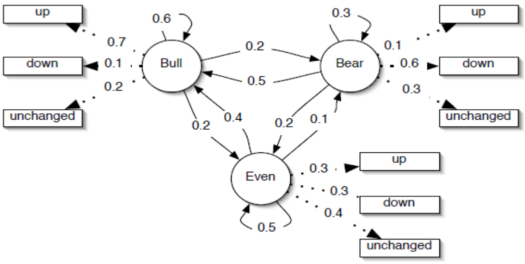

Figure 2: hmm

The key difference is that now if we have the observation sequence up-down-down then we cannot say exactly what state sequence produced these observations and thus the state sequence is ‘hidden’. We can however calculate the probability that the model produced the sequence, as well as which state sequence was most likely to have produced the observations. The next three sections describe the common calculations that we would like to be able to perform on a HMM. The formal definition of a HMM is as follows:

$$\lambda = (A, B, \pi) \qquad (1)$$

S is our state alphabet set, and V is the observation alphabet set:

$$S=(s_1,s_2,\cdots,s_N)$$ (2)

$$V=(v_1,v_2,\cdots,v_M)$$ (3)

We define Q to be a fixed state sequence of length T, and corresponding observations O:

$$Q=q_1,q_2,\cdots,q_T$$ (4)

$$O=o_1,o_2,\cdots,o_T$$ (5)

A is a transition array, storing the probability of state j following state i . Note the state transition probabilities are independent of time:

$$A=[a_{ij}],a_{ij}=P(q_t=s_j|q_{t-1}=s_i)$$ (6)

B is the observation array, storing the probability of observation k being produced from
the state j, independent of t:

$$B=[b_i(k)],b_i(k)=P(x_t=v_k|q_t=s_i)$$ (7)

π is the initial probability array:

$$\pi=[\pi_i],\pi_i=P(q_1=s_i)$$ (8)

Two assumptions are made by the model. The first, called the Markov assumption, states that the current state is dependent only on the previous state, this represents the memory of the model:

$$P(q_t|q_1^{t-1})=P(q_t|q_{t-1})$$ (9)

The independence assumption states that the output observation at time t is dependent only on the current state, it is independent of previous observations and states:

$$P(o_t|o_1^{t-1},q_1^t)=P(o_t|q_t)$$ (10) [3].

IV. METHODOLOGIES

4.1. Expression Recognition Using Emotion-Specific HMMs

Since the display of a certain facial expression in video is represented by a temporal sequence of facial motions it is natural to model each expression using an HMM trained for that particular type of expression. There will be six such HMMs, one for each expression: * *happy, angry, surprise, disgust, fear, sad*/ . There are several choices of model structure that can be used. The two main models are the left-to-right model and the ergodic model. In the left-to-right model, the probability of going back to the previous state is set to zero, and therefore the model will always start from a certain state and end up in an 'exiting' state. In the ergodic model every state can be reached from any other state in a finite number of time steps. Otsuka and Ohya used left-to-right models with three states to model each type of facial expression. The advantage of using this model lies in the fact that it seems natural to model a sequential event with a model that also starts from a fixed starting state and always reaches an end state. It also involves fewer parameters, and therefore is easier to train. However, it reduces the degrees of freedom the model has to try to account for the observation sequence. There has been no study to indicate that the facial expression sequence is indeed modeled well by the left-to-right model. On the other hand, using the ergodic HMM allows more freedom for the model to account for the observation sequences, and in fact, for an infinite amount of training data it can be shown that the ergodic model will reduce to the left-to-right model, if that is indeed the true model. In this work both types of models were tested with various numbers of states in an attempt to study the best structure that can model facial expressions.The observation vector O_t for the HMM represents continuous motion of the facial action units. Therefore, B is represented by the probability density functions (pdf) of the observation vector at time *t* given the state of the model. The Gaussian distribution is chosen to represent these pdf's, i.e.

$$B=\{b_i(O_t)\}\sim N(\mu_j,\Sigma_j),1\le j\le N$$ (11)

Where μ_j and Σ_i are the mean vector and full covariance matrix, respectively.

The parameters of the model of emotion-expression specific HMM are learned using the well-known Baum- Welch re estimation formulas For learning, hand labeled sequences of each of the facial expressions are used as ground truth sequences, and the Baum algorithm is used to derive the maximum likelihood (ML) estimation of the model parameters (λ)Parameter learning is followed by the construction of a ML classifier. Given an observation sequence O_t , where $t\in(1,T)$, the probability of the observation $P(O_t|\lambda_j)$ given each of the six models is computed using the forward backward procedure .The sequence is classified as the emotion corresponding to the model that yielded the highest probability, i.e.

$$c^* = \underset{1 \leq c \leq 6}{argmax}[P(O|\lambda_c)]$$

(12) [4, 5].

4.2. Automatic Segmentation and Recognition of Emotions Using Multilevel HMM.

The main problem with the approach taken in the previous section is that it works on isolated facial expression sequences or on presegmented sequences of the expressions P from the video. In reality, this segmentation is not available, and therefore there is a need to find an automatic way of segmenting the sequences. Concatenation of the HMMs representing phonemes in conjunction with the use of grammar has been used in many systems for continuous speech recognition. Dynamic programming for continuous speech has also been proposed in different researches. It is not very straightforward to try and apply these methods to the emotion recognition problem since there is no clear notion of language in displaying emotions. Otsuka and Ohya [4] used a hueristic method based on changes in the motion of several regions of the face to decide that an expression sequence is beginning and ending. After detecting the boundries, the sequence is classified to one of the emotions using the emotion-specific HMM. This method is prone to errors because of the sensitivity of the classifier to the segmentation result. Although the result of the HMM's are independent of each other, if we assume that they model realistically the motion of the facial features related to each emotion, the combination of the state sequence of the six HMM's together can provide very useful information and enhance the discrimination between the different classes. Since we will use a left-to-right model (with return), the changing of the state sequence can have a physical attribute attached to it (such as opening and closing of mouth when smiling), and therefore there we can gain useful information from looking at the state sequence and using it to discriminate between the emotions at each point in time.

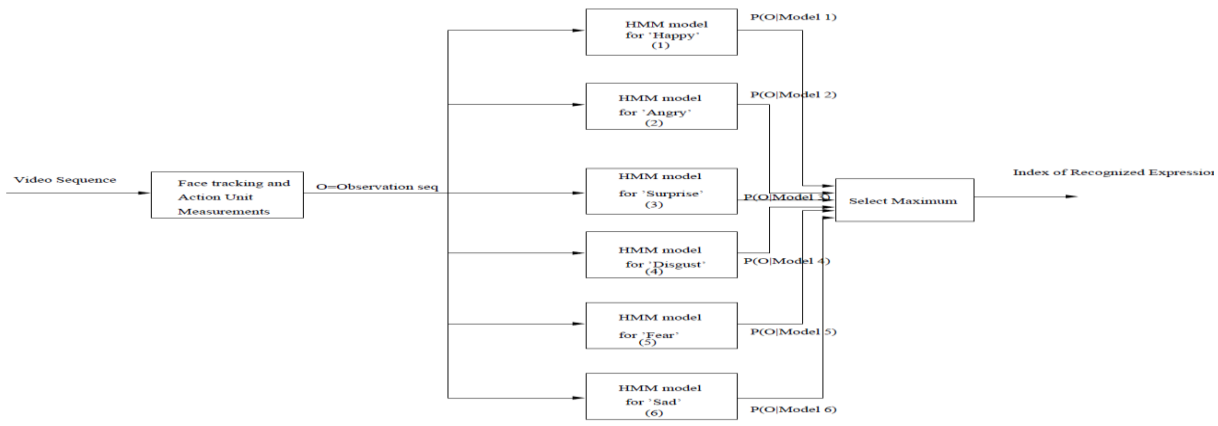
Video Sequence
Face tracking and Action Unit Measurements
O=Observation seq
HMM model for 'Happy' (1)
HMM model for 'Angry' (2)
HMM model for 'Surprise' (3)
HMM model for 'Disgust' (4)
HMM model for 'Fear' (5)
HMM model for 'Sad' (6)
P(O|Model 1)
P(O|Model 2)
P(O|Model 3)
P(O|Model 4)
P(O|Model 5)
P(O|Model 6)
Select Maximum
Index of Recognized Expressio

Figure 3: **Maximum likelihood classifier for emotion specific HMM**

To solve the segmentation problem and enhance the discrimination between the classes, a different kind of architecture is needed. Figure 4 shows the proposed architecture for automatic segmentation and recognition of the displayed expression at each time instance. As can be seen, the motion features are fed continuously to the six emotionsspecific HMMs. The state sequence of each of the HMMs is decoded and used as the observation vector for the high- level HMM. The high-level HMM consists of seven states, one for each of the six emotions and one for *neutral*. The *neutral* state is necessary as for the large portion of time, there is no display of emotion on a person's face. The transitions between emotions are imposed to pass through the *neutral* state since it is fair to assume that the face resumes a neutral position before it displays a new emotion. For instance, a person cannot go from expressing happy to sad without returning the face to its neutral position (even for a very brief interval). The recognition of the expression is done by decoding the state that the high-level HMM is in at each point in time since the state represents the displayed emotion. To get a more stable recognition, output of the classifier will actually be a smoothed version of the state sequence, i.e., the high-level HMM will have to stay in a particular state for a long enough time in order for the output to be the emotion related to that state.

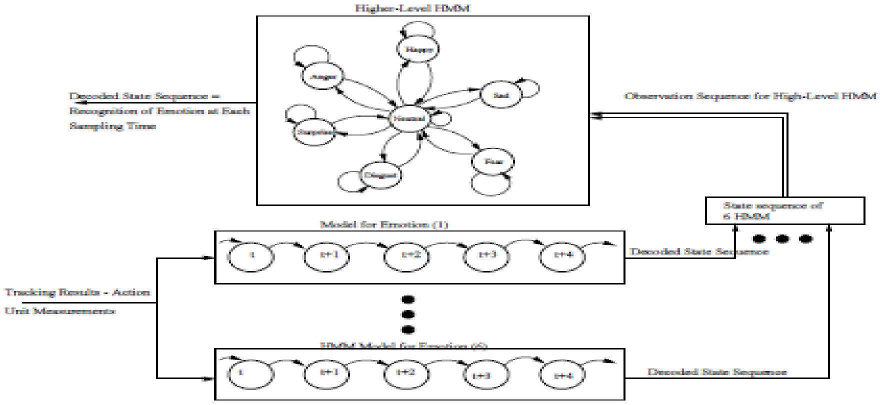
Higher-Level HMM
Happy
Anger
Sad
Neutral
Surprise
Fear
Disgust
Decoded State Sequence = Recognition of Emotion at Each Sampling Time
Observation Sequence for High-Level HMM
State sequence of 6 HMM
Model for Emotion (1)
t
t+1
t+2
t+3
t+4
Decoded State Sequence
Tracking Results - Action Unit Measurements
HMM Model for Emotion (6)
Decoded State Sequence

Figure4: **Multilevel HMM architecture for automatic segmentation and recognition of emotion**

The training procedure of the system is as follows:

- Train the emotion-specific HMMs using a hand segmented sequence as described in the previous section.
- Feed all six HMMs with the continuous (labeled) facial expression sequence. Each expression sequence contains several instances of each facial expression with *neutral* instances separating the emotions.
- Obtain the state sequence of each HMM to form the six-dimensional observation vector of the higher-level HMM,

$$O_t^h = [q_t^{(1)}, \ldots, q_t^{(6)}]^T \qquad (13)$$

where q_t^i is the state of the *i*th emotion-specific HMM. The decoding of the state sequence is done using the Vitterbi algorithm [8].

- Learn the probability observation matrix for each state of the high-level HMM using $P(q_j^{(i)}|S_k) = \{\text{expected frequency of model } i \text{ being in state } j \text{ given that the true state was } k\}$and,

$$B^{(h)} = \{b_k(O_t^h)\} = \{\prod_{i=1}^{6}(P(q_j^{(i)}|S_k)\} \qquad (14)$$

where $j \in (1$,*Number of States for Lower Level HMM).*

- Compute the transition probability $A = \{a_{kl}\}$/ of the high-level HMM using the frequency of transiting from each of the six emotion classes to the *neutral* state in the training sequences and from the *neutral* state to the other emotion states. For notation, the *neutral* state is numbered 7, and the other states are numbered as in the previous section. It should be noted that the transition probabilities from one emotion state to another that is not *neutral* are set to zero.
- Set the initial probability of the high-level HMM to be 1 for the *neutral* state and 0 for all other states. This forces the model to always start at the *neutral* state and assumes that a person will display a *neutral* expression in the beginning of any video sequence. This assumption is made just for simplicity of the testing.

The steps followed during the testing phase are very similar to the ones followed during training. The face tracking sequence is fed into the lower-level HMMs and a decoded state sequence is obtained using the Viterbi algorithm The decoded lower-level state sequence O_t^h is fed into the higher-level HMM and the observation probabilities are computed using equtation. Note that in this way of computing the probability, it is assumed that the state sequences of the lower-level HMMs are independent given the true labeling of the sequence. This assumption is reasonable since the HMMs are trained independently and on different training sequences. In addition, without this assumption, the size of B will be enormous, since it will have to account for all possible combinations of states of the six lower-level HMMs, and it would require a huge amount of training data.Using the Viterbi algorithm again for the high-level HMM, a most likely state sequence is produced. The state that the HMM was in at time *t* corresponds to the expressed emotion in the video sequence at time *t*. To make the classification result robust to undesired fast changes, a smoothing of the state sequence is done by not changing the actual classification result if the HMM did not stay in a particular state for more than *T* times, where *T* can vary between 1 and 15 samples (assuming a 30-Hz sampling rate). The introduction of the smoothing factor *T* will cause a delay in the decision of the system, but of no more than *T* sample times [4].

4.3. Facial Emotion Recognition by Computing Derivatives of Features.

The system overview is shown in Fig:

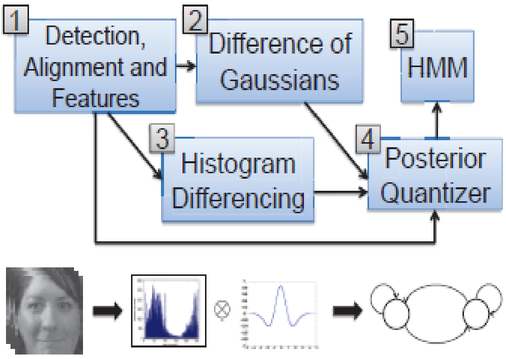

Figure 5: overview of system.

(1)Face ROI is detected with a boosted cascade of Haar-like features, aligned with SIFT Flow to a reference image, and Local Phase Quantization (LPQ) texture features are extracted. The derivatives of features are estimated with two methods: (2) with a fine spatial granularity with Difference of Gaussians (DoG) and (3) with a course spatial granularity with histogram differencing (HD) of LPQ histograms. (4) A support vector machine (SVM) outputs posterior probabilities for emotion labels from each of the three feature vectors and yhe posterior probabilities are quantized into a single observation vector. (5) A hidden Markov model computes the optimal emotion labels, taking of advantage of the co-occurrences between x (t) and x′ (t).Where T is the length of the video. U has mi observable symbols.

4.3.1. Detection, Alignment and Features

Faces are extracted with a boosted cascade of Haar-like features. After extraction, faces are aligned with SIFT Flow to the Avatar Reference Image . The parameters of this algorithm are the number of iterations I. After alignment, LPQ histograms are extracted in each of n × n local regions and the histograms are concatenated to form the feature vector (this process is also called cells, or gridding).

4.3.2. Modeling Temporal Changes

The derivative of features x′ (t) is approximated by two methods: convolution with a DoG filter and difference of feature histograms. DoG has a fine spatial granularity, in that it captures local changes happening at the pixel. Histogram differencing has a course spatial granularity, in that captures global changes happening between the histograms of each cell. **Local derivatives with DoG**: A DoG filter is employed as opposed to a finite difference because the finite difference is sensitive to noise. The i-th feature $\langle x(t)\rangle_i$ is convolved with the DoG filter to approximate the gradient of x (t) with the following equation:

$$\langle x'_{\mathrm{DoG}}(t)\rangle_i \approx \langle x(t)\rangle_i \otimes h(t) \qquad (15)$$

Where $h(t) \sim N(0,\sigma_1) - N(0,\sigma_2)$ and σ1 = 4σ2.
The effect of Eq. 1 is a 1-D temporal gradient of $\langle x(t)\rangle_i$ that has been low-pass filtered to remove noise. h(t) is discretized to 2l, where 3σ2 = l, retaining approx. 99% of the energy of the larger Gaussian.

Global derivatives with HD: Let the feature vector x (t) be composed of a set of n^2 histograms {H1 (t) ,H2 (t) , ...,Hn2 (t)}. The histogram difference is computed with the l_1 metric, for each histogram, between t − δ and t + δ. This is similar to shot transition detection for key_frames, except the histogram of features is used, as opposed to color histograms. A new feature vector $x'_{\mathrm{HD}}(t) \epsilon \Re^{1\times n^2}$ is generated where:

$$\langle x'_{\mathrm{HD}}(t)\rangle_i = \|H_i(t-\delta) - H_i(t+\delta)\|_1 \qquad (16)$$

Where Hi (t) is the i-th histogram at time t, and δ is a spacing parameter.

Observation quantization: A linear SVM is trained to output posterior probabilities. Let $\tilde{w}_i(t)$ be the estimated label at time t from a matcher using the i-th feature set. We hypothesize that a co-occurrence exists, where the feature derivatives perform better when emotion is weak or transitioning, e.g. $\tilde{w}_{DoG}(t)$would properly classify t_0 in Fig. 1, and $\tilde{w}_{LPQ}(t)$would not. The output of the SVM must be fused in such a way as to capture the combination of outputs of the SVM. First, the posterior probabilities across all videos for each matcher are quantized into m bins with k-means clustering. Let $v_i(t)$ be the set of membership of $\tilde{w}_i(t)$at time t, ranging from 0 to m − 1. Second, the quantized probabilities $v_i(t)$ of each matcher are combined into a single observation matrix. Let *u(t)* be the combined, quantized observation at time t:

$$u(t) = \sum_{i=1}^{n} m^{(i-1)} v_i(t) \qquad (17)$$

Where n is the number of different matchers. Let U be the observation sequence defined as:

$$\mathbf{U = \{u\ (t): 0 < t \leq T\}} \qquad (18)$$

Hidden Markov Model: Co-occurrence aware fusion is realized with a Hidden Markov model (HMM). We formulate out HMM as follows: given the observation sequence U, and the HMM, optimal corresponding state sequence Y = y (0) y (1) ...y (T) must be chosen. Y is taken to be the estimated labels; the number of states of the model is equal to the number of classes p. The state transition probability distribution matrix A and observation probability distribution matrix B are estimated from training data. We assign labels with:

$$\mathbf{Y} = \text{argmax}_{y(0)\ldots y(T)} p(y(0)\ldots y(T), \ldots \mathbf{U} \,|\, \lambda(A, B))$$

(19) Where λ is the model. Above equation is solved with dynamic Programming, with the Viterbi algorithm. Because the joint probabilities of the matchers are estimated, the model can fuse information from each matcher in a more meaningful way, as opposed to simply aggregating the labels [6].

V. APPLICATIONS

Facial emotion recognition has applications in: Facial emotion recognition has applications in medicine [6] in treatment of Asperger. Asperger syndrome (AS), is an autism spectrum disorder (ASD) that is characterized by significant difficulties in social interaction, alongside restricted and repetitive patterns of behavior and interests. It differs from other autism spectrum disorders by its relative preservation of linguistic and cognitive development. Although not required for diagnosis, physical clumsiness and atypical (peculiar, odd) use of language are frequently reported. So it is easy to recognize the facial expressions than language.It has application in video games [6] such as Xbox Kinect. Kinect is a motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. Based around a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with the Xbox 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands. Most important use of any facial emotion technique is human-computer interaction to make intelligent tutoring systems and Affective computing [6] is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects. The machine should interpret the emotional state of humans and adapt its behavior to them, giving an appropriate response for those emotions.

VI. CONCLUSION:

In this paper method for emotion recognition from video sequences of facial expression were explored. For facial emotion recognition where a subject can freely express emotions, a derivative of features was more suitable than using the features themselves. While Emotion-specific HMM, relied on segmentation of a continuous video into sequences of emotions (or neutral state), multilevel HMM, performed automatic segmentation and recognition from a continuous signal. By giving feedback to the computer, a better interaction can be achieved. This can be used in many ways. For example, it can help in education by helping children learn effectively with computers. Among the both above methods multilevel HMM is easy as compared to derivative of features.

REFERENCES

[1] Spiros V. Ioannou, Amaryllis T. Raouzaiou, Vasilis A. Tzouvaras, Theofilos P. Mailis, Kostas C. Karpouzis, Stefanos D. Kollias “Emotion recognition through facial expression analysis based on a neurofuzzy network”. Special Issue, Greece, 2005.

[2] “Phil Blunsom.” Hidden Markov Models” August 19, 2004.

[3] Ramon van Handel “Hidden Markov Models” Lecture Note July 28, 2008.

[4] Ira Cohen, Ashutosh Garg , Thomas S. Huang Beckman Institute for Advanced Science and Technology “Emotion Recognition from Facial Expressions using Multilevel HMM”.

[5] Ira Cohen, Nicu Sebe , Larry Chen , Ashutosh Garg, Thomas S. Huang. Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana "Facial Expression Recognition from Video Sequences: Temporal and Static Modeling".

[6] Albert Cruz, Bir Bhanu and Ninad Thakoor." Facial Emotion Recognition in Continuous Video" 21st International Conference on Pattern Recognition (ICPR 2012) November 11-15, 2012. Tsukuba, Japan.

[7] L. S. Chen. "Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction."PhD thesis, University of Illinois at Urbana-Champaign, Dept. of Electrical Engineering, 2000.

[8] Dilbag Singh "Human Emotion Recognition System" I.J. Image, Graphics and Signal Processing, 2012, 8, 50-56 Published Online August 2012 in MECS.

[9] V.V. Starovoitov, D.I Samal, D.V. Briliuk "Three Approaches For Face Recognition" The 6-th International Conference on Pattern Recognition and Image Analysis October 21-26, 2002, Velikiy Novgorod, Russia, pp. 707-711.

[10] Kwang-Eun Ko, Kwee-Bo Sim "Development of a Facial Emotion Recognition Method based oncombining AAM with DBN" 2010 International Conference on Cyberworlds.

[11] Mohammad Ibrahim Khan and Md. Al-Amin Bhuiyan "Facial Expression Recognition for Human-Robot Interface" IJCSN, VOL.9 No.4, April 2009.

[12] Mayur S. Burange, S. V. Dhopte "Neuro Fuzzy Model for Human Face Expression Recognition" IJEAT ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012.