International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.06, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

ENHANCED EMOTION BASED ANALYSIS FOR STRESS ASSESSMENT AND MONITORING AT WORKPLACE

N.Selvarani.,M.E(Assistant Professor)[1], K.Priyadharsini[2], M.Aishwarya[3], G.Rama[4], J.Selvapriya[5]

[1,2,3,4,5] *Computer science and Engineering, A.V.C College of Engineering.(India)*

ABSTRACT

Facial expression is an effective way for humans to communicate since it contains critical and necessary information regarding human affective states. It is a critical part of affective computing systems that aim to recognize and therefore better respond to human emotions. Automatic recognition of facial expressions can be an important component in human-machine interfaces, human emotion analysis, and decision making. However, the task of automatically recognizing various facial expressions challenging. As a result, facial expression recognition has become a prominent research topic in human-computer interaction, as well as in the fields of image processing, pattern recognition, machine learning, and human recognition. In this paper, we will implement the techniques to automatically detect facial parts using HAAR CASCADES algorithm and classify the emotions using Support Vector Machine algorithm. And present playlist of games which is suitable for his current mood using K-Nearest Neighbor classification algorithm. In testing side, would supply a test image whose expression it desires to recognize. This test image would be matched with facial databases to play game based on recognized emotions. Finally provide emotion based game player with improved recognition rate.

Keywords—Emotion Recognition, Human-computer interaction, Pattern Recognition, K-Nearest Neighbor Algorithm

I INTRODUCTION

Affective Computing is computing that relates to, arises from, or deliberately influences emotion or other affective phenomena.Emotion is fundamental to human experience, influencing cognition, perception, and everyday tasks such as learning, communication, and even rational decision-making. However, technologists have largely ignored emotion and created an often frustrating experience for people, in part because affect has been misunderstood and hard to measure. Our research develops new technologies and theories that advance basic understanding of affect and its role in human experience. We aim to restore a proper balance between emotion and cognition in the design of technologies for addressing human needs.

Our research has contributed to: (1) Designing new ways for people to communicate affective-cognitive states, especially through creation of novel wearable sensors and new machine learning algorithms that jointly analyze multimodal channels of information; (2) Creating new techniques to assess frustration, stress, and mood indirectly, through natural interaction and conversation; (3) Showing how computers can be more emotionally intelligent, especially responding to a person's frustration in a way that reduces negative feelings; (4) Inventing personal technologies for improving self-awareness of affective state and its selective communication to others; (5) Increasing understanding of how affect influences personal health; and (6) Pioneering studies examining ethical issues in affective computing. Affective Computing research combines engineering and computer science with psychology, cognitive science, neuroscience, sociology, education, psychophysiology, value-centered design, ethics, and more. We bring together individuals with a diversity of technical, artistic, and human abilities in a collaborative spirit to push the boundaries of what can be achieved to improve human affective experience with technology**.** Detecting emotional information begins with passive sensors which capture data about the user's physical state or behavior without interpreting the input. The data gathered is analogous to the cues humans use to perceive emotions in others. For example, a video camera might capture facial expressions, body posture and gestures, while a microphone might capture speech. Other sensors detect emotional cues by directly measuring physiological data, such as skin temperature and galvanic resistance. Recognizing emotional information requires the extraction of meaningful patterns from the gathered data. This is done using machine learning techniques that process different modalities, such as speech recognition, natural language processing, or facial expression detection, and produce either labels or coordinates in a valence-arousal space.The detection and processing of facial expression is achieved through various methods such as optical flow, hidden Markov model, neural network processing or active appearance model. More than one modalities can be combined or fused to provide a more robust estimation of the subject's emotional state.Another area within affective computing is the design of computational devices proposed to exhibit either innate emotional capabilities or that are capable of convincingly simulating emotions. A more practical approach, based on current technological capabilities, is the simulation of emotions in conversational agents in order to enrich and facilitate interactivity between human and machine. While human emotions are often associated with surges in hormones and other neuropeptides, emotions in machines might be associated with abstract states associated with progress in autonomous learning systems. In this view, affective emotional states correspond to time-derivatives (perturbations) in the learning curve of an arbitrary learning system. The basic human interaction is shown in fig 1.

IJARSE
ISSN: 2319-8354

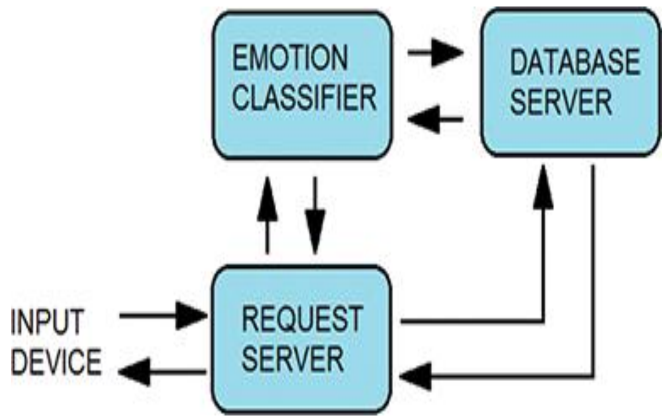

Fig 1: Emotion Recognition

II.RELATED WORK

M. Liu, et.al,.. [1] attempted to take both issues into account via spatio-temporal manifold modeling based on a set of mid-level representations, i.e. expressionlets. The proposed mid-level expressionlet is a kind of modeling that aims to characterize the variations among a group of low level features. The notation "-let" means that it serves as a local (both spatially and temporally) dynamic component within a whole expression process, which shares similar spirit with "motionlet" in action recognition community. Thus expressionlet bridges the gap between low-level features and high-level semantics desirably. Specifically, given an individual video clip, we first model it as a Spatio-Temporal Manifold (STM) spanned by its low-level features. To conduct spatio-temporal alignment among STMs, we build a Universal Manifold Model (UMM), represented as a number of universal local ST modes, which can be learned by EM-like methods on all possible low-level features. By fitting to UMM, the local modes on each STM can be instantiated respectively and all of the different STMs are inherently well-aligned to UMM via these corresponding modes. Finally, our expressionlet is constructed by modeling each local mode on STM. To characterize the correlations and variations among low-level features within each mode, the expressionlet is represented as the covariance matrix of the feature set in a statistic manner, which also makes it robust to local misalignment. To further enhance the discriminative ability of expressionlet, we perform discriminant learning with these midlevel representations on all of the STMs. Inspired by, while only considering the "margin" among corresponding expressionlets; we exploit a graph-embedding method by constructing partially connected graphs to keep the links between expressionlets with the same semantics. In the end, the embedded features are correspondingly concatenated into a long vector as the final manifold (video) representation for classification.

H. Jung, et.al,…[2] interested in recognizing facial expressions using a limited amount of (typically a few hundreds of) image sequence data with a deep network. In order to overcome the problem of having a small amount of data, we construct two small deep networks that complement each other. One of the deep networks is trained using image sequences, while the other deep network learns the temporal trajectories of facial landmark points. In other words, the first network focuses more on appearance changes of facial expressions over time, while the second network is directly related to the motion of facial parts. Furthermore, we present a new integration method called joint fine-tuning, which performs better than simple weighted summation method. Two deep network models are presented in order to extract useful temporal representations from two kinds of sequential data: image sequences and the trajectories of landmark points. We observed that the two networks automatically detect moving facial parts and action points, respectively. We presented a joint fine-tuning method integrating these two networks with different characteristics, and performance improvement was achieved in terms of the recognition rates. In general, the length of image sequences is variable, but the input dimension is usually fixed in a deep network. Consequently, the normalization along the time axis is required as input for the networks, which makes an image sequence into a fixed length. Then, the faces in the input image sequences are detected, cropped, and rescaled to 64×64. From these detected faces, facial landmark points are extracted using the algorithm called IntraFace. This algorithm provides accurate facial landmark points consisting of 49 landmark points, including two eyes, a nose, a mouth, and two eyebrows. In this paper, a CNN is used for capturing temporal changes of appearance. Conventional CNN uses still images as input, and 3D CNN was presented recently for dealing with image sequences. As mentioned, the 3D CNN method shares the 3D filters along the time axis. However, we use the n-image sequences without weight sharing along the time axis. This means that each filter plays a different role depending on the time.

Y. Sun, et.al,..[3] proposed a cascaded regression approach for facial point detection with three levels of convolutional networks. Different from existing approaches which roughly estimate the initial positions of facial points, our convolutional networks make accurate predictions at the first level, even on very challenging cases. It effectively avoids the local minimum problem faced by other approaches. The convolutional networks take the full face as input to make the best use of texture context information, and extract global high-level features at higher layers of the deep structures, which can effectively predict key points even when low-level features from local regions are ambiguous or corrupted in challenging image examples. Our convolutional networks are trained to predict all the keypoints simultaneously and the constraints of keypoints are implicitly encoded. The remaining two levels of convolutional networks refine the initial estimation of keypoints. Different from existing methods which apply the same regressor at different cascade stages, we design different convolutional networks. The network structures at these two levels are shallower, since their tasks are low-level and their input is limited to small local regions around the initial positions. At each level, multiple convolutional networks are fused to improve the accuracy and reliability of estimation. Through detailed empirical investigation, we find that several factors regarding the network structures are critical for achieving good performance in facial point detection. Detailed experimental evaluations show that our approach outperforms state-of-the-art methods on both accuracies and reliability.

L. Zhong, et.al,..[4] provided a solid validation for an important psychology discovery, that only a few facial muscles (areas) are discriminative for expression recognition. A two-stage multi-task sparse learning framework is proposed to formulate the commonalities among expressions, and find out the locations of common and specific patches for expressions. Extensive experiments on two public databases demonstrate that these active patches are effective in recognizing expressions, and can be utilized to further improve the performances of state-of-the-arts. Facial expressions are usually manifested by local facial appearance variations. However, it is not easy to automatically localize these local active areas on a facial image. A facial image is divided into p local patches, and then local binary pattern (LBP) features are used to represent the local appearance of the patch. These features have been proven to be a powerful descriptor in expression recognition and face verification. For each patch, the uniform LBP features are extracted with the LBP operator LBP and mapped to a m-dimensional histogram (m = 59 in our paper). Based on these local patches, the common patches across all expressions are learned for expression recognition. Then, some specific patches for each expression are explored to further enhance the performance. Discovering the common patches across all the expressions is actually equivalent to learning the shared discriminative patches for all the expressions. Since Multi-task sparse learning (MTSL) can learn common representations among multiple related tasks, our problem can be transferred into a MTSL problem. Although learned common patches can discriminate all facial expressions, the performance could not be the best, because each expression also has its special properties besides the common properties. Here, we aim to explore some specific facial patches for each expression with the help of face verification, and then they are used to further boost the performance of common facial patches. The motivation to employ the face verification task is that those special facial patches are important face regions, which are not only useful for recognizing this expression, but also very significant for identifying the subjects.

X. Xiong, et.al,…[5]addressed previous limitations, this paper proposes a Supervised Descent Method (SDM) that learns the descent directions in a supervised manner. The top image shows the application of Newton's method to a Nonlinear Least Squares (NLS) problem, where *f*(x) is a nonlinear function and y is a known vector. In this case, *f*(x) is a non-linear function of image features (*e.g*., SIFT) and y is a known vector (i.e., template). x represents the vector of motion parameters (i.e., rotation, scale, non-rigid motion). The traditional Newton update has to compute the Hessian and the Jacobian ,the main idea behind SDM. The training data consists of a set of functions sampled at different locations yi (i.e., different people) where the minima are known. Using this training data, SDM learns a series of parameter updates, which incrementally, minimizes the mean of all NLS functions in training. In the case of NLS, such updates can be decomposed into two parts: a sample specific component (e.g., yi) and a generic descent directions Rk. SDM learns average descent directions Rk during training. In testing, given an unseen y, an update is generated by projecting specific components onto the learned generic directions Rk. We illustrate the benefits of SDM on analytic functions, and in the problem of facial feature detection and tracking. We show how SDM improves state-of-the-art performance for facial feature detection in two "face in the wild" databases and demonstrate extremely good performance tracking faces in the YouTube celebrity database .Although they have shown improvements using those re-sampled features, feature re-generation in regression are not well understood and invalidates some properties of gradient boosting. In SDM, the linear regressor and feature re-generation come up naturally in our derivation from Newton method. That a Newton update can be expressed as a linear combination of the feature differences between the one extracted at current landmark locations and the template. In previous work, it was unclear what the alignment error function is for discriminative methods. This work proposes, which is the error function minimized by discriminative methods, and connect it with PAMs.

III.EXISTING METHODOLOGIES

There are numerous areas in human–computer interaction that could effectively use the capability to understand emotion. The problem of face detection can be viewed as a problem of binary classification of image frame as either containing or not containing a face. In order to be able to learn such a classification model, we first need to describe an image in terms of features, which would be good indicators of face presence or absence on a given image. The existing approach is generally involves two tasks: The first is for extracting ASM motion based a pyramid ASM model fitting method and the second for the projected motion classification obtained by applying Adaboost classifiers. After the segmentation of face candidates, 68 feature points in each face are then extracted using ASM fitting technique. The system then line up three extracted feature points, eyes and nose part, to the mean shape of ASM, and ignore the other portion of the ASM against the mean face shape of ASM to estimate the geometrical dislocation information between current and mean ASM points coordinates. Then, facial expressions recognition is the obtained based on this geometrical motion using Adaboost classifier. And also extracting features using viola jones. The features that Viola and Jones used are based on wavelets. Wavelets are single wavelength square waves (one high interval and one low interval). In two dimensions, a square wave is a pair of adjacent rectangles - one light and one dark.

IV.PROPOSED METHODOLOGIES

In this paper, propose a novel emotion recognition system based on the processing of physiological signals is presented. This system shows a recognition ratio much higher than chance probability, when applied to physiological signal databases obtained from tens to hundreds of subjects. The system consists of characteristic face detection, feature extraction and pattern classification stages. Although the face detection and feature extraction stages were designed carefully, there was a large amount of within-class variation of features and overlap among classes. In order to detect Emotion from an image, used frontal view facial images. If computers can understand more of human emotion, we can make better systems to reduce the gap of human computer interaction.To handle the emotion recognition problem from arbitrary view facial images. The facial region and others part of the body have been segmented from the complex environment based on skin color model. Thus, in this paper showed some differences between different color models that are used to implement the system and which color model can be used where. Another aspect is to extract facial parts from the face. And for that used HAAR CASHCADES to detect the eye and lips region from a face and then by the help of SVM classification detected emotion from those features. From the positioning of mouth and eyes, tried to detect emotion of a face. The proposed system tries to provide an interactive way for the user to carry out the task of creating a playlist. The working is based on KNN mechanisms carrying out their function in a pre-defined order to get the desired output. The classified expression acts as an input and is used to select an appropriate playlist from the initially generated player and the games from the database are played. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate. And draw the bounding box and also calculate distance measurement from web cameras.

HAAR-LIKE Features:

Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector. Historically, working with only image intensities (i.e., the RGB pixel values at each and every pixel of image) made the task of feature calculation computationally expensive. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case). The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of integral images, a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).

Rectangular HAAR like features:

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. This modified feature set is called 2-rectangle feature. Viola and Jones also defined 3-rectangle features and 4-rectangle features. The values indicate certain characteristics of a particular area of the image. Each feature type can indicate the existence (or absence) of certain characteristics in the image, such as edges or changes in texture.

CASCADE Classifier:

The cascade classifier consists of a list of stages, where each stage consists of a list of weak learners. The system detects objects in question by moving a window over the image. Each stage of the classifier labels the specific region defined by the current location of the window as either positive or negative – positive meaning that an object was found or negative means that the specified object was not found in the image. If the labeling yields a negative result, then the classification of this specific region is hereby complete and the location of the window is moved to the next location. If the labeling gives a positive result, then the region moves of to the next stage of classification. The classifier yields a final verdict of positive, when all the stages, including the last one, yield a result, saying that the object is found in the image. A true positive means that the object in question is indeed in the image and the classifier labels it as such – a positive result. A false positive means that the labeling process falsely determines that the object is located in the image, although it is not. A false negative occurs when the classifier is unable to detect the actual object from the image and a true negative means that a non-object was correctly classifier as not being the object in question. In order to work well, each stage of the cascade must have a low false negative rate, because if the actual object is classified as a non-object, then the classification of that branch stops, with no way to correct the mistake made. However, each stage can have a relatively high false positive rate, because even if the n-th stage classifies the non-object as actually being the object, then this mistake can be fixed in n+1-th and subsequent stages of the classifier.

The calculation steps as follows

- Feature = $w_1 * \mathrm{RecSum}(r_1) + w_2 * \mathrm{RecSum}(r_2)$
- Weights can be positive or negative
- Weights are directly proportional to the area
- Calculated at every point and scale

It includes weak classifier such as

A weak classifier ($h(x,f,p,\vartheta)$) consists of

-feature(f)

-threshold(ϑ)

-polarity(p), such that

$$h(x, f, p, \vartheta) = \begin{cases} 1 & \text{ifpf}(x) < p\vartheta \\ 0 & \text{otherwise} \end{cases}$$

Requirement

- Should perform better than the random chance

Cascade creation algorithm as follows:

$f_0 = 1$

i=0

while $f_i > f_{target}$ and i<nStages

i=i=1

Train classifier for stage i

Initialize the weights

Normalize the weights

Pick the (next) best weak classifier

Update weights

Evaluate f_i

if $f_i > f$

go back from normalize the weights

Combine weak classifiers to form the strong stage classifier

Evaluate F_i

The Haar features are extracted using detector windows of various sizes. The cascading of classifier allows highest probability sub images to analyze for all haar features to distinguish an object. The number of features evaluated when scanning real images is necessarily a probabilistic process. Any given sub-window will progress down through the cascade, one classifier at a time, until it is decided that the window is negative or, in rare circumstances, the window succeeds in each test and is labeled positive. The expected behavior of this process is determined by the distribution of image windows in a typical test set. The key measure of each classifier is its positive rate, the proportion of windows which are labeled as potentially containing the object of interest. This cascaded classifier evaluates the input image and check for the presence of a rectangular window in a given image. Here each window is subdivided into several sub windows contributes for the presence of face or non face when passing it over several stages of classifiers. Each classifier evaluates the input window for the facial features which contributes for the face.

SVM classifier:

Support Vector Machines (SVMs) view the classification problem as a quadratic optimization problem. The technique has successfully been applied to standard classification tasks, such as text classification and medical diagnosis. SVMs avoid the "curse of dimensionality" by placing an upper bound on the margin between the different classes, making it a practical tool for large, dynamic datasets. The feature space may even be reduced further by selecting the most distinguishing features through minimization of the feature set size. SVMs plot the training vectors in high-dimensional feature space, and label each vector with its class. A hyperplane is drawn between the training vectors that maximize the distance between the different classes. The hyperplane is determined through a kernel function, which is given as input to the classification software. The kernel function may be linear, polynomial, radial basis, or sigmoid. The shape of the hyperplane is generated by the kernel function, though many experiments select the polynomial kernel as optimal. SVMs provide a generic mechanism to fit the surface of the hyperplane to the data through the use of a kernel function. The user may provide a function, such as a line, polynomial, or sigmoid curve, to the SVM, which selects support vectors along the surface of this function. This capability allows a broader range of problems to be classified, since the user may input any function, customized to a specific dataset. In the case of linearly inseparable datasets, the cost of misclassification is accepted through the use of 'slack variables'. The input space is planned into a high dimensional feature space. Then, the hyper plane that exploits the margin of separation between classes is constructed. The points that lie closest to the decision surface are called support vectors directly involves its location. When the classes are non-separable, the optimal hyper plane is the one that minimizes the probability of classification error. Initially input image is formulated in feature vectors. Then these feature vectors mapped with the help of kernel function in the feature space. And finally division is computed in the feature space to separate out the classes for training data. A global hyper plane is required by the SVM in order to divide both the program of examples in training set and avoid over fitting. This phenomenon of SVM is higher in comparison to other machine learning techniques which are based on artificial intelligence. Here the important feature for the classification is the width of the vessels. With the help of SVM classifier we can easily separate out the vessels into arteries and veins. The SVMs demonstrate various attractive features such as good generalization ability compared to other classifiers. Indeed, there are relatively few free parameters to adjust and it is not required to find the architecture experimentally. The SVMs algorithm separates the classes of input patterns with the maximal margin hyper plane. This hyper plane is constructed as:

$$f(x) = \langle w, x \rangle + b$$

Where x is the feature vector, w is the vector that is perpendicular to the hyper plane and $b\|w\|^{-1}$ specifies the offset from the beginning of the coordinate system. To benefit from non-linear decision boundaries the separation is performed in a feature space F, which is introduced by a nonlinear mapping φ the input patterns. This mapping is defined as follows:

$$\langle \varphi\,(x_1), \varphi(x_2) \rangle = K(x_1, x_2)\ \forall (x_1, x_2) \in X$$

for some kernel function K (·, ·). The kernel function represents the non-linear transformation of the original feature space into the F.

KNN classification:

In game recognition, the KNN algorithm is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The KNN is the fundamental and simplest classification technique when there is little or no prior knowledge about the distribution of the data. This rule simply retains the entire training set during learning and assigns to each query a class represented by the majority label of its k-nearest neighbors in the training set. The Nearest Neighbor rule (NN) is the simplest form of KNN when K = 1. In this method each sample should be classified similarly to its surrounding samples. Therefore, if the classification of a sample is unknown, then it could be predicted by considering the classification of its nearest neighbor samples. Therefore, the unknown sample may be classified based on the classification of this nearest neighbor. The algorithm steps as follows:

for all the unknown samples UnSample(i)

for all the known samples Sample(j)

compute the distance between

IJARSE
ISSN: 2319-8354

Unsamples(i) and Sample(j)

end for

fing the k smallest distances

locate the corresponding samples

Sample(j1),….,Sample(jK)

assign UnSample(i) to the class which appears more frequently

end for. The proposed framework is shown in fig 2.

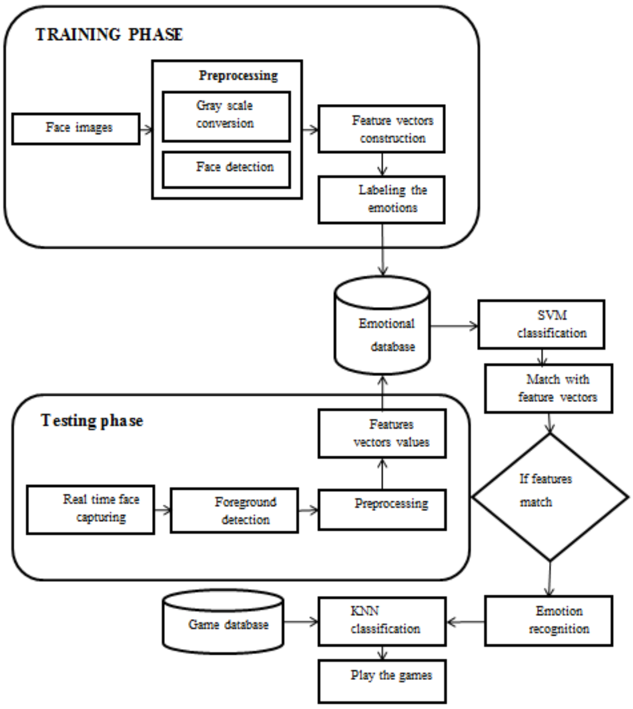

Fig 2: Proposed work

V.CONCLUSION

In this paper proposed support vector machine algorithm for emotion recognition. Considering an expressive face as a superposition of a neutral face with expression component, we proposed an algorithm to decompose an expressive test face into its building components. For this purpose, we first generate grids for captured face using HAAR Cascade algorithm. Knowing that the face component of the test face has sparse representation in the face database and the expression part can be sparsely represented using the expression database; we decompose the test face into these feature vectors. The elements of the test face along with the vectors are then used for face and expression recognition. For this purpose, the separated components are sparsely decomposed using vectors while the grouping structures of the vectors are enforced into the sparse decomposition. The experimental results on both databases showed that the proposed method achieves competitive recognition performance compared with the state of the art methods under same experimental settings and same facial feature. Based on their emotions, play the games to recover from depression. In this project we can be implemented the system to using image processing techniques to detect the faces from camera capturing. Then efficiently track the faces and to provide bounding boxes on face images. Finally set the distance limits to identify whether the person is near to the system or not. And also calculated the person constant seeing conditions. This system can be useful to all aged peoples in various applications such as gaming applications, project works and so on.

REFERENCES

[1] M. Liu, S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014.

[2] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognitin," in The IEEE International Conference on Computer Vision (ICCV), IEEE, 2015.

[3] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013.

[4] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012.

[5] Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013.

[6] A. Graves et al., Supervised sequence labelling with recurrent neural networks, vol. 385. Springer, 2012.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NIPS), 2012.

[8] Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014.

[10] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction.," in The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2003