An Intelligent Music Player based on Emotion Recognition

Ramya Ramanathan*, Radha Kumaran†, Ram Rohan R‡, Rajat Gupta§ and Vishalakshi Prabhu¶
Department of Computer Science and Engineering, R V College of Engineering
Bangalore, India
Email: *ramya.ramanathan28@gmail.com, †kumaranradha08@gmail.com, ‡rohan@brainsnskills.com, §rgupta18101996@gmail.com,
¶vishalaprabhu@rvce.edu.in

***Abstract*—This paper proposes an intelligent agent that sorts a music collection based on the emotions conveyed by each song, and then suggests an appropriate playlist to the user based on his/her current mood. The user's local music collection is initially clustered based on the emotion the song conveys, i.e. the mood of the song. This is calculated taking into consideration the lyrics of the song, as well as the melody. Every time the user wishes to generate a mood-based playlist, the user takes a picture of themselves at that instant. This image is subjected to facial detection and emotion recognition techniques, recognizing the emotion of the user. The music that best matches this emotion is then recommended to the user as a playlist.**

***Index Terms*—music classification, emotion recognition, intelligent music player**

I. Introduction

Emotion recognition is an aspect of artificial intelligence that is becoming increasingly relevant, for the purpose of automating various processes that are relatively more tedious to perform manually. Identifying a person's state of mind based on emotions they display is an important part of making efficient automated decisions best suited to the person in question, for a variety of applications.

One important aspect of this would be in the entertainment field, for the purpose of providing recommendations to a person based on their current mood. We study this from the perspective of providing a person with customised music recommendations based on their state of mind, as detected from their facial expressions.

Most music connoisseurs have extensive music collections that are often sorted only based on parameters such as artist, album, genre and no. of times played. However, this often leaves the users with the arduous task of making mood based playlists, i.e. sorting the music based on the emotion conveyed by the songs - something that is much more essential to the listening experience than it often appears to be. This task only increases in complexity with larger music collections, and automating the process would save many a user the effort spent in doing the same manually, while improving their overall experience and allowing for a better enjoyment of the music.

II. Related Work

The first part of this system involves human emotion recognition, as research has found that the largest part of any message is conveyed through facial expressions. Thus, facial emotion recognition has been extensively researched, in order to find optimal techniques recognition. The emotion of the person is identified from an image of them, using various image processing and facial recognition techniques, along with learning techniques for correlating the facial expression with an emotion. The learning can be achieved by using ANNs and HMMs [1] for clustering and classification of the emotions, which can be trained on available datasets of faces such as the Cohn Kanade dataset.

The next part of the system involves classifying music and labelling each piece based on the emotion expressed by the music. There has been research in the field of music emotion recognition, starting from generating appropriate datasets, to feature extraction from each song [2] based on the best features to extract in order to obtain most accurate results.

Determination of the emotion expressed by the audio piece has been performed based on the Arousal-Valence model of emotion proposed by Thayer [3]. This model suggests that the emotion of a song is accurately expressed as a combination of two factors - the arousal, or energy of the song, and the valence, or stress of the same piece. A high valence indicates a positive emotion, while a low value represents a negative vibe.

All the music, thus, is divided over four quadrants, with emotions ranging from calm to energetic on the arousal scale and from negative to positive on the valence scale. A number of similar emotions are represented within each quadrant, as shown in the figure. Sadness, for example, would be represented by a low valence and low arousal. The descriptors that have been found using feature extraction can be used to classify the music by clustering, after they have been assigned appropriate arousal and valence values [4].

The goal of our project is to combine these two features, and allow for a user to be provided with customised music recommendations, based on their emotion as detected from an image taken of them at the same instant. The music database will then be queried to find all the music in the collection that

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

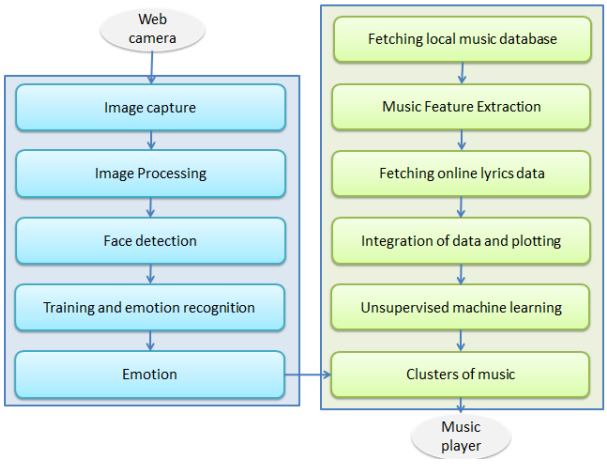


Figure 1: Design of the system

III. Design

The system uses a video capture object in order to access the web camera of the computer being used. Multiple images are captured from web camera. To predict the emotion accurately, we might want to have more than one facial image. Blurred images can be an error source (especially in low light conditions) and hence, the multiple images are averaged to get an image devoid of any blur. Histogram equalization is an image processing technique used to enhance the contrast of the image by normalizing the image throughout its range. This image is then cropped and converted to grey scale so that only the foreground of the image remains, thereby reducing any ambiguity.

A Haar classifier is used for face detection where the classifier is trained with pre-defined varying face data which enables it to detect different faces accurately. A Viola- Jones Haar cascade classifier [5] is used for face detection. In the ViolaJones object detection framework, the Haar-like features are therefore organized in something called a classifier cascade to form a strong learner or classifier. The key advantage of a Haarlike feature over most other features is its calculation speed. For this we use the fisherfaceclassifier package [13] in python. A feature is provided to customize this model by taking multiple pictures of a specific person. This helps to reduce the variance in the pictures. Hence, any further change detected would be in the emotion of the user. Finally, the emotion is detected by a generalized model which can be customized by using argparse in order to update the training set for emotion as per the user requirements.

For the music emotion recognition, the music dataset needs to first be obtained. Initially, a small dataset of songs from a local collection is used, after which a subset of the Million Song Dataset [6] can be used for further testing and training. The music lyrics extraction has been done with the help of the musiXmatch dataset [7]. The music feature extraction can be performed in python by using MIR packages such as LibROSA [8] and PyAudioAnalysis [9]. These are python packages that aid the analysis and retrieval of musical features such as tempo, rhythm, timbre, etc. which, along with the lyrics, are further used for clustering by unsupervised learning, using the k-means algorithm. For efficient execution of this technique, the centroids are manually set in order to select the locations of the clusters. Once this has been accomplished, each song must be labelled with appropriate descriptors, to make the search process more efficient once the facial emotion recognition has been completed. The music database can then be queried using appropriate descriptors, to obtain the desired music playlist automatically.

IV. Methodology

A. Audio Feature Extraction and plotting on Thayer's Graph

One component of the system that has been implemented is one that retrieves music that has been locally stored on a system, and processes this music with the aid of various Python packages for Music Information Retrieval. These packages are are available for use freely, and prove essential to the process in the absence of software such as MATLAB. Once musical feature extraction has been completed, the music collection can be clustered using any efficient algorithm.

Classification of the music begins with retrieving all the music files locally available on the system, through the music database. The location of all the *.mp3* files can be retrieved from the database, and then input to the *glob* package in python, which recursively finds all the music files in that location. This can further be used to automatically update the database as and when new music is added to the collection.

Once all the music files have been found, the process of feature extraction can be begun. This starts with loading the audio waveform, generally represented as y, and storing the sample rate in sr. The audio features to be extracted can be classified as contributing to either the arousal or the valence calculation. Arousal typically represents the energy of the song, while the valence is a measure of the stress of the song. For the purpose of calculating arousal of the song, the tempo and dynamics of the song are calculated. Tempo can be retrieved using the *beat_track* method offered by LibROSA which returns an approximation of the tempo of the song, along with the beat frames, in case visualization is desired. Calculation of the average dynamics of the song requires the *rmse* feature offered by LibROSA, which returns the root-mean-square energy for each frame, from the audio sample y. This RMS energy can be averaged over the entire song as a root-mean-square to give a measure of the dynamic of the song. Both of these features, viz. tempo and dynamics, are scaled to values between 0 and 1, to ensure more accurate representation on the arousal-valence graph.

The valence of each song has been found by estimating the pitch class that the song belongs to, out of the twelve pitch classes (where each pitch class represents all pitches that map to a C, for example). Initially, the Short Time Fourier Transform of the audio waveform is calculated, and a Discrete Cosine Transform for each window is found. This eventually returns a chromagram, which is a set of normalised energy

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

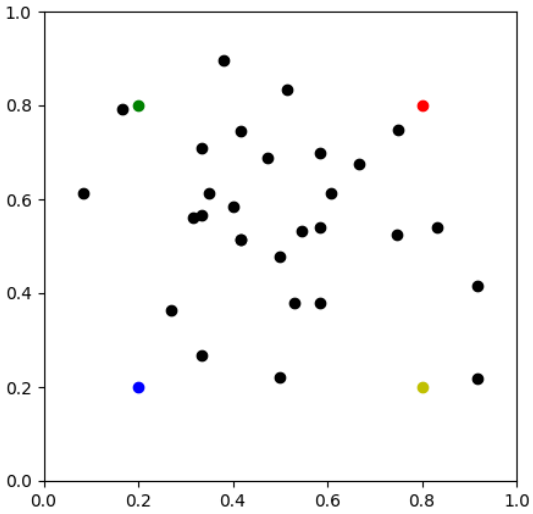


Figure 2: Setting of Centroids

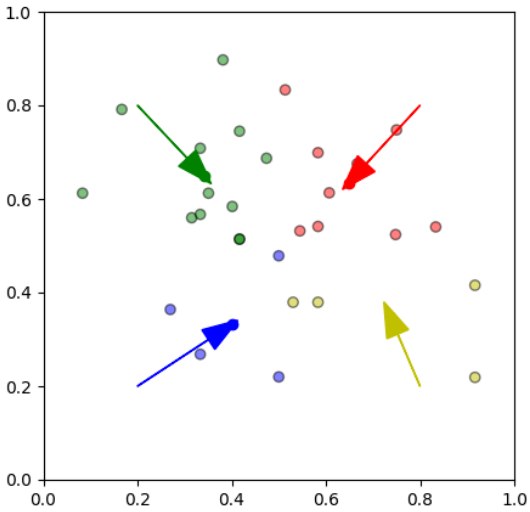


Figure 3: k-means clustering

estimates for each of the twelve chroma bins. These values can be used to determine which of the twelve pitch classes the song primarily falls in. The higher the pitch class, the higher the valence of the song is estimated to be.

The valence value is also independently calculated from the lyrics of each song. The musiXmatch dataset, the largest dataset for lyrics is accessed in order to get the lyrics. A function by the name sentiment.polarity provided by the package TextBlob returns the valence of the song solely from the lyrics. This too is reduced to a scale of 0 to 1.

The valence values are independently calculated from both the lyrics and the melody in order to allow the user to decide how important the lyrics and the melody is to him/her. Based on the user's inputs, a final valence value for each song in calculated giving weights to the initial valence values.

B. Music Clustering

Once valence and arousal values (on a scale of 0 to 1) have been found for all songs in the collection, these can be plot on Thayers arousal-valence graph. This graph, with all the data points on it, is subjected to k-means clustering, to obtain clusters of music based on their emotion. The objective of k-means is to cluster all the data points into k clusters in such a way that the sum of distances from each data point to the mean of its respective cluster is minimized. Initially, the centroids of each cluster are set manually such that each of the emotions that can be returned from the image captured by the web camera is associated with a unique cluster.

Upon clustering, the music collection can be classified, with emotion descriptors for each song. These descriptors are used to then select music from the collection to be played, based on the user's current mood. Therefore, the playlist that will be returned will have all the songs that belong to the specific

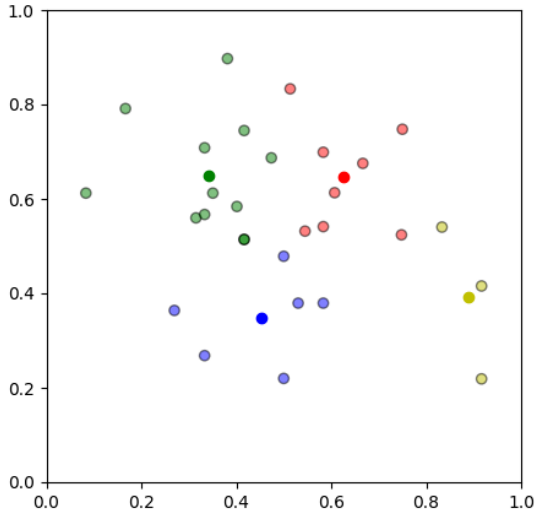


Figure 4: clustered dataset

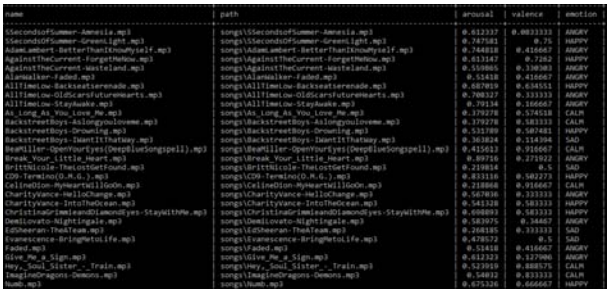
cluster associated with the emotion that is obtained from the image.

C. User Emotion Recognition

For this model, the Cohn Kanade extended dataset [11] is used. A python script is used to fetch images containing faces along with their emotional descriptor values. The images are contrast enhanced by contrast limiting adaptive histogram equalization and converted to grayscale in order to maintain uniformity and increase effectiveness of the classifiers.

A cascade classifier [5], trained with face images is used for face detection where the image is split into fixed-size windows. Each window is passed to the cascade classifiers and is accepted if it is passes through all the classifiers, otherwise

978-1-5386-2044-1/17/$31.00 ©2017 IEEE



name	path	arousal	valence	emotion
5SecondsofSummer-Amnesia.mp3	songs\5SecondsofSummer-Amnesia.mp3	0.612337	0.0833333	ANGRY
5SecondsOfSummer-GreenLight.mp3	songs\5SecondsOfSummer-GreenLight.mp3	0.747581	0.75	HAPPY
AdamLambert-BetterThanIKnowMyself.mp3	songs\AdamLambert-BetterThanIKnowMyself.mp3	0.744618	0.416667	ANGRY
AgainstTheCurrent-ForgetMeNow.mp3	songs\AgainstTheCurrent-ForgetMeNow.mp3	0.613147	0.7262	HAPPY
AgainstTheCurrent-Wasteland.mp3	songs\AgainstTheCurrent-Wasteland.mp3	0.559865	0.330303	ANGRY
AlanWalker-Faded.mp3	songs\AlanWalker-Faded.mp3	0.51418	0.416667	ANGRY
AllTimeLow-Backseaterenade.mp3	songs\AllTimeLow-Backseaterenade.mp3	0.687019	0.634551	HAPPY
AllTimeLow-OldScarsFutureHearts.mp3	songs\AllTimeLow-OldScarsFutureHearts.mp3	0.708327	0.333333	ANGRY
AllTimeLow-StayAwake.mp3	songs\AllTimeLow-StayAwake.mp3	0.79134	0.166667	ANGRY
As_Long_As_You_Love_Me.mp3	songs\As_Long_As_You_Love_Me.mp3	0.379278	0.574518	CALM
BackstreetBoys-Aslongyouloveme.mp3	songs\BackstreetBoys-Aslongyouloveme.mp3	0.379278	0.583333	CALM
BackstreetBoys-Drowning.mp3	songs\BackstreetBoys-Drowning.mp3	0.531789	0.507481	HAPPY
BackstreetBoys-IWantItThatWay.mp3	songs\BackstreetBoys-IWantItThatWay.mp3	0.363824	0.114394	SAD
BeaMiller-OpenYourEyes(DeepBlueSongspell).mp3	songs\BeaMiller-OpenYourEyes(DeepBlueSongspell).mp3	0.415613	0.916667	CALM
Break_Your_Little_Heart.mp3	songs\Break_Your_Little_Heart.mp3	0.89716	0.271922	ANGRY
BrittNicole-TheLostGetFound.mp3	songs\BrittNicole-TheLostGetFound.mp3	0.219814	0.5	SAD
CDO-Termino(O.M.G.).mp3	songs\CDO-Termino(O.M.G.).mp3	0.833116	0.502273	HAPPY
CelineDion-MyHeartWillGoOn.mp3	songs\CelineDion-MyHeartWillGoOn.mp3	0.218868	0.916667	CALM
CharityVance-HelloChange.mp3	songs\CharityVance-HelloChange.mp3	0.567036	0.333333	ANGRY
CharityVance-IntoTheOcean.mp3	songs\CharityVance-IntoTheOcean.mp3	0.541328	0.583333	HAPPY
ChristinaGrimmieandDiamondEyes-StayWithMe.mp3	songs\ChristinaGrimmieandDiamondEyes-StayWithMe.mp3	0.698893	0.583333	HAPPY
Demilovato-Nightingale.mp3	songs\Demilovato-Nightingale.mp3	0.583975	0.34667	ANGRY
EdSheeran-TheATeam.mp3	songs\EdSheeran-TheATeam.mp3	0.268185	0.333333	SAD
Evanescence-BringMetoLife.mp3	songs\Evanescence-BringMetoLife.mp3	0.478572	0.5	SAD
Faded.mp3	songs\Faded.mp3	0.51418	0.416667	ANGRY
Give_Me_a_Sign.mp3	songs\Give_Me_a_Sign.mp3	0.812323	0.127906	ANGRY
Hey,_Soul_Sister_-_Train.mp3	songs\Hey,_Soul_Sister_-_Train.mp3	0.523919	0.888575	CALM
ImagineDragons-Demons.mp3	songs\ImagineDragons-Demons.mp3	0.58912	0.833333	CALM
Numb.mp3	songs\Numb.mp3	0.675326	0.666667	HAPPY

Figure 5: Music clustered and labelled

it is rejected. The detected faces are then used to train the face, which works on reduction of variance between classes. Fisher face recognition method proves to be efficient as it works better with additional features such as spectacles and facial hair. It is also relatively invariant to lighting.

A picture is taken during the runtime of the application, which after preprocessing, is predicted to belong to one of the emotion classes by the fisher face classifier. The model also permits the user to customize the model in order to reduce the variance within the classes further, initially or periodically, such that only variance would be that of emotion change.

D. Music Recommendation

The emotion detected from the image processing is given as input to the clusters of music to select a specific cluster. In order to avoid the interfacing with a music app or music modules which would involve extra installation, the support from the operating system is used instead to play the music file.

The playlist selected by the clusters are played by creating a subprocess, as the forked subprocess returns the control back to the python script on the completion of its execution, so that other songs can be played. This makes the program play music on any system regardless of its music player.

V. Experimental Analysis

Validating the quality of clustering is done based on the value of Silhouette Index and Silhouette Graph. The values for the Silhouette Graph are calculated for each object with respect to the cluster that the object belongs to. This graph gives an accurate qualification of the intra-cluster similarity and inter-cluster dissimilarity by taking Manhattan distance as a measure.

The Silhouette Graph obtained indicated very few objects whose Silhouette values fall in the negative range. Since negative values indicate poor clustering, the graph demonstrates the consistency in clustering.

The Silhouette score of the entire clustering is 0.317 which is an average of the score for every object.

Taking 20% of dataset created in the image expression detection as testing data and the rest as training data, the accuracy of the dataset for image emotion detection is 72.3%.

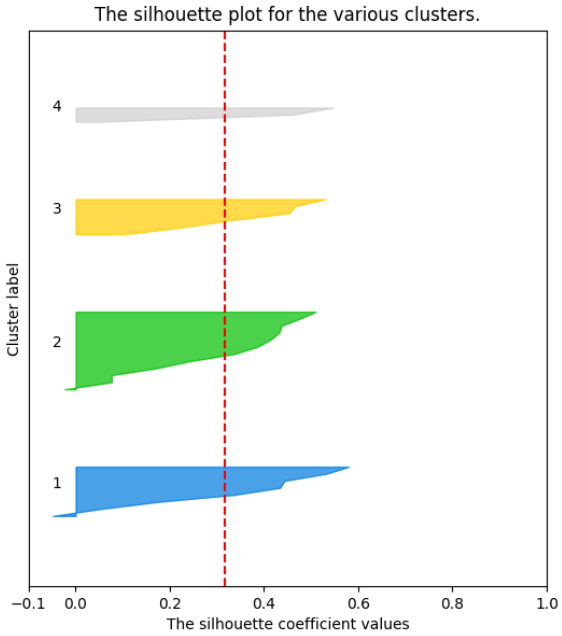


Figure 6: Silhouette Graph

Figure 7: Accuracy of image emotion recognition

VI. Future Work

This system, although completely functioning, does have scope for improvement in the future. There are various aspects of the application that can be modified to produce better results, and a smoother overall experience for the user. Some of these, that we hope to address in the near future, include having any number of emotions that could be returned independently of the number of clusters which can also vary, by plotting the emotion returned from the user's image based on the arousal and valence value, and finding the cluster whose mean is closest to the emotion returned than any other mean. This way, there are no restrictions due to the number of clusters, and there is no one-to-one mapping which will also ensure more accurate results.

Another addition would be automatic smart setting of the number of clusters based on the number of songs which will ensure that the number of songs per cluster is optimal. The silhouette index of the graph can be improved by increasing the number of emotions (K value in k-means) and can be

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

brought closer to 1 by reducing overlapping of clusters.

An alternate method that could be explored is elastic graph matching, which is the basic process of compare graphs with images to generate new graphs [12]. In its simplest version a single labeled graph is matched onto an image. A labeled graph has a set of jets arranged in a particular spatial order. A corresponding set of jets can be selected from the Gaborwavelet transform of the image. Elastic Bunch Graph Matching make use of Gabor features, being the output of band pass filters, and this are closely related to derivatives and are therefore less sensitive to lighting change. Also, this approach uses features only at a key node of the image rather than the whole image, this can reduce the noise taken from the background of the face images. It is also insensitive to facial positions.

Further study of the impact of various audio features on the general emotion of the song will be carried out, to ensure better and more accurate calculation of arousal and valence values by choosing the optimal combination of features extracted.

REFERENCES

[1] I. Cohen, A. Garg and T. S. Huang, *Emotion Recognition from Facial Expressions using Multilevel HMM* ,2015.

[2] Yading Song, Simon Dixon, and Marcus Pearce. *Evaluation of Musical Features for Emotion Classification.* In Proceedings of the 13th International Society for Music Information Retrieval Conference, Portugal. pp. 523-528. October 2012.

[3] Thayer, R. E. *The Biopsychology of Mood and Arousal.* Oxford University Press, USA, 1989.

[4] Nikalaou, N. *Music emotion classification.* PhD diss., Dissertation for the Diploma of Electronic and Computer Engineer, Technical University of Crete, 2011.

[5] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, pp. I-I. IEEE, 2001.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. *The Million Song Dataset.* In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011) Florida, 2011.

[7] musiXmatch dataset, the official lyrics collection for the Million Song Dataset, available at: http://labrosa.ee.columbia.edu/millionsong/musixmatch

[8] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python." In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

[9] Giannakopoulos, Theodoros. "pyaudioanalysis: An open-source python library for audio signal analysis." PloS one 10, no. 12 (2015): e0144610.

[10] Kim, Youngmoo E., Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. *Music emotion recognition: A state of the art review.* In Proc. ISMIR, Utrecht, pp. 255-266. 2010.

[11] Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression." In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 94-101. IEEE, 2010.

[12] Jaiswal, Sushma. "Comparison between face recognition algorithm-eigenfaces, fisherfaces and elastic bunch graph matching." Journal of Global Research in Computer Science 2, no. 7 (2011): 187-193.

[13] van Gent, P. (2016). Emotion Recognition With Python, OpenCV and a Face Dataset. A tech blog about fun things with Python and embedded electronics. Retrieved from: http://www.paulvangent.com/2016/04/01/emotion-recognition-with-python-opencv-and-a-face-dataset/

978-1-5386-2044-1/17/$31.00 ©2017 IEEE