IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 2, February 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.512

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

Automatic Facial Emotion Recognition and Turing on the Relative Music

Pankaja R[1], Chethan C[2], Maria Navin J R[3], Chandini D[4]

Assistant Professor, Department of Information Science and Engineering, Sri Venkateshwara College of Engineering, Bangalore, Karnataka, India[1,2,3]

Business Technology Analyst, Deloitte[4]

ABSTRACT: The human face is very important factor for expressing emotions and is considered to play an important role in identifying an individual's mood. Human emotions and intentions are expressed through facial expressions and deriving an efficient and effective feature is the fundamental component of facial expression system. Extracting the required features of human face can be done by capturing the image though web cam. Once the facial emotion is classified, this data can be used to deduce the mood of an individual, and an automatic music will be played relative to the emotion. This paper includes overview of facial emotion recognition system using CNN (Convolutional Neural Network).

KEYWORDS: Facial emotion recognition, Facial expression, CNN.

I. INTRODUCTION

Generally human beings can convey intentions and emotions through nonverbal ways such as gestures, facial expressions and involuntary languages, but most of the time emotions are expressed through facial expressions. These emotions have a huge impact on communication understanding, making decision and also helps in understanding behavioral aspects of human. Emotions play a very important role during communication. The human emotions are basically classified into six basic emotions namely happy, sad, anger, fear, surprise and neutral. Happiness is most desired expression by human. Secondary emotions are cheerfulness, pride, relief, hope, pleasure and thrill. Sadnessis opposite emotion of happiness. Secondary emotion is suffering, hurt, despair, pitty and hopelessness. Anger is one of the most dangerous emotions. This emotion may be harmful, so humans are trying to avoid this emotion. Human should self-control emotions, otherwise they may suffer from mental health issues. Fear is the emotion of danger. It may be because of danger of physical or psychological harm. Secondary emotions of fear are horror, nervousness, panic, worry and dread. The surprise emotion comes when unexpected things happen. Secondary emotions of surprise are amazement, astonishment. Neutral does not tell any kind of emotion of thehuman.

The main objective of this paper is to design an appropriate and efficient algorithm that would recognize the current emotional and behavioural state of the user. The first step is the facial feature extraction from the image that has been captured. We have used CNN (Convolutional Neural Network) algorithm to create a model that could recognize human emotions. Once the emotions are classified an appropriate music will be played.

II. RELATED WORK

Various techniques and approaches have been proposed and developed to classify facial expressions of humans. The proposed approaches have focused only on the some of the emotions. [1]. Vijaykumar R Gule, Abhijeet B. Benke and Shubham S. Jadhav proposed Emotion Based Music Player using Facial Recognition. In this system there are three phases 1. Face Detection 2. Feature Extraction and 3. Classification of Emotion. In the first phase the face detection of the image is done, through in-built web-cam. In the second phase, we use SVM algorithm to extract facial features through landmark points. In the final phase, the generated landmark points are sent to SVM for training. Then the final emotion from the SVM is passed on to the music player and the appropriate music will be played. [2]. Krittrin Chankuptarat, Raphatsak Sriwatanaworachai and Supannada Chotipant proposed Emotion Based Music Player. The proposed system applies both the heart rate and face image. When the application receives a user heart rate from a smart band or a face image from a mobile camera, it analyses what the user emotion is. Then, it suggests songs whose moods are relevant to that user emotion. [3]. Shlok Glida, Husain Zafar and Chintan Soni proposed Smart Music Player Integrating Facial Emotion Regognition and Music Mood Recommendation. This system contains three modules 1.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

Emotion module 2. Music classification module 3. Recommendation module. Emotion module takes an image of the user's face as an input and makes use of machine learning algorithms to identify their mood. Music classification module makes use of audio features for classifying songs into four different mood classes. Recommendation module suggests songs to the user by mapping their emotions to the mood type of the song. [4]. Arto Lehtiniemi and Jukka Holm proposed animated mood picture in music recommendation. In this system the user interacts with a collection of images to receive music recommendation with respect to genre of picture. This system is developed by Nokia research center and it uses textual meta tags for describing the genre and audio signal processing.

III. SYSTEM OVERVIEW

The emotion recognition system is implemented using convolutional neural network. Fig 1 shows overall system architecture, with data processing and classifying emotions based on facial expression, and training the model. This model recognizes facial expression and play the music relevant to the person mood.

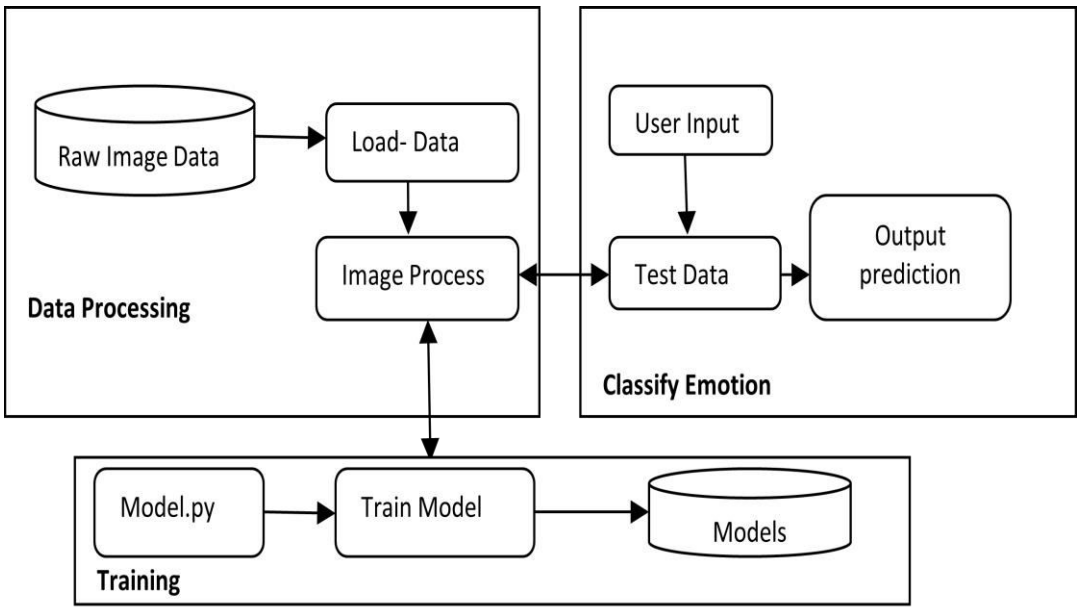

Figure 1: System Architecture

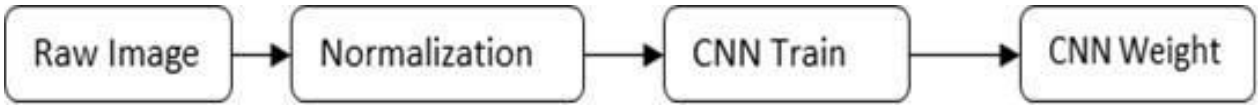

Figure 2: Training Phase

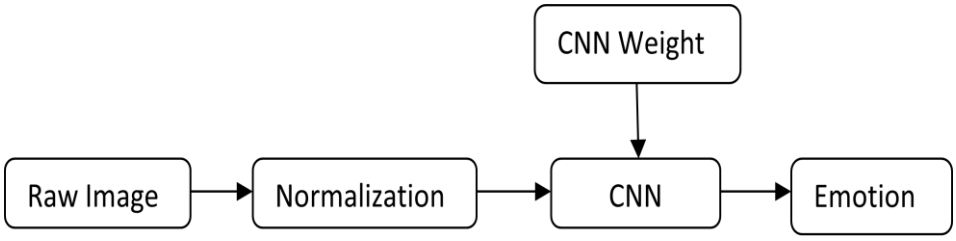

Figure 3: Testing Phase

The training phase is described in Fig.2, the system received a training data comprising gray scale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

To ensure that the training performance is not affected by the order of presentation of the examples, validation dataset is used to choose the final best set of weights out of a set of trainings performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data. The testing phase is described in Fig. 3, the system received a gray scale image of a face from test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a label that represents one of the six basic expressions.

These data splits make sense, but what if you have parameters to tune? Neural networks have a number of knobs and levers (ex., learning rate, decay, regularization, etc.) that need to be tuned and dialed to obtain optimal performance. Well call these types of parameters hyper parameters, and it's critical that they get set properly. In practice, we need to test a bunch of these hyper parameters and identify the set of parameters that works the best. You might be tempted to use your testing data to tweak these values, but again, this is a major no-no! The test set is only used in evaluating the performance of your network. Instead, you should create a third data split called the

validation set. This set of the data (normally) comes from the training data and is used as ―fake test data‖ so we can tune our hyper parameters. Only after have we determined the hyper parameter values using the validation set do we move on to collecting final accuracy results in the testing data. We normally allocate roughly 10-20% of the training data for validation. If splitting your data into chunks sounds complicated, it_s actually not.

From os import path
BASE_PATH= ―/raid/dataets/fer2013‖
INPUT_PATH=path.sep.join([BASE_PATH, ―FER2013/FER2013.CSV‖])

We define BASE_PATH to our root project directory. This location is where the input FER13 dataset will live, along with the output HDF5 dataset files, logs, and plots.

IV. METHODOLOGY

1. Input Layer

The input layer has pre-determined, fixed dimensions, so theimage must be pre-processed before it can be fed into the layer. Normalized gray scale images of size 48 X 48 pixels from Kaggle dataset are used for training, validation and testing. For testing propose laptop webcam images are also used, in which face is detected and cropped using OpenCV Haar Cascade Classifier and normalized.

2. Convolution and pooling (ConvPool) Layers

Convolution and pooling is done based on batch processing. Each batch has N images and CNN filter weights are updated on those batches. Each convolution layer takes image batch input of four-dimension N X Color channel x width x height. Feature map or filter for convolution are also four dimensional. After each convolutional layer down sampling or subsampling is done for dimensionality reduction. This process is called Pooling. Max pooling and Average pooling are two famous polling method. In this project max pooling is done after convolution. Pool size of (2x2) is taken, which splits the image into grid of blocks each of size 2x2 and takes maximum of 4 pixels. After pooling only height and width are affected.

3. Fully Connected Layer

This layer is inspired by the way neurons transmit signals through the brain. It takes a large number of input features and transform features through layersconnected with trainable weights. Two hidden layers of size 500 and 300 unit are used in fully-connected layer. The weights of these layers are trained by forward propagation of training data then backward propagation of its errors. Back propagation starts from evaluating the difference between prediction and true value, and back calculates the weight adjustment needed to every layer before. The output from the second pooling layer is of size Nx20x9x9 and input of first hidden layer of fully- connected layer is of size Nx500. So, output of pooling layer is flattened to Nx1620 size and fed to first hidden layer. Output from first hidden layer is fed to second hidden layer. Second hidden layer is of size Nx300 and its output is fed to output layer of size equal to number of facial expression classes.

4. Output Layer

Output from the second hidden layer is connected to output layer having seven distinct classes. Using Softmax activation function, output is obtained using the probabilities for each of the seven class. The class with the highest probability is the predicted class.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

The network we are going to implement to recognize various emotions and facial expressions is inspired by the family of VGG networks:

1. The CONVlayers in the network will only be 3 × 3.
2. We_ll double the number of filters learned by each CONV layer the deeper we go in the network.

To aid in training of the network, we_ll apply some a priori knowledge gained from experimenting with VGG and ImageNet.
We should initialize our CONVand FClayers using the MSRA/He et al. method – doing so will enable our network to learn faster.
Since ELUs and PReLUs have been shown to boost classification accuracy throughout all of our experiments, let_s simply start with an ELU instead of a ReLU.
From there, open emotionvggnet. pyand start by importing our required Python packages

1. from keras.model import Sequential
2. from keras.layers.normalization import Batch Normalization
3. from keras. layers. Convolutional import Conv2D
4. from keras.layers.convolutional import MaxPooling2D
5. from keras.layers.advanced_activations import ELU
6. from keras.layers.core import Activation
7. from keras.layers.core import Flatten
8. from keras.layers.core import Dropout
9. from keras.layers.core import Dense
10. from keras import backend as k

All of these imports are standard when building Convolutional Neural Networks with Keras.
These data splits make sense, but what if you have parameters to tune? Neural networks have a number of knobs and levers (ex., learning rate, decay, regularization, etc.) that need to be tuned and dialed to obtain optimal performance. Well call these types of parameters hyper parameters, and it's critical that they get set properly. In practice, we need to test a bunch of these hyper parameters and identify the set of parameters that works the best. You might be tempted to use your testing data to tweak these values, but again, this is a major no-no! The test set is only used in evaluating the performance of your network. Instead, you should create a third data split called the validation set. This set of the data (normally) comes from the training data and is used as—fake test data‖ so we can tune our hyper parameters. Only after have we determined the hyper parameter values using the validation set do we move on to collecting final accuracy results in the testing data. We normally allocate roughly 10-20% of the training data for validation. If splitting your data into chunks sounds complicated, it_s actually not.

1. From os import path
2. BASE_PATH = —/raid/dataets/fer2013‖
3. INPUT_PATH = path.sep.join([BASE_PATH, —FER2013/FER2013.CSV])

We define BASE_PATH to our root project directory. This location is where the input FER13 dataset will live, along with the output HDF5 dataset files, logs, and plots.
Let_s also define the number of classes in the FER13 dataset:
#define the number of classes (set to 6)
# NUM_CLASS = 7
NUM_CLASS = 6
In total there are six classes in FER13: angry, fear, happy, sad, surprise, and neutral.
We will be converting the fer2013.csv file into series of
HDF5 datasets for training, validation, and testing, we
need to define the path to these output HDF5 file.
TRAIN_HDF5 = path.sep.join([BASE_PATH, —hdf5/train.hdf5‖]) VAL_HDF5 = path.sep.join([BASE_PATH, —hdf5/val.hdf5]) TEST_HDF5 = path.sep.join([BASE_PATH, —hdf5/test.hdf5‖])
Finally, we_ll initialize the batch size when training our CNN along with the output directory where any logs or plots will be stored:
BATCH_SIZE = 128

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

#define the path to where output logs will be stored OUTPUT_PATH = path.sep.join ([BASE_PATH, —output])

V. RESULTS AND DISCUSSION

CNN architecture facial expression recognition mentioned above was implemented in Python. Along with Python programming language, Numpy, Tensorflow, and OpenCV libraries were used. Training image batch size was taken as 30, while filter map is of size 20x5x5 for both convolution layer. Validation set was used to validate the training process. In last batch of every epoch in validation cost, validation error, training cost, training error are calculated. Input parameters for training are image set and corresponding output labels. The testing of the model is carried out using images. The classifier provided 92.77% accuracy. This model gives highest accuracy for happy with 88.86%, surprise with 74.52%, neutral with 69.32%, anger with 42.16%, fear 38.31% and lowest accuracy for sad emotion as 38.23%. It is interesting to see that the model performed well in predicting the happy label, which implies that learning the features of a happy face is easier than other expressions. Fig 4 shows different type emotions after training the model.

Existing emotion recognition methods are tested using smart band [2] using smart band and heart rate is difficult to know mode of the person it's because of environmental condition. Using machine learning algorithm [1] using SVM machine learning algorithm identifying expression is not accurate that is due to noise and glitches in image. Mixed mode expression can be detecting. Using CNN give accurate result for emotion recognition and also simultaneously detect multiple expression shown in Fig 6. And play the music according to facial expression.

Above table shows that this model gives highest accuracy for happy with 88.86%, surprise with 74.52%, neutral with 69.32%, anger with 42.16%, fear 38.31% and lowest accuracy for sad emotion as 38.23%. It is interesting to see that the model performed well in predicting the happy label, which implies that learning the features of a happy face is easier than other expressions. Also, for most of the expressions, using CNN has increased the classification accuracy. Hence, the CNN has improved the model accuracy, which means that the convolutional networks can learn the key facial features by using only the raw pixel data.

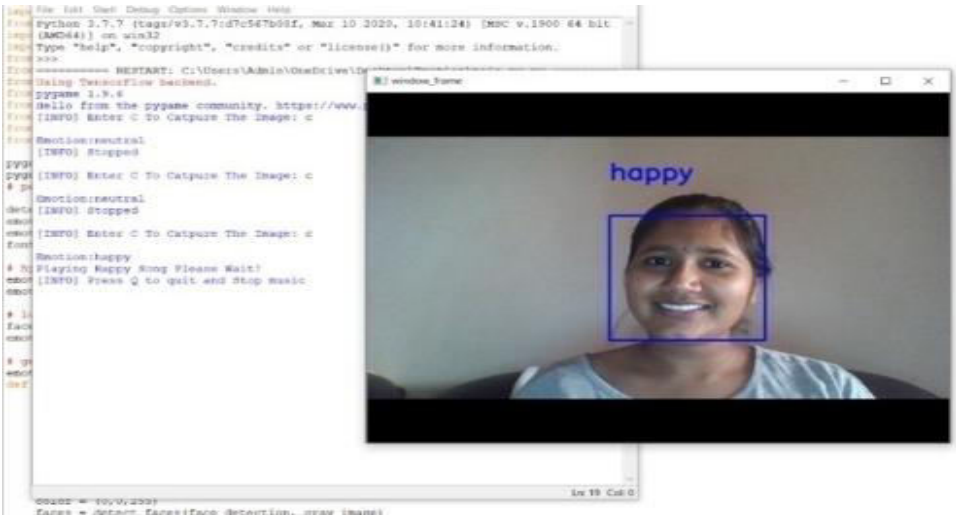

Figure 4: Result for Happy

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

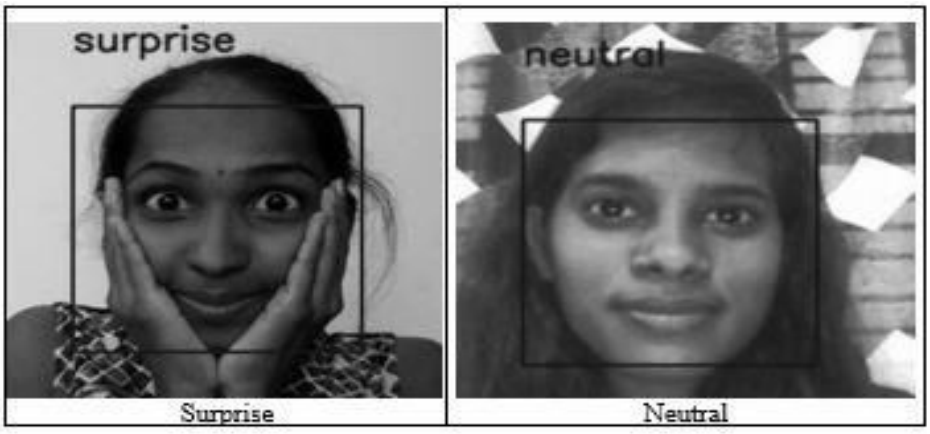

Figure 5: Results

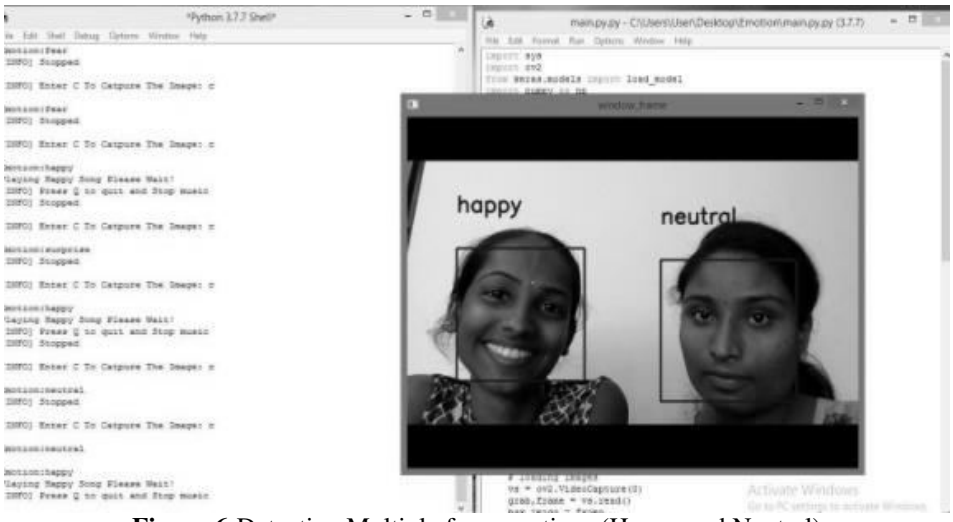

Figure 6: Detecting Multiple face emotions (Happy and Neutral)

VI. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, a architecture based six layer convolution neural network is implemented to classify human facial expressions that is happy, sad, surprise, fear, anger, disgust, and neutral and efficiency of classification using CNN is effiencet. The system would be to design a mechanism that would be helpful in music therapy treatment and provide the music therapist the help needed to treat the patients suffering from disorders like mental stress, anxiety, acute depression and trauma.

REFERENCES

[1] Vijaykumar R G, Abhijeet B. Benke and Shubham S. Jadhav. (2017). Emotion Based Music Player using Facial Recognition. International Journal of Innovative Research in Computer and Communication Engineering, 5(2), 2188-94.DOI: 10.1109/ICSC45622.2019.8938384

[2] Krittrin Chankuptarat, Raphatsak Sriwatanaworachai and Supannada Chotipant. (2019). Emotion Based Music Player". In2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST),1-4. DOI: 10.1109/ISSPIT.2017.8388671

[3] Shlok Glida, Husain Zafar and Chintan Soni. (2017). Smart Music Player Integrating Facial Emotion Regognition and Music Mood Recommendation. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 154-158. DOI: 10.1109/WiSPNET.2017.8299738

[4] Arto Lehtiniemi and Jukka Holm. (2012). Animated Mood Picture in Music Recommendation. International Conference on Information Visualisation, 143-150.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 10, Issue 2, February 2021 ||

DOI:10.15680/IJIRSET.2021.1002010

[5] Anagha S.Dhavalikar and Kulkarni. (2014). Face Detection and Facial Expression Recognition System. Interntional Conference on Electronics and Communication System, 1-7.DOI: 10.1109/ECS.2014.6892834

[6] Yong-Hwan Lee, Woori Han and Youngseop Kim. (2013). Emotional Recognition from Facial Expression Analysis using Bezier Curve Fitting. International Conference on Network-Based Information Systems, 250-254.DOI: 10.1109/NBiS.2013.39

[7] Faiza Abdat, Choubeila Maaoui and Alain Pruski. (2011). Human-computer interaction using emotion recognition from facial expression. European Symposium on Computer Modelling and Simulation, 196-201.DOI: 10.1109/EMS.2011.20

[8] Te Hsun Wang and Jenn Jeir James Lien. (2009). Facial Expression Recognition System Based on Rigid and Non-Rigid Motion separation Pose Estimation. Journal of Pattern Recognition, 42(5), 962-977.DOI: https://doi.org/10.1007/978-0-387-78414-4_21

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.512

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details