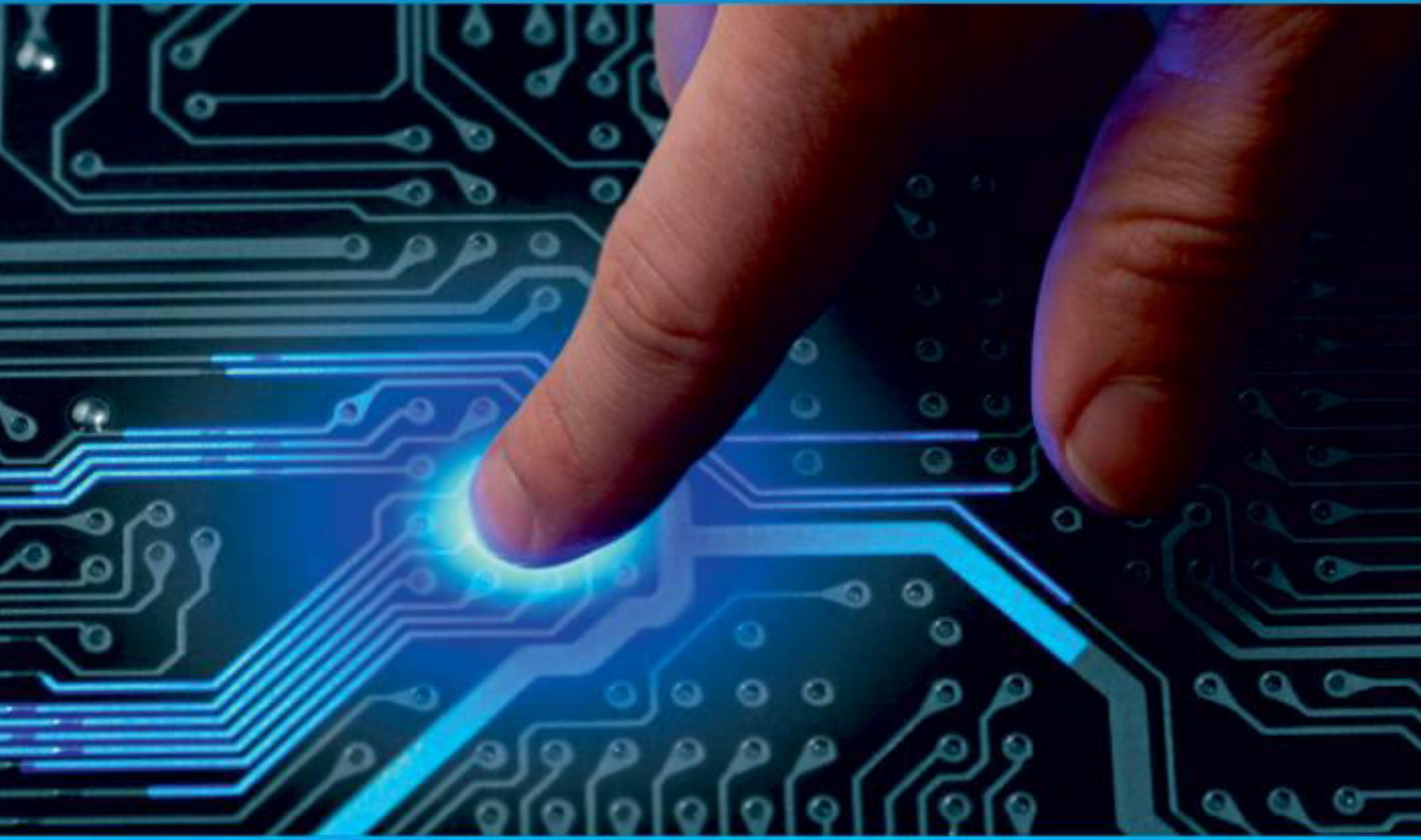
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 4, April 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004161|

Intelligent Supervising System Using Expression Processing

Vishal Upadhyay, Yash Ghodegaonkar, Rutuja More, Poonam Bhawari

UG Students, Dept. of I.T., Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India

ABSTRACT: One of the fundamental challenges faced by the teachers - whether human or robot - is determining how well his/her students are receiving a lecture at a given moment. each individual student may be content, confused, bored, or excited by the lesson at a particular point in time, and one student’s perceptionof the lecture may not necessarilybe shared by his/her peers. While explicit feedback signalsto the teachersuch as a question or a request to repeat a sentence are useful, they are limited in there effectiveness for several reasons. If the student is confused, he may feel embarrassed in asking a question. If the student is bored, it may be inappropriate to ask the teacher to speed up the rate of presentation. some research has also shown that students are not always aware of when they need help. Finally, even when students do ask questions, this feedback may, in a sense come too late - the student may already have missed an important point, and the teacher must spend lesson time to clear up the misunderstanding.

KEYWORDS: Energy efficient algorithm; Manets; total transmission energy; maximum number of hops; network lifetime

I. INTRODUCTION

Intelligent Supervising System is solution for a must needed problem of switching from offline to online tutoring environment that is as productive as offline learning by providing of feedback-based solution of every interactive response by viewers for the tutor to get along with the flow.

The real problem is the live interaction with students that happen in offline tutoring but not available in online tutoring and lecture delivery. the lecture is driven majorly by the student’s response to the modules taught accordingto the difficulty level of that section in module. but this didn't solve the problem as the video conferencing platforms used for online tutoring and lecture delivery world developed keeping in mind the meeting heldin video call to eradicate the distance barriers where conversations and interactions happen in a two -way communication.

In 2020, the lecture deliveries throughout the universities were switched in online mode due to the prevailing COVID - 19. And the biggest challenge put forth was the continuation of academic activities, of which lecture delivery through video conferencing platforms were utilized. But in online lecture delivery the two-way communication doesn’t happen that frequently as compared to video calls. therefore, the idea of utilizing the emotions/expressions using facial images in runtime environment so that the lecturer gets a glimpse of the state of the class and module progress.

II. RELATED WORK

Jiawen Jiang, Haiyang Zhang, Chenxu Dai, Qingjuan Zhao, Hao Feng, Zhanlin Ji, and Ivan Ganchev[1] introduced four ATS novel models in a Sequence-to-Sequence format (Seq2Seq), using Long-based bidirectional Long Short- Time Memory (LSTM), with additional additions to increase the interaction between the generated text summary and source text and to solve unregistered word problems, overcome duplicate words and avoid an increase in compound errors in created text abbreviations. In this paper Experiments conducted on two public data sets have confirmed that the ATS models introduced achieve better performance than the basics and some of the technologies considered.

IyanuPelumiAdegun and Hima Bindu Vedapalli [2]described the micro expression transitions from images and the concerned expressions. Micro-expression recognition is a growing research area owing to its application in re-veiling subtle intention of humans, especially while under high stake conditions. With the rapid increase in security issues all over the world, the use of micro-expressions to understand one’s state of mind has received major interest.

The transition from intelligent to affective tutoring systemin this paper is written by Muhammad Asif Hasan, Nurul farmyard mohdnoor,sitiSoraya binti abdulrahman, and mohammadmustaneerRahman[3], the research describes the swelling use of computerized learning, accompanied by the rapid growth of information technology has become a surge of interest in the research community. Consequently, several technologies have been developed to maintain and promote computerized learning. In this study, we provided an in-depth analysis of two of the prominent computerized learning systems i.e., intelligent tutoring system (its) and affective tutoring system (ats).

The Face Expression Based Emoticon Identification (FEBEI) system is an open-source extension to the Tracker.js framework which converts a human facial expression to the best matching emoticon. The contribution of this project was to build this robust classifier which can identify facial expression in real time without any reliance on an external server or computation node. An entirely client-side JavaScript implementation has clear privacy benefits as well as the avoidance of any lag inherent in uploading and downloading images. We accomplished this by utilizing several computationally efficient methods. Tracking.js provided a Viola Jones based face detector which we used to pass facial images to our own implementation of an eigen emotion detection system which was trained to distinguish between happy and angry faces. We have implemented a similar eigenface classifier in python and have trained a Convoluted Neural Network (CNN) to classify emotions to provide a comparative view of its advantages. We aim to make FEBEI easily extendable so that the developer community will be able to add classifiers for more emoticon.

No.	Title and Author	Description	Algorithms & Techniques used	Conclusion / Remarks
1.	Emotion Recognition of Students Based on Facial Expressions in Online Education Based on the Perspective of Computer Simulation (Weiqing Wang, Kunliang Xu, HongliNiu and Xiangrong Miao)	The framework mainly consists of two parts:1. Online Education Platform 2. The Learning Model Based on CNN.	1.CNN 2.FER	The overall result can be presented in a histogram intuitively, and teachers can adjust their teaching strategies accordingly to improve the efficiency of online teaching.
2	Automatic Facial Expression Recognition in Standardized and Non-standardized Emotional Expressions(Theresa Küntzler , T. Tim A. Höfling and Georg W. Alpers)	The classic approach to analyse emotional facial responses is either an expert observation such as the Facial Action Coding System.	1.FER 2.FACS	This study contributes to the literature by comparing the accuracy of three state-of-the-art FER systems to classify emotional facial expressions. We show that all systems and human coders perform well for standardized, prototypical facial expressions.
3	Face recognition using facial symmetry (Avinash Kumar Singh)	The algorithm is divided into two parts (a)Training and (b) Testing. Generally testing is performed offline and testing is performed in real time scenario.	1.PCA	Time is also a big factor in this regard. Time is directly proportional to the size (dimension) of the data (images). As human faces are almost identical in nature. Hence, we have used only left half of the human faces for training and testing. We have tested our hypothesis on ORL

4.	A Review of Person Recognition Based on Face Model (Shakir Fattah Kak, Pedro Valente, Firas Mahmood Mustafa)	These strategies analyze a person's physiological as well as behavioral attributes with a specific end goal to decide and/or ascertain his/her identity.	1.Support Vector Machine (SVM) 2.Artificial Neural Network (ANN)	The article reviews a significant number of papers that cover the recent development in the field of face recognition.

III. PROPOSED ALGORITHM

Design Considerations:

A facial expression recognition works on the pre-trained models extracting the features of the facial symmetry wherein different emotions/expressions make it possible. Therefore, the projects core part is emotions/expressions identification at the runtime live fraction of sessions. Haar cascade classifiers are an effective way for object detection. This method was proposed by paul viola and michael jones in their paper rapid object detection using a boosted cascade of simple features.Haar cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier. An open-source video conferencing platform i.e. Big blue button is utilized for creating video-rooms, also the tuning in the video - rooms is possible as the open-source allows adding desired features as per the requirement. A server on digital ocean is hosted for all the backend and frontend operations of facial expression images parsing, feature extraction, pie-chart generation and overlay on video – conferencing platforms.

Description of the Proposed Algorithm:

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.While initially a form of computer application, facial recognition systems have seen wider uses in recent times on smartphonesand in other forms of technology, such as robotics.

Because computerized facial recognition involves the measurement of a human's physiological characteristics facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognitionand fingerprint recognition, it is widely adopted due to its contactless process. Facial recognition systems have been deployed in advanced human-computerinteraction,video surveillanceand automatic indexingof images. They are also used widely by law enforcement agencies.

eq.(3)

Methodology

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.While initially a form of computer application, facial recognition systems have seen wider uses in recent times on smartphonesand in other forms of technology, such as robotics.

Because computerized facial recognition involves the measurement of a human's physiological characteristics facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognitionand fingerprint recognition, it is widely adopted due to its contactless process. Facial recognition systems have been deployed in advanced human-computerinteraction,video surveillanceand automatic indexingof images. They are also used widely by law enforcement agencies.

V. STAGES OF IMPLEMENTATION PREPARATION DATA

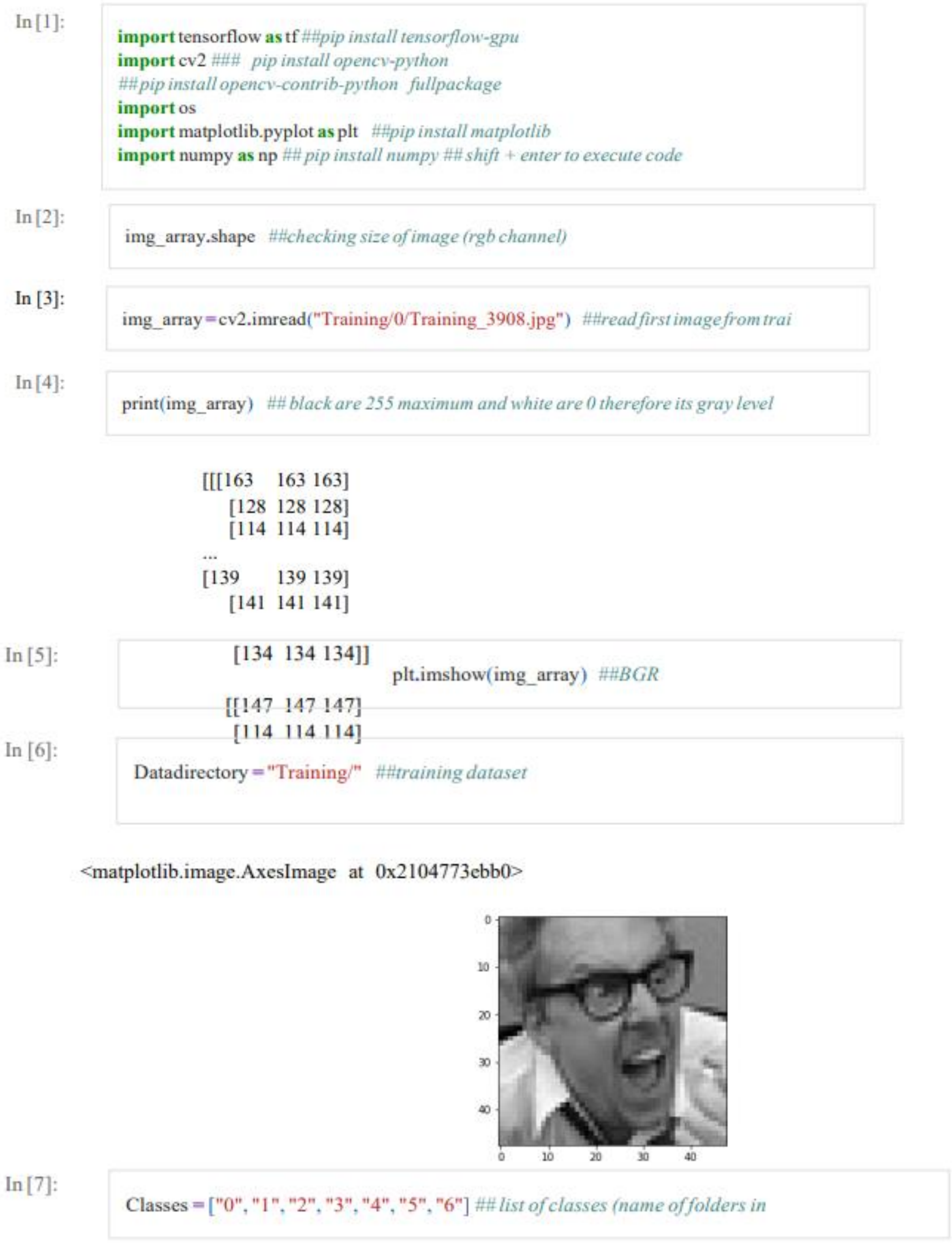
Facial expression recognition is categorized into four stages: pre-processing,registration of face, feature extraction of face, and classification of expressions. It is usedin several applications such as understanding mental disorders, analysing what is goingon in a human's mind, lie detection, etc. Here, a convolutional neural network is utilizedwhich is very effective for image processing. The data used is supervised data. TheFER2013 dataset is used for analysis and

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004161|

popular deep learning frameworks such asKeras and OpenCV for the detection of facial expressions in an accurate manner. The step-by-step model training on FER2013 and facial expression recognition part is shown below:



In [1]:

```
import tensorflow as tf ##pip install tensorflow-gpu
import cv2 ###  pip install opencv-python
##pip install opencv-contrib-python  fullpackage
import os
import matplotlib.pyplot as plt  ##pip install matplotlib
import numpy as np ## pip install numpy ## shift + enter to execute code
```

In [2]:

```
img_array.shape  ##checking size of image (rgb channel)
```

In [3]:

```
img_array=cv2.imread("Training/0/Training_3908.jpg")  ##read first image from trai
```

In [4]:

```
print(img_array)  ## black are 255 maximum and white are 0 therefore its gray level
```

```
[[[163   163 163]
  [128  128 128]
  [114  114 114]
...
[139      139 139]
  [141  141 141]
  [134  134 134]]
[[147  147 147]
  [114  114 114]
```

In [5]:

```
plt.imshow(img_array)  ##BGR
```

In [6]:

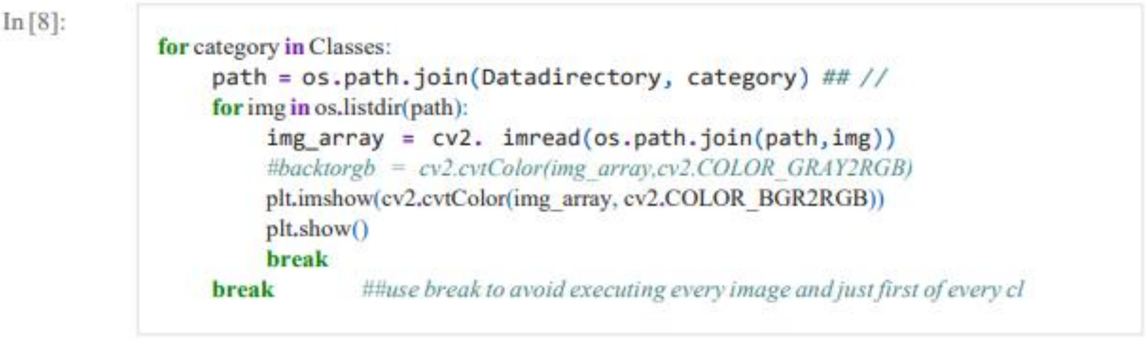
```
Datadirectory="Training/"   ##training dataset
```

```
<matplotlib.image.AxesImage at 0x2104773ebb0>
```

In [7]:

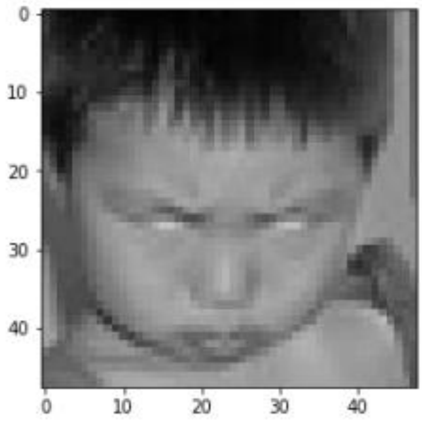
```
Classes=["0","1","2","3","4","5","6"] ## list of classes (name of folders in
```

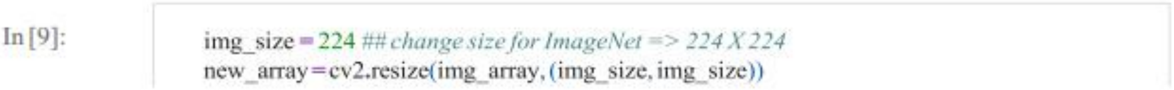
| DOI: 10.15680/IJIRCCE.2022.1004161|



In [8]:

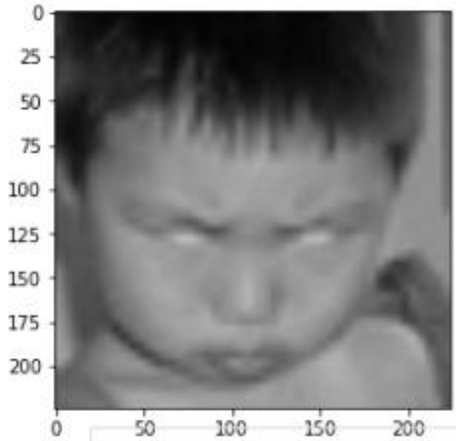
```
for category in Classes:
    path = os.path.join(Datadirectory, category) ## //
    for img in os.listdir(path):
        img_array = cv2. imread(os.path.join(path,img))
        #backtorgb  = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
        plt.imshow(cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB))
        plt.show()
        break
    break          ##use break to avoid executing every image and just first of every cl
```

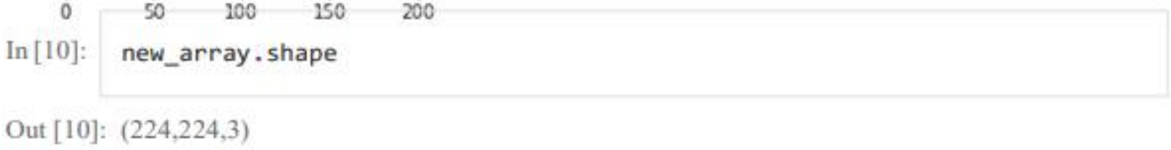




In [9]:

```
img_size = 224 ## change size for ImageNet => 224 X 224
new_array=cv2.resize(img_array, (img_size, img_size))
```





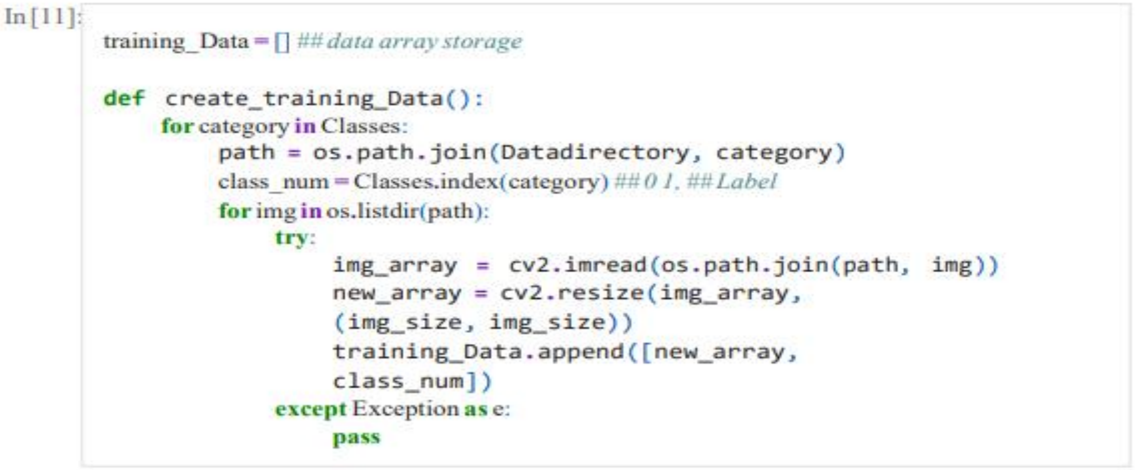
In [10]:

```
new_array.shape
```

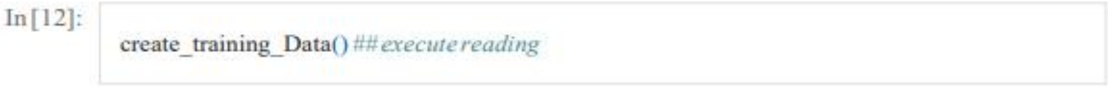
Out [10]:

```
(224,224,3)
```

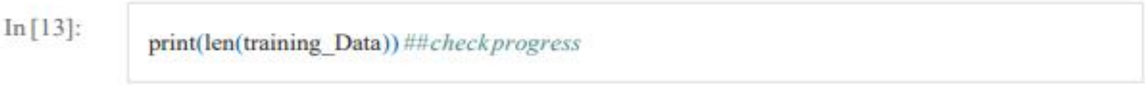
Read all images and convert them to array:



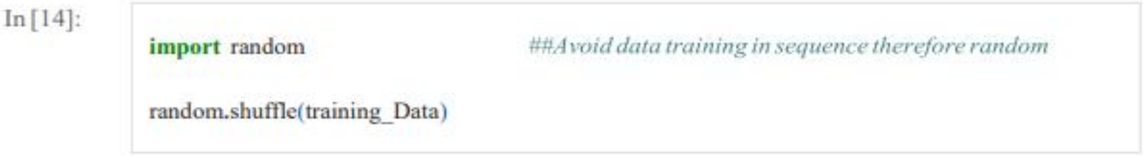
```
In [11]:
training_Data = [] ## data array storage

def  create_training_Data():
     for category in Classes:
          path = os.path.join(Datadirectory, category)
          class_num = Classes.index(category) ## 0 1, ## Label
          for img in os.listdir(path):
               try:
                    img_array  =  cv2.imread(os.path.join(path,  img))
                    new_array = cv2.resize(img_array,
                    (img_size, img_size))
                    training_Data.append([new_array,
                    class_num])
               except Exception as e:
                    pass
```



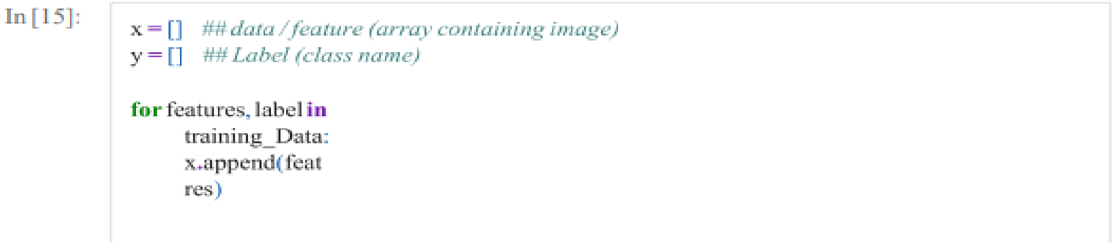
```
In [12]:
create_training_Data() ## execute reading
```



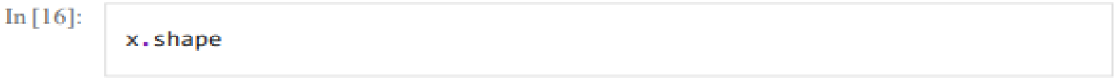
```
In [13]:
print(len(training_Data)) ##check progress
```



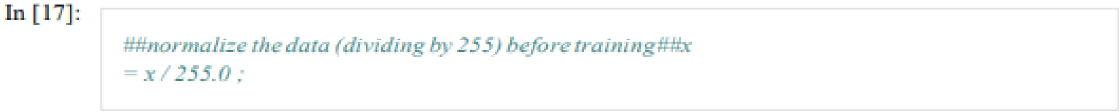
```
In [14]:
import random                    ##Avoid data training in sequence therefore random

random.shuffle(training_Data)
```



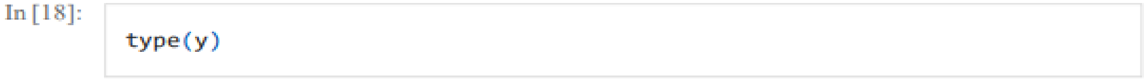
```
In [15]:
x = []   ## data / feature (array containing image)
y = []   ## Label (class name)

for features, label in
       training_Data:
       x.append(feat
       res)
```



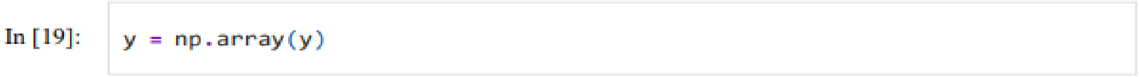
```
In [16]:
x.shape
```



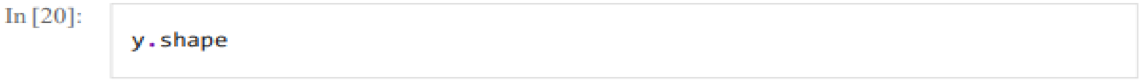
```
In [17]:
##normalize the data (dividing by 255) before training##x
= x / 255.0 ;
```



```
In [18]:
type(y)
```

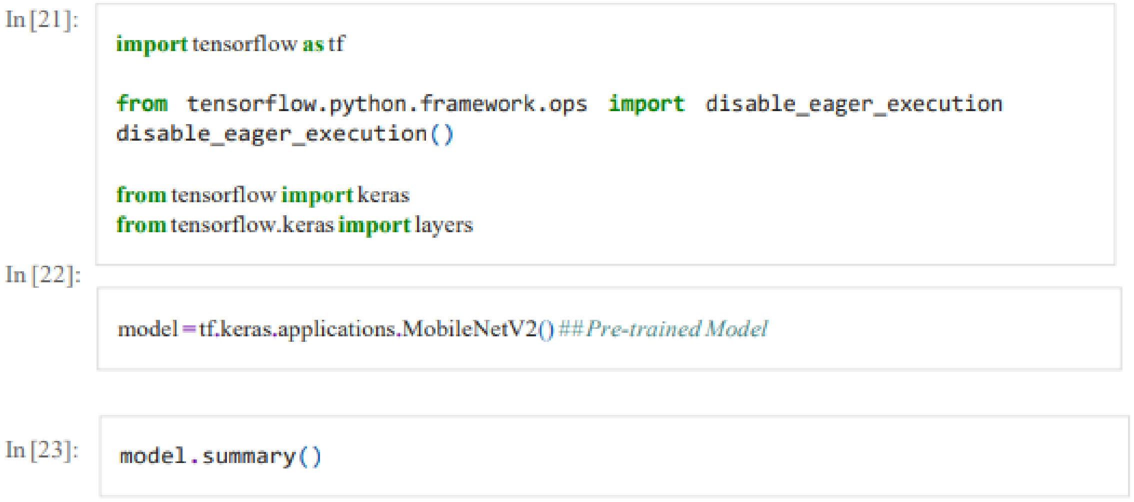


```
In [19]:
y = np.array(y)
```



```
In [20]:
y.shape
```

Deep Learning model for training - TransferLearning



```
In [21]:
import tensorflow as tf

from tensorflow.python.framework.ops import disable_eager_execution
disable_eager_execution()

from tensorflow import keras
from tensorflow.keras import layers
```

```
In [22]:
model=tf.keras.applications.MobileNetV2() ##Pre-trained Model
```

```
In [23]:
model.summary()
```

```
================================================================================
=
Layer (type)            Output Shape              Param #            Connected to
============================================================================
input_1 (InputLayer)       [(None, 224, 224, 3   0       []
                                                         )]

Conv1 (Conv2D)                      (None,  112,  112,  32   864         ['input_1[0][0]']
                                    )

bn_Conv1  (BatchNormalization)      (None,) 112,  112,  32   128         ['Conv1[0][0]']

Conv1_relu  (ReLU)                  (None,  112,  112,  32   0           ['bn_Conv1[0][0]']
                                    )

expanded_conv_depthwise (Depth  (None, 112, 112, 32  288     ['Conv1_relu[0]
[0]']                                             )
wiseConv2D)

expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128   ['expanded_conv_depthwise[0][0]']          )
tchNormalization)
==============================================================

Total params: 3,538,984
Trainable params: 3,504,872
Non-trainable  params:  34,112
```

Transfer Learning -Tuning, weights will start from lastcheck point

In [24]:
```
base_input = model.layers[0].input #check with first layer as 0
```

In [25]:
```
base_output = model.layers[-2].output ##model contains 1000 classes so cutting down
```

In [26]:
```
new_model.compile(loss="sparse_categorical_crossentropy",  optimizer  =
"adam",  metric
```

In [27]:
```
new_model  =  tf.keras.models.load_model("Final_model_95p07.h5")
```

In [28]:
```
##new_model.evaluate ##test data, not used here coz of live image test
```

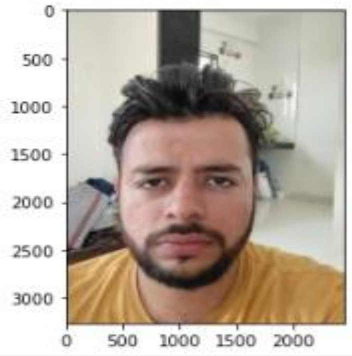
In [29]:
```
frame = cv2.imread("images/sad.jpg")
```

In [30]:
```
frame.shape   ##BGR Channel
```

Out[31]: (3264, 2448, 3)

In [32]:
```
plt.imshow(cv2.cvtColor(frame,  cv2.COLOR_BGR2RGB))
```

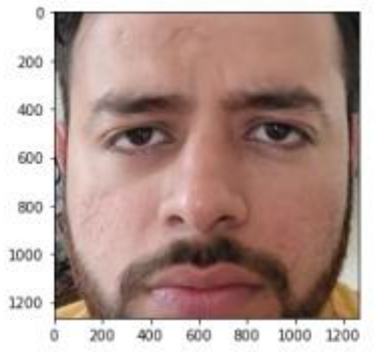
Out[32]: <matplotlib.image.AxesImage at 0x2122989bf40>



In [33]:
```
plt.imshow(cv2.cvtColor(face_roi,  cv2.COLOR_BGR2RGB))  ##  Crop  image
```

Out[34]: <matplotlib.image.AxesImage at 0x2123ae71070>

In [35]:
```
final_image = cv2.resize(face_roi, (224, 224)) ##changing dimension
final_image = np.expand_dims(final_image, axis = 0)                    ##need
forth dimension##final_image = final_image / 255.0
                                                ##normalizing
```



VI. PROCESSING

After the face is obtained, the next step is to extract facial features. Two types of features can be extracted: geometric features and appearance features. Geometric features present the shape and locations of facial components (including mouth, eyes, brows, and nose). The facial components or facial feature points are extracted to form a feature vector that represents the face geometry. The appearance features present the appearance (skin texture) changes of the face, such as wrinkles and fur- rows.

The appearance features can be extracted on either the whole-face or specific regions in a face image. The HaarCascade Frontal Face works on the same logic as described above. Then the BBB API frameworks are utilised for video – conferencing platform. Also, the algorithm is implemented on Server console for output.

Parent and child python scripts govern the whole execution of the Algorithm initiation, image capturing in frames per second and then processing the facial feature extraction for expression recognition, mapping it to a pie chart and then dynamically overlaying it on the video conference.

The image of pie -chart is generated which is updated every 5 seconds in the frame overlay depicting a live pie-chart of present state of class as shown below:

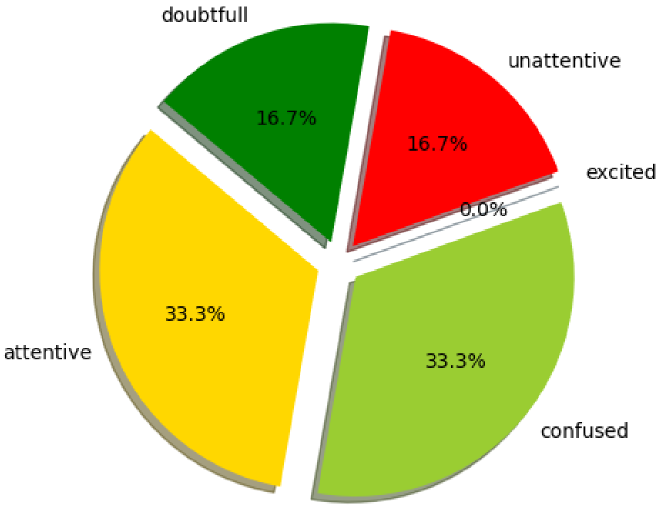


Figure: Live Pie Chart

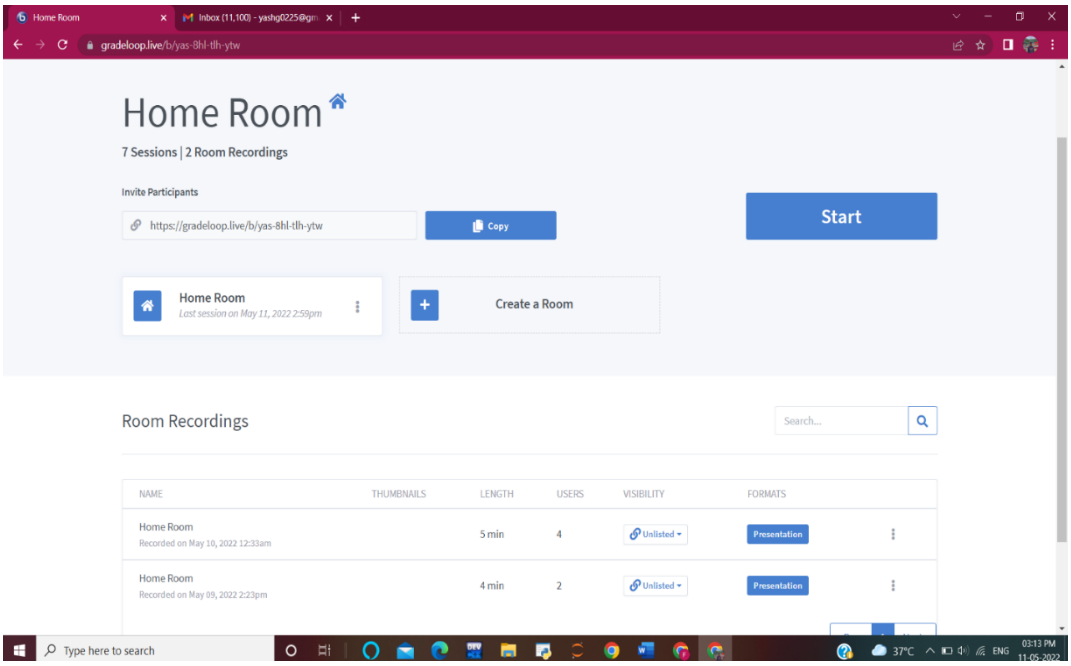
| DOI: 10.15680/IJIRCCE.2022.1004161|

IMPLEMENTATION ISSUES & SOFTWARE TOOLS

As we know we are using Machine Learning algorithms for Facial Expression summarization, we need high computation power. For image processing techniques, we need computer supported GPU. Also for normalization of all the values in array stored , a free space RAM of 33.1 GB is required on our model with 28709 images. Therefore we haven't done the normalization as we are limited with 8 GB RAM locally and on server side.

VII. SIMULATION RESULTS

EXPERIMENTS



Home Room

7 Sessions | 2 Room Recordings

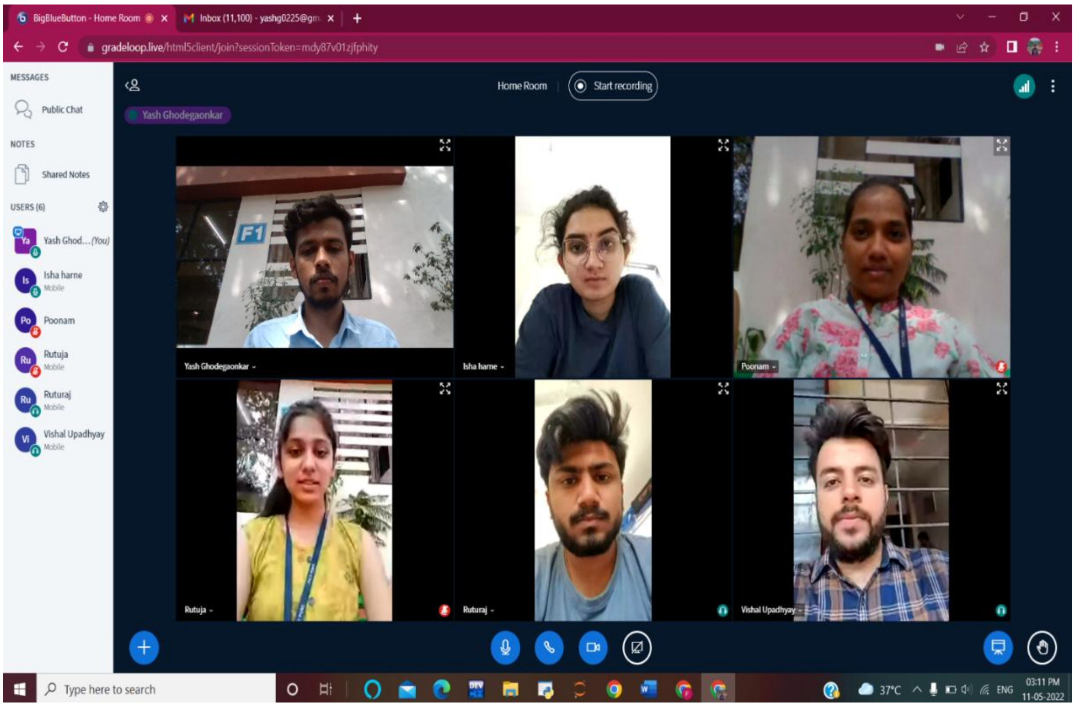
Invite Participants

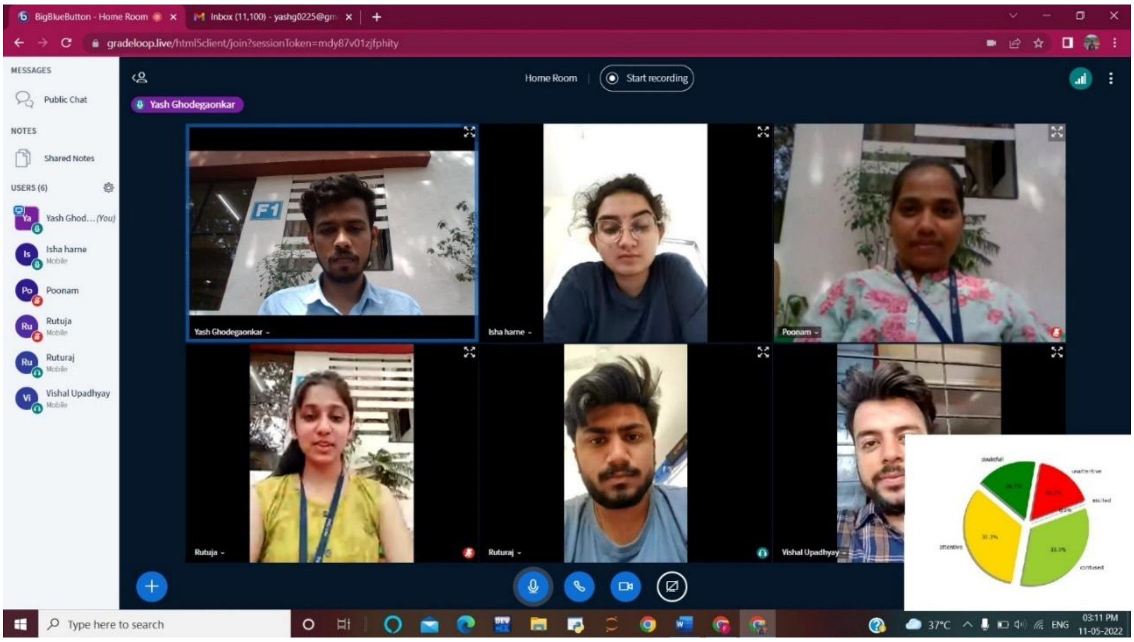
https://gradeloop.live/b/yas-8hl-tlh-ytw

Home Room
Last session on May 11, 2022 2:59pm

Room Recordings

NAME	THUMBNAILS	LENGTH	USERS	VISIBILITY	FORMATS
Home Room Recorded on May 10, 2022 12:33am		5 min	4	Unlisted	Presentation
Home Room Recorded on May 09, 2022 2:23pm		4 min	2	Unlisted	Presentation





TESTING

Software testing may be a critical element of software quality assurance and represents the last word review of specification, design, and code generation. If a system is implemented without being tested it may lead to erroneous working and dissatisfaction on part of the customer.

It will also prove disastrous to the reputation of the organization or one that developed the system and cause a loss in business. Software testing involves a software test environment for testing, identifies the tests to be performed, and provides schedules for test activities to be conducted.

Testing:

White Box Testing is a software testing technique in which the internal structure, design, and coding of software are tested to verify the flow of input-output and to improve design, usability, and security. It is also called Clear box testing, Open box testing, transparent box testing, Code-based testing, and Glass box testing because code is visible to testers. This testing is one of two parts of the Box Testing approach to software testing.

Blackbox testing is all about testing from an external or end-user type perspective. While White box testing in software engineering is based on the internal workings of an application and revolves around internal testing.

Unit Testing:

To ensure the quality, individual components are tested independently. The focus is to uncover errors in design and implementation focus is given on the following parameters.

- Registration of new user in the login module.
- Program logic and program structure in a module.
- Each module was tested.

| DOI: 10.15680/IJIRCCE.2022.1004161|

Integration Testing:

A Group of dependent components is tested together to ensure the quality of their integration unit. The purpose is to uncover errors when different modules of projects are integrated. For this parameter are taken.

Black Box:

In Black Box Testing the functionalities of software applications are tested without having the knowledge of internal code structure, implementation details, and internal paths. Input and output of software applications are considered while performing black box testing and it is entirely based on software requirements and specifications. Black Box is popularly known as behavioural Testing.

Test Cases:

TC ID	OBJECTIVE	STEPS	STATUS
TC1	Detect the attentiveness of the student in live lecture	First, the Face detection model will detect the face then count how much time student is attending lecture.	Pass
TC2	Display the expressions of students while attending lectures	First, the face detection will detect the face then capture the expression of student and it will display the name of expression on screen.	Pass
TC3	Display the overall graph of expressions of students in a lecture	First, it detects the face and count attentiveness of student then detect expressions of every student and finally display graph of overall expression of class.	Pass
TC4	Activating live video conferencing with one admin and multiple participants	Admin creates video conferencing sessions and shares the URL to the participants. Participants can join the session by URL.	Pass
TC5	Microphone and camera permission be available to every participant in the session	Microphone is enabled Camera is enabled	Pass
TC6	Chat box is available	Writing text in chat is available	Pass
TC7	Polling feature is available	Creation of poll at admin side and polling option is on participant side	Pass
TC8	Session recording is available	Recording must be click by admin and pop up is displayed to every participant	Pass
TC9	The analysis frame is shown	A separate window for analysis opens up	Pass
TC10	Continuous analysis of every frame within video stream of participant	Frames are analysed with the batch of five frames of images.	Fail

Table Test Cases

Summary of Black Box Testing:

This tested as a whole, if verifies all elements properly to make sure that all system function and performance are achieved in target environment.

- System function and performance.
- All possible input condition, output and test log is maintained.

VIII. Conclusion and Future Work

We went through many platforms and tools and saw several research papers, so we concluded that we're going to build a system that aims to show the Realtime Graph indication of current state of class in all aspects of attentiveness, learning deliverables and its understanding displayed to the faculty. These include image acquisition, pre-processing of an image, face detection, feature extraction, classification and then when the emotions are classified the system assigns the user particular music according to his Facial expression.

Our system focuses on live videos taken from the webcam. The main aim of this project is to develop an automatic facial expression recognition system post which graph indicators will define the objective of our project. The expressions used for the experiments include happiness, Sadness, Surprise, Fear, Disgust, and Anger that are universally accepted.

In the future, there are many points that deserve our attention.

One-to-Many :

- Online Lectures
- Webinars / Workshops / Tutorials
- General Meetings and Discussions

One-to-One :

- Personal Tutoring
- Counselling sessions
- Online Tests.

REFERENCES

1. Emotion Recognition of Students Based on Facial Expressions in Online Education Based on the Perspective of Computer Simulation Hindawi , Complexity Volume 2020.
2. Factors Affecting the Perception of 3D Facial Symmetry from 2D Projections Michael B. Lewis ID School of Psychology, Cardiff University, Published: 21 October 2017
3. The Transition From Intelligent to Affective Tutoring System: A Review and Open Issues Received October 14, 2020, accepted October 27, 2020, date of publication November 9, 2020, date of current version November 20, 2020.
4. Automatic facial expression recognition in standardized emotional expressions. Published: 05 May 2021
5. ‘‘The Facial Symmetry of Faces’, ‘Michaela Benz’, ‘Januay 2017’
6. ‘Attention Based Facial Symmetry Detection’, ‘Fred Stentiford’, ‘October 2015’
7. Facial Expression Recognition-YingliTian,TakeoKanade, and Jeffrey F. Cohn- Oct2019
8. Ahlberg,J.,Forchheimer,R.:Face tracking for model-based coding and face animation .Int. J.Imaging Syst. Technol.2016

9. A review of person recognition based on face model January 2018 Engineering ISSN

BIOGRAPHY

Vishal Upadhyay, Yash Ghodegaonkar ,Rutuja More, Poonam Bhawari ;are the students of Society for Computer Technology and Research's ,Pune Institute of Computer Technology 2022 Batch UG, I.T Department who have worked on this research and implemented the above project .

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi cross ref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details