IOT Based Smart Candidate Assessment System During Interviews

Fathima Farsana V S
Department of Computer Science & Engineering,
Albertian Institute of Science & Technology, India
Email: fathimafarsanavs@gmail.com

Adarsh T D
Department of Computer Science & Engineering,
Albertian Institute of Science & Technology, India
Email: adarshdilish2001@gmail.com

Gayathri Suresh
Department of Computer Science & Engineering,
Albertian Institute of Science & Technology, India
Email: gayathrisuresh173@gmail.com

Suja C Nair
Department of Computer Science & Engineering,
Albertian Institute of Science & Technology, India
Email: sujacnair@aisat.ac.in

Alfiya Yousaf
Department of Computer Science & Engineering,
Albertian Institute of Science & Technology, India
Email: alfiyayousaf25@gmail.com

Abstract – ***An interview is a conversation between a candidate and company professionals to assess if the candidate is the right fit. As an interviewer, this part of the recruitment process allows you to find out more about the candidate, such as their personality and background. The paper highlights the importance of job interviews in the hiring process to evaluate an applicant's knowledge, skills, abilities, and behaviour. The evaluation is based on both verbal and nonverbal communication, including facial expressions, hand gestures, and body postures, which enrich the vocal content of the interview. Nonverbal communication carries relevant information that can reveal a person's personality, state of mind, or job interview outcome, conveying information in parallel to speech. To address the evaluation of verbal and nonverbal behaviour, the paper proposes an automated, predictive expert system framework for the computational analysis of HR job interviews. The framework analysis visual cues, audio cues and pulse rate details of the interviewees and quantifies their verbal and nonverbal behaviour. Based on this analysis, the system predicts the interviewee's overall performance rating, behaviour traits, personality, and hire ability.***

The proposed system can potentially have significant implications for the recruitment process, including reducing subjective biases and discrimination, increasing efficiency, and improving the identification of suitable candidates. However, it is important to ensure that the system is designed and evaluated carefully to avoid any biases or inaccuracies in its evaluation.

I.INTRODUCTION

The importance of nonverbal communication in job interviews cannot be overstated. Nonverbal behaviour conveys information about our personality and traits, which is in addition to the information conveyed through our speech. Since nonverbal communication is an unconscious process, it is difficult to manipulate and can have a significant impact on the outcome of the employment interview. Visual cues and audio cues are the two types of nonverbal communication. Visual cues or facial cues have a powerful effect on interview performance. Our physical appearance and how we dress are part of visual cues that are essential for an interview. Facial expressions such as smiling, neutral, and surprise during the conversation also contribute to visual cues. In addition, hand gestures, eye contact, and body orientation are also important visual cues during communication. Therefore, it is crucial to pay attention to nonverbal communication during an interview. Practicing positive body language, maintaining eye contact, and dressing appropriately can improve the overall impression that the interviewee creates.

The focus of this paper is on developing an automated computational analysis of cross-modal communication in job interviews, including visual, language, and prosody, in order to create a predictive expert system that can help predict the hire ability of candidates and rate different traits. The results of this analysis are displayed on a dashboard for the interviewer, providing a useful tool for hiring managers to make informed decisions.

This predictive system has potential applications in the screening and hiring of candidates for job interviews, and it can act as an expert support system for hiring managers to make appropriate decisions. The system can also be used for fully automated screening of candidates in online interviews, as shown in Figure 1.1, and this can help with the initial shortlisting of suitable candidates for further domain-specific evaluation. The development of this predictive system represents an important step towards automating the hiring process, reducing human bias, and ensuring a fair and objective evaluation of candidates. It has the potential to revolutionize the way we conduct job interviews and hire new employees, making the process more efficient and effective.

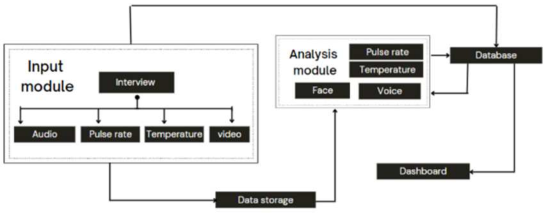


FIG:1.1 Block diagram of the proposed system

II. RELATED WORK

A. Multimodal Hierarchical Attention Neural Network [1]

Looking for Candidates Behaviour which Impact Recruiter's Decision [1] Automatic analysis of job interviews has gained in interest amongst academic and industrial research. The particular case of asynchronous video interviews allows the collection of a vast corpora of videos where candidates answer standardized questions in monologue videos, enabling the use of deep learning algorithms. On the other hand, state-of-the-art approaches still face some obstacles, among which the fusion of information from multiple modalities and the interpretability of the predictions. They study the task of predicting candidates' performance in asynchronous video interviews using three modalities (verbal content, prosody and facial expressions) independently or simultaneously, using data from real interviews which take place in real conditions. They propose a sequential and multimodal deep neural network model, called Multimodal Hire Net and compare this model to state-of-the-art approaches and show a clear improvement of the performance. Moreover, the architecture they proposed is based on an attention mechanism, which provides interpretability about which questions, moments and modalities contribute the most to the output of the network. While other deep learning systems use attention mechanisms to offer a visualization of moments with attention values, the proposed methodology enables an in-depth interpretation of the predictions by an overall analysis of the features of social signals contained in these moments.

B. Face Occlusion Recognition with Deep Learning in Security Framework for the IoT [2]

This paper designs a novel face occlusion recognition framework in the security scene of IOT, which is used to detect some crime behaviour. This designed framework utilizes the gradient and shape cues in a deep learning model, and it has been demonstrated to be robust for its superiority to detect faces with severe occlusion.

Their contributions contain three main aspects: Firstly, they present a new algorithm based on energy function for face detection; Secondly, they use the CNN models to create deep features of occluded face. Finally, to check whether the detected face is occluded, a novel sparse classification model with deep learning scheme is constructed. Statistical results demonstrate that, compared with the state of the arts, their algorithm is superior in both accuracy and robustness. Their designed head detection algorithm can achieve 98.89% accuracy rate even though there are various types of severe occlusions in faces, and their designed occlusion verification scheme can achieve 97.25% accuracy rate, at a speed of 10 frames per second.

C. Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing [3]

To avoid the complex process of explicit feature extraction in traditional facial expression recognition, a face expression recognition method based on a convolutional neural network (CNN) and an image edge detection is proposed. Firstly, the facial expression image is normalized, and the edge of each layer of the image is extracted in the convolution process. The extracted edge information is superimposed on each feature image to preserve the edge structure information of the texture image. Then, the dimensionality reduction of the extracted implicit features is processed by the maximum pooling method. Finally, the expression of the test sample image is classified and recognized by using a SoftMax classifier. To verify the robustness of this method for facial expression recognition under a complex background, a simulation experiment is designed by scientifically mixing the Fer-2013 facial expression database with the LFW data set. The experimental results show that the proposed algorithm can achieve an average recognition rate of 88.56% with fewer iterations, and the training speed on the training set is about 1.5 times faster than that on the contrast algorithm.

D. Facial Expression Recognition using Local Gravitational Force [4]

Descriptor based Deep Convolutional Neural Network [4]. In this paper, a deep-learning-based scheme is proposed for identifying the facial expression of a person like neutral (NE), anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA), and surprise (SU) from still images and videos. The proposed method consists of two parts. It finds out the local features from face images using a local gravitational force descriptor is fed into a novel deep convolutional neural networks model (DCNN). The DCNN has two branches. The first branch explores geometric features such as edges, curves, and lines whereas holistic features are extracted by the second branch which can differentiate one expression from others. To estimate the final prediction of seven basic expressions, the score-level fusion technique

E. Predicting Personality Using Answers to Open-Ended Interview Questions [5]

This paper shows that textual content of answers to standard interview questions related to past behaviour and situational judgement can be used to reliably infer personality traits. Paper used data from over 46,000 job applicants who completed an online chat interview that also included a personality questionnaire based on the six-factor HEXACO personality model to self-rate their personality. Using natural language processing (NLP) and machine learning methods built a regression model to infer HEXACO trait values from textual content. The compared performance of five different text representation methods and found that term frequency-inverse document frequency (TF-IDF) with Latent Dirichlet

Allocation (LDA) topics performed the best with an average correlation of r = 0.39.As a comparison, a large study of Facebook messages based inference of Big 5 personality found an average correlation of r = 0.35 and IBM's Personality Insights service built using twitter text data reports an average correlation of r = 0.31.This paper further validates the model with a group of 117 volunteers who used an agreement scale of yes/no/maybe to rate the individual trait descriptors generated based on the model outcomes. On average, 87.83% of the participants agreed with the personality description given for each of the six traits. The ability of algorithms to objectively infer a candidate's personality using only the textual content of interview answers presents significant opportunities to remove the subjective biases involved in human interviewer judgement of candidate personality.

F. Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing [6]

Currently, data generated by smart devices connected through the Internet is increasing relentlessly. Deep learning and edge computing are the emerging technologies, which are used for efficient processing of huge amounts of data with distinct accuracy. In this world of advanced information systems, one of the major issues is authentication. Face recognition is considered as one of the most reliable solutions. Usually, for face recognition, scale-invariant feature transforms (SIFT) and speeded up robust features (SURF) have been used by the research community. This paper proposes an algorithm for face detection and recognition based on convolution neural networks (CNN), which outperform the traditional techniques. In order to validate the efficiency of the proposed algorithm, a smart classroom for the student's attendance using face recognition has been proposed. The face recognition system is trained on publicly available labelled faces in the wild (LFW) dataset. The system can detect approximately 35 faces and recognizes 30 out of them from the single image of 40 students.

The proposed system achieved 97.9% accuracy on the testing data. Moreover, generated data by smart classrooms is computed and transmitted through an IoT-based architecture using edge computing. A comparative performance study shows that our architecture outperforms in terms of data latency and real-time response.

III. DATASET

A. FER2013(Facial Expression Recognition 2013 dataset)

Facial Expression Recognition 2013 (FER-2013) dataset was prepared in Challenges in Representation Learning: Facial Expression Recognition Challenge, which is hosted by Kaggle. The FER-2013 database has seven facial expression categories (e.g., angry, disgust, fear, happy, sad, surprise, and neutral) and three different sets such as a training set (28.709 images), validation set (3.589 images), and test set (3.589 images). All images in this dataset are grayscale with 48×48 pixels, thus corresponding to faces with various poses and illumination, where several faces are covered by hand, hair, and scarves.

FIG:3.1: FER 2013 Dataset Sample

IV. METHODOLOGY

A. Python

In addition, Python has a vast and active community of developers who continuously contribute to the improvement of the language through packages and modules. This makes it easier to learn and use, as there are numerous resources available online for beginners and experts alike. Python is also widely used in data science and machine learning applications due to its powerful libraries and frameworks, such as NumPy, Pandas, and TensorFlow. This makes it a popular choice for businesses and industries that require data analysis and automation.

Overall, Python is a versatile and efficient programming language that can be used for a variety of applications, making it a preferred choice for developers across industries. It's simple syntax, platform independence, and active community make it a popular language for beginners and advanced programmers alike.

B. TensorFlow

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine-learning applications using various tools, libraries, and community resources. Currently, the most famous deep learning library in the world is Google's TensorFlow. TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensor board. This tool is helpful to debug the program. Finally, TensorFlow is built to be deployed at scale. It runs on CPU and GPU.

TensorFlow architecture works in three parts:

- Pre-processing the data
- Build the model
- Train and estimate the model

C. Keras

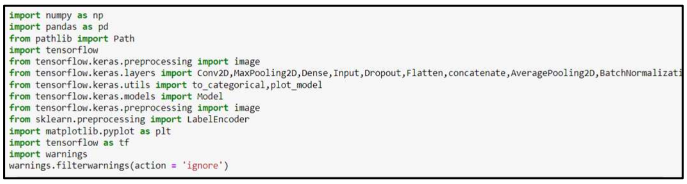
Keras simplifies this process by providing an intuitive interface that allows users to quickly construct neural networks by defining the model's architecture and the parameters that define it. The modular structure of Keras also makes it easy to add new layers and functionalities to the existing model, making it a versatile tool for developing complex deep learning models. Additionally, Keras provides a high-level API that abstracts away the details of the underlying machine learning library, making it easier for users to work with different backends without having to learn the underlying code. Overall, Keras provides a powerful yet accessible framework for creating deep learning models that is ideal for developers and researchers of all skill levels.

D. Model

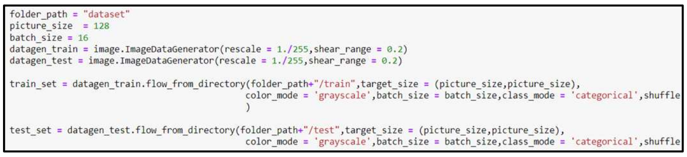
Here we use Functional API rather than a Sequential model since we are training this model to predict between 7 different emotions like sad, happy, anger, surprise, neutral, disgust, and fear, and have 7 different layers.

The functional API in tf.Keras is an alternative way of building more flexible models, including formulating a further complex model. On the other hand, the Functional API allows for more flexibility in designing models. With the Functional API, you can define a graph of layers, which can be connected in more complex ways than just one after the other. Additionally, you can create models with multiple inputs or outputs, and share layers between models. This added flexibility comes with some additional complexity in the construction of the models, as you must explicitly define the input and output layers and how they are connected. However, the Functional API offers a more versatile approach to model design and is recommended for more complex models.

In functional API, you can design models that produce a lot more versatility. You can undoubtedly fix models where layers relate to more than just the preceding and succeeding layers. You can combine layers with several other layers. As a consequence, producing heterogeneous networks such as Siamese networks and residual networks becomes feasible.



```
import numpy as np
import pandas as pd
from pathlib import Path
import tensorflow
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dense,Input,Dropout,Flatten,concatenate,AveragePooling2D,BatchNormalizati
from tensorflow.keras.utils import to_categorical,plot_model
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import tensorflow as tf
import warnings
warnings.filterwarnings(action = 'ignore')
```



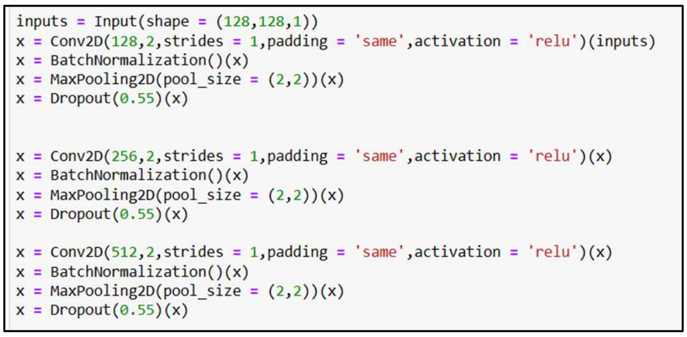
```
folder_path = "dataset"
picture_size  = 128
batch_size = 16
datagen_train = image.ImageDataGenerator(rescale = 1./255,shear_range = 0.2)
datagen_test = image.ImageDataGenerator(rescale = 1./255,shear_range = 0.2)

train_set = datagen_train.flow_from_directory(folder_path+"/train",target_size = (picture_size,picture_size),
                                              color_mode = 'grayscale',batch_size = batch_size,class_mode = 'categorical',shuffle
                                              )

test_set = datagen_test.flow_from_directory(folder_path+"/test",target_size = (picture_size,picture_size),
                                              color_mode = 'grayscale',batch_size = batch_size,class_mode = 'categorical',shuffle
```

In the code above,

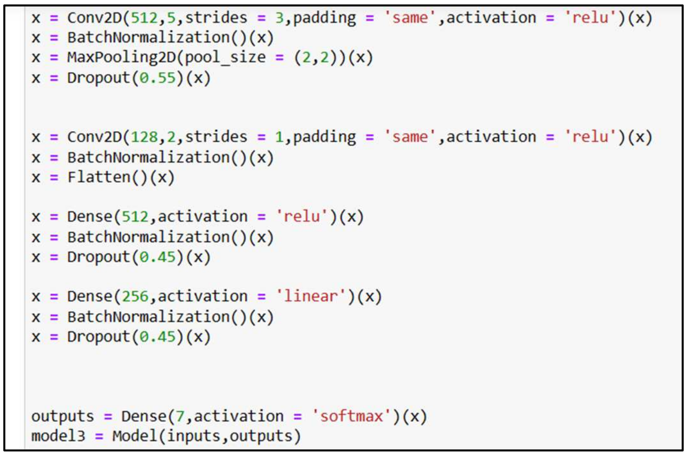
- We distributed these two groups as a train set and test set and distributed the labels and the inputs.
- It is a good practice to normalize our data as it is constantly required in deep learning models. We can accomplish this by dividing the RGB codes by 255.
- batch size: a hyperparameter that determines the number of samples to run through before refreshing the internal model parameters



```
inputs = Input(shape = (128,128,1))
x = Conv2D(128,2,strides = 1,padding = 'same',activation = 'relu')(inputs)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size = (2,2))(x)
x = Dropout(0.55)(x)


x = Conv2D(256,2,strides = 1,padding = 'same',activation = 'relu')(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size = (2,2))(x)
x = Dropout(0.55)(x)

x = Conv2D(512,2,strides = 1,padding = 'same',activation = 'relu')(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size = (2,2))(x)
x = Dropout(0.55)(x)
```



```
x = Conv2D(512,5,strides = 3,padding = 'same',activation = 'relu')(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size = (2,2))(x)
x = Dropout(0.55)(x)


x = Conv2D(128,2,strides = 1,padding = 'same',activation = 'relu')(x)
x = BatchNormalization()(x)
x = Flatten()(x)

x = Dense(512,activation = 'relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.45)(x)

x = Dense(256,activation = 'linear')(x)
x = BatchNormalization()(x)
x = Dropout(0.45)(x)



outputs = Dense(7,activation = 'softmax')(x)
model3 = Model(inputs,outputs)
```

In the code above,

- The model holds an input layer, 7 hidden layers, and a product layer with 1 output.

- Rectified linear activation functions are applied in all hidden layers, and a SoftMax activation function is adopted in the product layer for binary classification.

- All layers of the neural network will collapse into one if a linear activation function is used. No matter the number of layers in the neural network, the last layer will still be a linear function of the first layer. So, essentially, a linear activation function turns the neural network into just one layer.

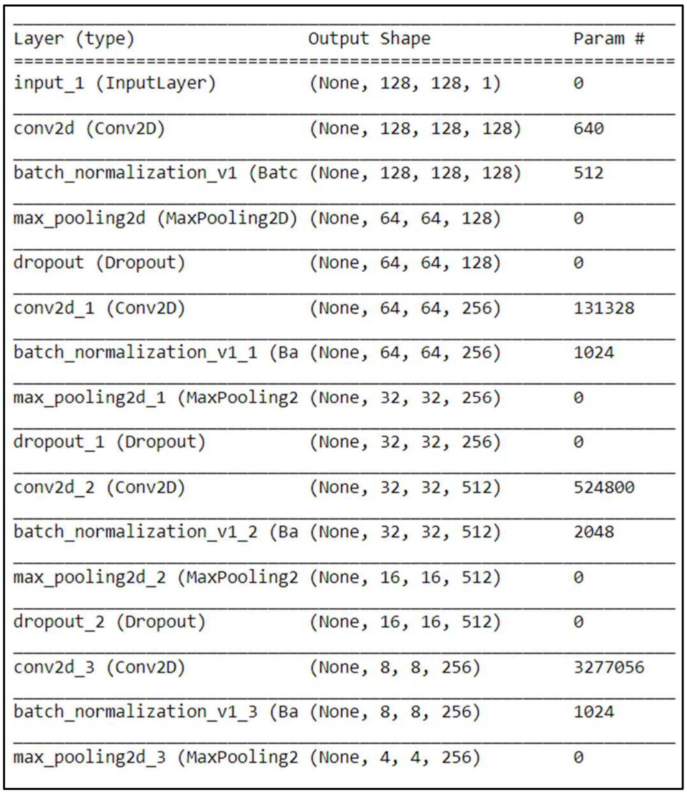
- inputs: variable represents a need to plan and style a standalone Input layer that designates input data. The input layer accepts a shape argument that is a tuple that describes the dimensions of the input data.

- kernel size: relates to the dimensions (height x width) of the filter mask. Convolutional neural networks (CNN) are essentially a pile of layers marked by various filters' operations on the input. Those filters are ordinarily called kernels.

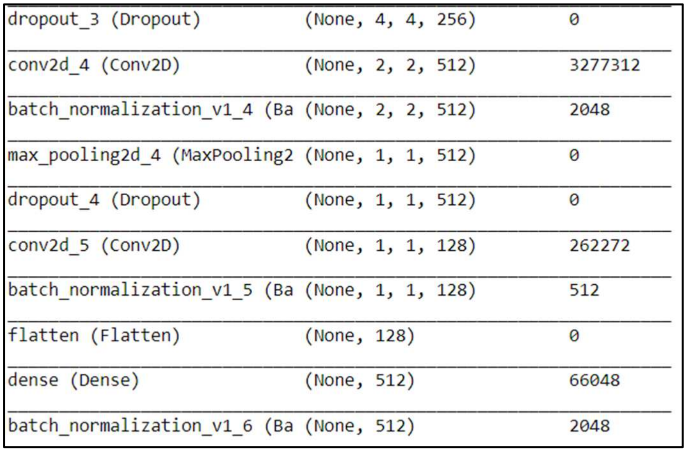
- filter: is expressed by a vector of weights among which we convolve the input.

- Dropout: a process where randomly picked neurons are neglected throughout training. This implies that their participation in the activation of downstream neurons is temporally dismissed on the front pass.

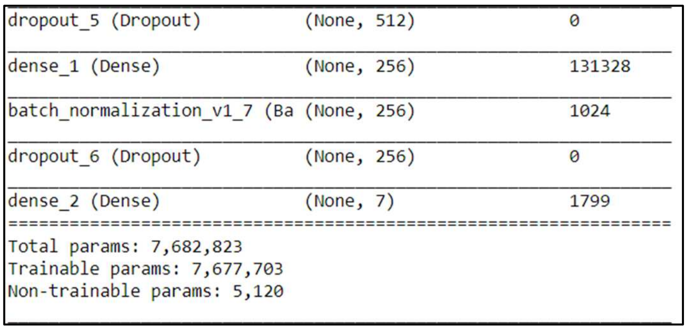
TensorFlow presents a model class that you can practice to generate a model from your developed layers. It demands that you only define the input and output layers—mapping the structure and model graph of the network architecture.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 128, 128, 1)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 128, 128, 128)     640
_________________________________________________________________
batch_normalization_v1 (Batc (None, 128, 128, 128)     512
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 64, 128)       0
_________________________________________________________________
dropout (Dropout)            (None, 64, 64, 128)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 256)       131328
_________________________________________________________________
batch_normalization_v1_1 (Ba (None, 64, 64, 256)       1024
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 256)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 512)       524800
_________________________________________________________________
batch_normalization_v1_2 (Ba (None, 32, 32, 512)       2048
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 512)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 512)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 256)         3277056
_________________________________________________________________
batch_normalization_v1_3 (Ba (None, 8, 8, 256)         1024
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 256)         0
```

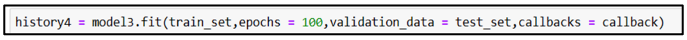


```
dropout_3 (Dropout)          (None, 4, 4, 256)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 2, 2, 512)         3277312
_________________________________________________________________
batch_normalization_v1_4 (Ba (None, 2, 2, 512)         2048
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 1, 512)         0
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 1, 512)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 1, 128)         262272
_________________________________________________________________
batch_normalization_v1_5 (Ba (None, 1, 1, 128)         512
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 512)               66048
_________________________________________________________________
batch_normalization_v1_6 (Ba (None, 512)               2048
```



```
dropout_5 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
batch_normalization_v1_7 (Ba (None, 256)               1024
_________________________________________________________________
dropout_6 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 1799
=================================================================
Total params: 7,682,823
Trainable params: 7,677,703
Non-trainable params: 5,120
_________________________________________________________________
```

Lastly, we train the model,



```
history4 = model3.fit(train_set,epochs = 100,validation_data = test_set,callbacks = callback)
```

This section details down the process of analysis on the recorded interviews and accordingly extracted features. Subsequently, we present the methods required to extract the audio, visual, pulse rate and temperature from the recorded data.

1) Audio cues: The major concern while extracting audio cue features is on the prosodic features. The preliminary step is to detect the individual who is speaking during the interview, the interviewer or the interviewee. Whilst during this process, we also need to segment the speech and non-speech chunks. Secondly, we need to make sure that the existence of any noise should be eliminated from the audio, if present. For audio cues, we extract the prosodic features of Pitch, Band energy, and the Speech Rate respectively

The tool utilized for this purpose is PRAAT, it is a software program used for speech analysis and synthesis. It was developed by Paul Boersma and David Weenink at the University of Amsterdam and is available for free download on their website. It provides a wide range of tools for analysing and manipulating audio recordings of speech, including tools for recording and editing audio files,Analyzing speech sounds and features (such as pitch, formant frequencies, and intensity),creating and editing spectrograms and waveforms, segmenting and labelling audio files, synthesizing speech sounds and creating speech stimuli. Steps used to extract audio cues using PRAAT are firstly,PRAAT's segmentation tool is used to divide the audio file into smaller segments based on pauses, changes in pitch or amplitude, or other criteria. Once the audio file is segmented, you can label the segments based on the verbal or nonverbal cues you want to extract. The analysis tool is used to extract data from each segment of the interview. Once you have extracted the data, you can analyse it to identify patterns and trends in the audio cues.

2) Visual cues: Visual cues are the facial features of the interviewee during the interview. For every video frame, facial features of the interviewee are extracted. In an interview, visual cues refer to the nonverbal cues that a person exhibits through their body language and facial expressions. The first step in this is to detect the face itself in the frames. The Haar Cascade classifier for face detection can be used to perform this task of face detection. Different facial cues extracted are facial expression and gaze. There are different methods to detect facial expressions. Machine Learning is one of the approaches to detect facial expressions. Since it's a classification problem we can classify facial images using ML into different expressions like angry, disgust, fear, happy, sad, surprise, and neutral.

Next dlib library is used to detect facial landmarks. It is a popular computer vision library that includes a facial landmark detection algorithm. The algorithm uses a machine learning approach to detect and localize key facial features, such as the eyes, nose, mouth, and eyebrows. By detecting the position of key facial features, machine learning algorithms can infer important information about a person's emotions, identity, and other aspects of nonverbal communication. The algorithm uses a machine learning approach to detect and localize key facial features, such as the eyes, nose, mouth, and eyebrows.

The dlib library includes a pre-trained facial landmark detector that is capable of estimating the location of 68 coordinates that correspond to key facial features. These coordinates include points on the eyebrows, eyes, nose, mouth, and jawline.

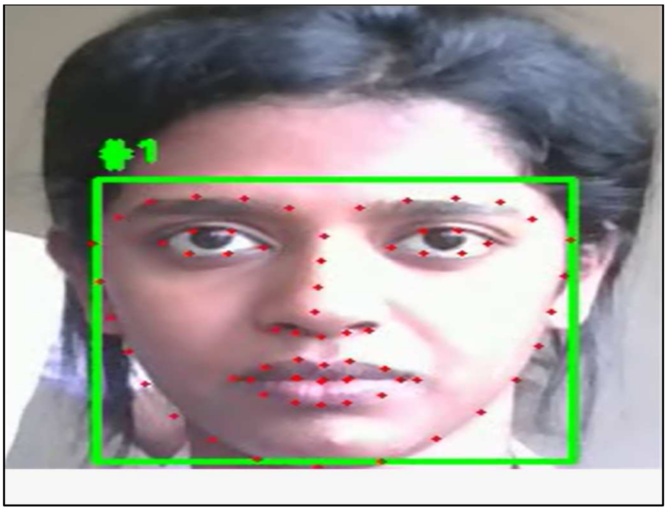


FIG: 4.1: Dots plotted on a facial Figures to help understand the production of various expressions according to the movements of these points.

3) Sensor:

MAX30102 Heart Rate Sensor with ESP32 Wi-Fi module is used to calculate the BPM and temperature. The MAX30102 is a pulse oximeter and heart rate sensor module that includes an on-chip temperature sensor for measuring the temperature. The MAX30102 sensor is the further optimized version of MAX30100 sensor. This sensor consists of two LEDs, a photodetector and two LDO regulators. It is easily used with microcontrollers such as Arduino, ESP32, etc. to generate an efficient heartbeat. Below you can view the diagram of MAX30102 Module:

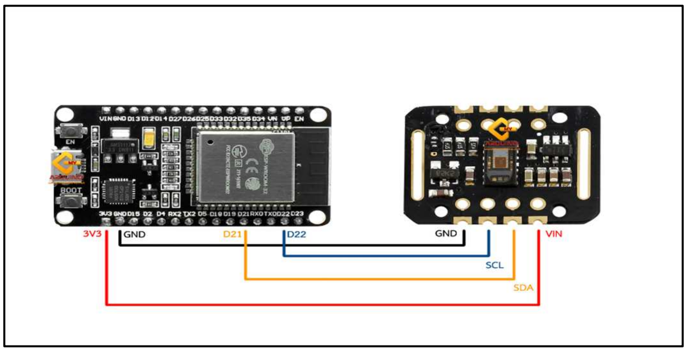


FIG:4.2: MAX30102 Heart Rate Sensor with ESP32 Wi-Fi module

The working of this sensor is checked by placing a human finger in front of this sensor. When a finger is placed in front of this sensor then the reflection of infrared light is changed based on the volume of blood change inside capillary vessels. This means during the heartbeat, the volume of blood in capillary vessels will be high and then will be low after each heartbeat. So, by changing this volume, the LED light is changed. This change of the LED light measures the heartbeat rate of a finger. This phenomenon is known as "Photoplethysmogram." Read the temperature value from the MAX30102. The temperature data is stored in the temperature register (TEMP_DATA) of the MAX30102.

ESP32 Wi-Fi / Bluetooth Wireless Microcontroller: The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities. The module is programmed using various software development tools, including the Arduino IDE, the Espressif IoT Development Framework (ESP-IDF), and Micro Python.

V. RESULT AND ANALYSIS

A. After training

After training the model for 100 epochs with FER 2013 dataset we were able to obtain an accuracy percentage of 70

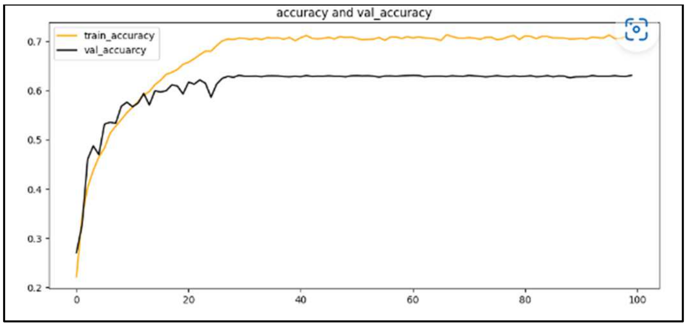


FIG:5.1: Model Accuracy and Validation Accuracy

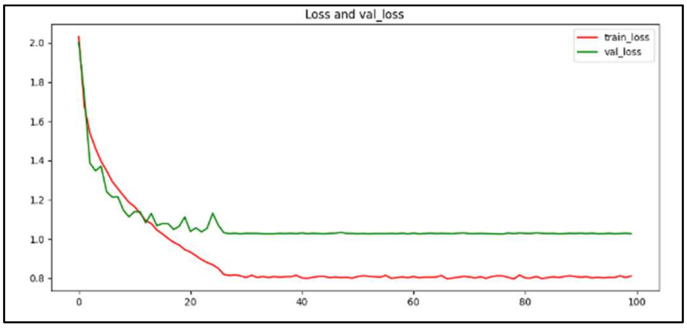


FIG:5.2: Model Loss **and** Validation Loss

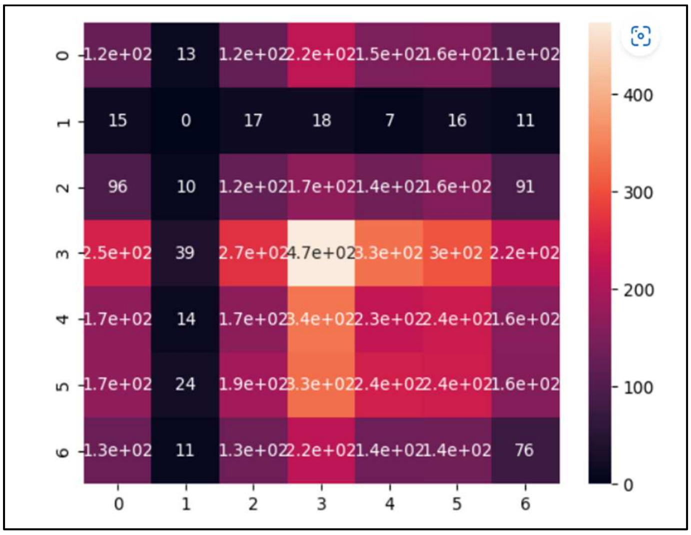
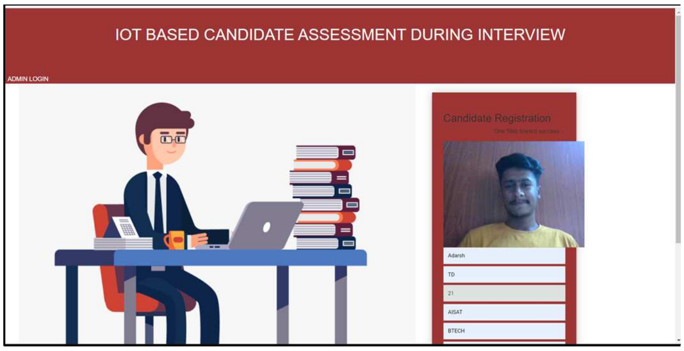


FIG:5.3: Confusion Matrix

B. Screenshots of the project

Initially HR has to register the candidate by entering the details of the candidate.



Candidate Registration

One Step toward success

Adarsh

TD

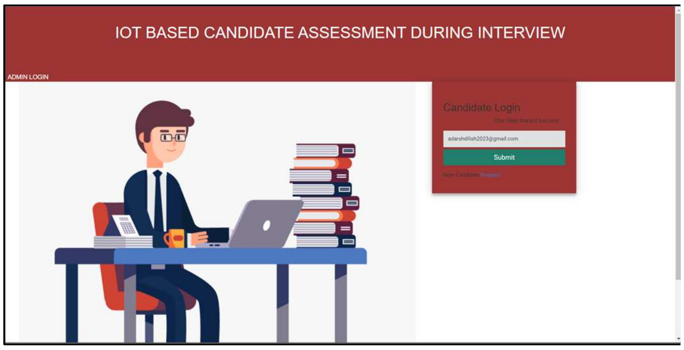
21

AISAT

BTECH

FIG:5.4: Candidate registration.

After candidate registration, HR can begin the interview procedures by login using candidate email id.



Candidate Login

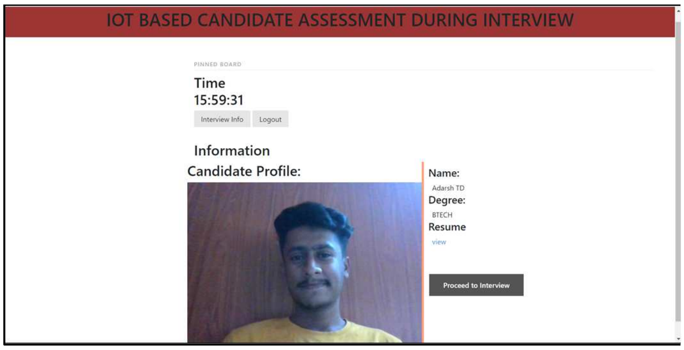
One Step toward success

adarshdilish2023@gmail.com

New Candidate Register

FIG:5.5: Candidate login.

Once HR has logged in using the candidate email id, it will direct to the candidate profile in which all the details of the candidate is shown which is entered by the HR during candidate registration. By clicking 'Proceed to interview button' HR can begin the interview of the candidate.



Time
15:59:31

Information

Candidate Profile:

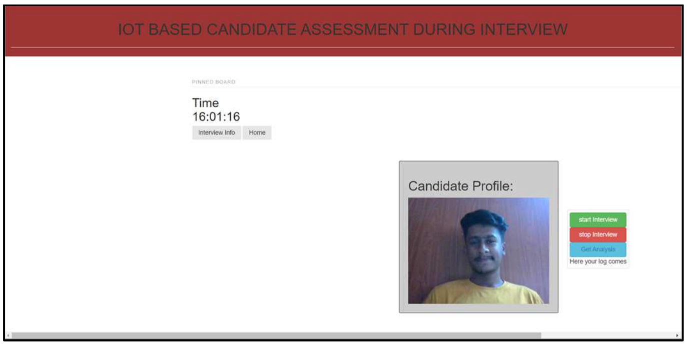
Name:
Adarsh TD

Degree:
BTECH

Resume

FIG:5.6: Candidate profile

After selecting the 'Proceed to interview' button it will direct to a page in which three buttons are available: start, stop and get analysis. By clicking the 'Start' button, the camera is on and starts recording the interview of the candidate. At this time the candidate is asked to place his finger on the sensor in order to get the pulse rate. It also takes the temperature of the candidate. Once the interview is completed 'Stop' button is selected in order to turn off the camera and stop recording. 'Get analysis' button is selected to get the results.



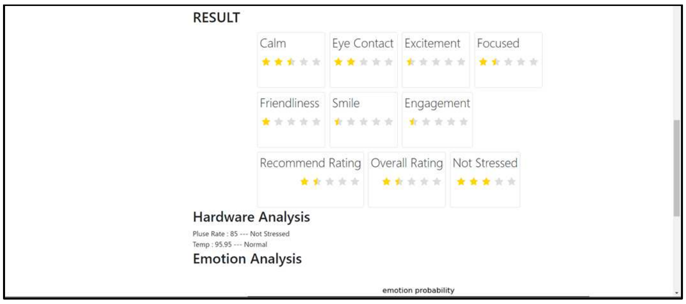
Time
16:01:16

Candidate Profile:

Here your log comes

FIG:5.7: Buttons for the interview procedures.

Results of the interview are displayed on the dashboard in the form of a five-star rating. Pulse rate and temperature of the candidate is also shown.



RESULT

Calm ★★⯨☆☆
Eye Contact ★★☆☆☆
Excitement ⯨☆☆☆☆
Focused ★⯨☆☆☆
Friendliness ★☆☆☆☆
Smile ⯨☆☆☆☆
Engagement ⯨☆☆☆☆
Recommend Rating ★⯨☆☆☆
Overall Rating ★⯨☆☆☆
Not Stressed ★★★☆☆

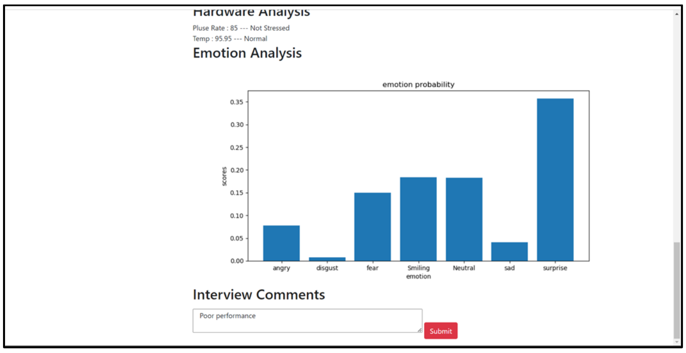
Hardware Analysis

Pluse Rate : 85 --- Not Stressed
Temp : 95.95 --- Normal

Emotion Analysis

FIG:5.8: Results in five-star rating along with hardware analysis.

Emotions of the candidate throughout the interview is depicted using a bar graph. Comment session is provided for the HR to comment what he/she thinks about the candidate. It helps to obtain the ground truth.



Hardware Analysis

Pluse Rate : 85 --- Not Stressed
Temp : 95.95 --- Normal

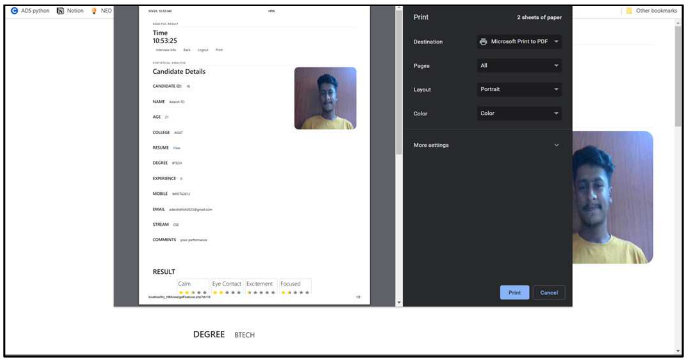
Emotion Analysis

Interview Comments

Poor performance

FIG:5.9: Emotion bar graph and comment session.

By combining all these results HR is able to hire the candidate based on his performance. Print option is available to print the results of the candidate.



ANALYSIS RESULT

Time
10:53:25

STATISTICAL ANALYSIS

Candidate Details

CANDIDATE ID: 18

NAME Adarsh TD

AGE 21

COLLEGE AISAT

RESUME View

DEGREE BTECH

EXPERIENCE 0

MOBILE 9495762612

EMAIL adarshdilish2023@gmail.com

STREAM CSE

COMMENTS poor performance

RESULT

Calm	Eye Contact	Excitement	Focused
★★★★★	★★★★★	★★★★★	★★★★★

Print — 2 sheets of paper

Destination: Microsoft Print to PDF

Pages: All

Layout: Portrait

Color: Color

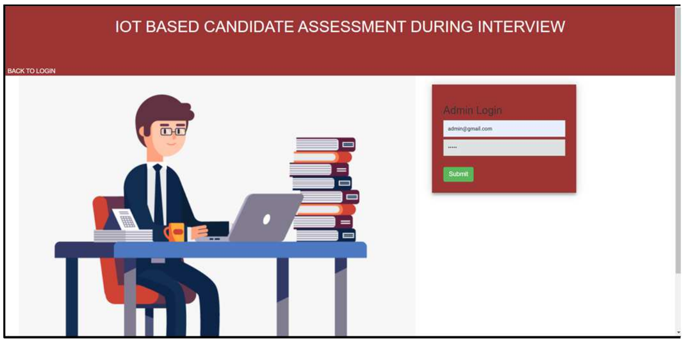
More settings

Print Cancel

DEGREE BTECH

FIG:5.10: Printing the results.

List of candidate's option is available in admin portal. HR can login in the admin portal and can view the list of candidates in which details of all candidates who have attended the interview is available.



IOT BASED CANDIDATE ASSESSMENT DURING INTERVIEW

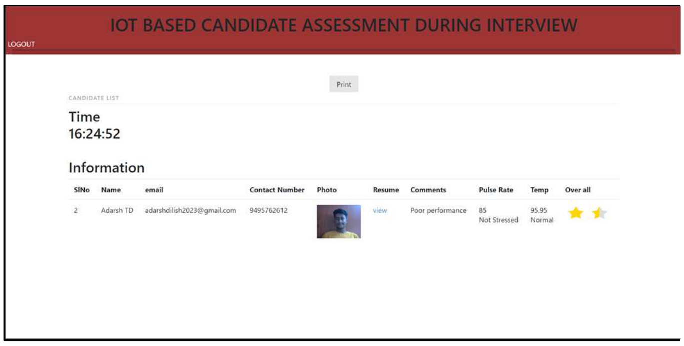
Admin Login

admin@gmail.com

•••••

Submit

FIG:5.11: Admin login



IOT BASED CANDIDATE ASSESSMENT DURING INTERVIEW

CANDIDATE LIST

Time
16:24:52

Information

SlNo	Name	email	Contact Number	Photo	Resume	Comments	Pulse Rate	Temp	Over all
2	Adarsh TD	adarshdilish2023@gmail.com	9495762612		view	Poor performance	85 Not Stressed	95.95 Normal	★★

FIG:5.12: List of candidates

VI. FUTURE SCOPE

- Can be used in military, upsc and other high profile job interviews.
- Can be used while questioning the culprit by adding a lie detection system. Lie detection can also be added while performing high profile job interviews.

VII. CONCLUSION

The system developed here uses a multimodal approach, we have used Audio and Video cues for preparing the prediction model. We are also taking the pulse rate and temperature of the candidate. We cannot prioritize only one from both verbal and nonverbal signatures. Both are equally important when it comes to HR interviews. One of the most important cases to consider is that the existing systems do not propose a multimodal nature existence. They are either dependent on audio or on video cues for analysis. The proposed system produced a variety of results, from a single personality trait to the overall hire ability recommendation of the candidate.

This modal simplifies the task of an organization for conducting HR based interviews, by allowing the interview process to be automated, carrying out the tasks of evaluating the verbal and nonverbal signatures of the interview candidate.

REFERENCES

[1]Multimodal Hierarchical Attention Neural Network: Looking for Candidates Behaviour which Impact Recruiter's Decision Le ́o Hemamou, Arthur Guillon, Jean-Claude Martin,and Chloe ́ Clavel,IEEE 2021.

[2]Face Occlusion Recognition With Deep Learning in Security Framework for the IoTLIBMAO 1, FUSHENG SHENG 2, AND TAO ZHANG 1,31 Department of Computer Science and Technology, Jiangnan University, Wuxi 214122 China,2019.

[3]A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing Hongli Zhang1, Alireza Jolfaei2, and Mamoun Alazab3,IEEE 2020 .

[4]Facial Expression Recognition using Local Gravitational Force Descriptor based Deep Convolution Neural Networks Karnati Mohan, Ayan Seal, Senior Member IEEE, Ondrej Krejcar, Anis Yazidi, Senior Member IEEE ,2020.

[5] Predicting Personality Using Answers to Open-Ended Interview Questions Madhura Jayaratne 1,2, (Member, IEEE), AND Buddhi Jayatilleke,2020.

[6] Deep Unified Model For Face Recognition Based on Convolution Neural Network and Edge Computing Muhammad Zeeshan Khan1, Saad Harous 2, Saleet Ul

Hassan1, Muhammad Usman Ghani Khan1, Razi Iqbal 3, (Senior Member, IEEE),And Shahid Mumtaz.

[7] Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. Video description: A survey of methods, datasets, and evaluation metrics. ACM Computing Surveys, 52(6):1–28, 2019

[8] C. L. Lisetti and D. J. Schiano, "Automatic facial expression interpretation", Facial Information Processing, vol. 8, no. 1, pp. 185-235, 2000.

[9] J. P. Skelley, "Experiments in expression recognition", thesis, 2005.

[10] A. Jaiswal, A. Krishnama Raju and S. Deb, "Facial Emotion Detection Using Deep Learning", 2020 International Conference for Emerging Technology (INCET), pp. 1-5, 2020.

[11] K. Yang, C. Wang, Z. Sarsenbayeva, B. Tag, T. Dingler, G. Wadley, et al., "Benchmarking commercial emotion detection systems using realistic distortions of facial image datasets", The Visual Computer, vol. 37, no. 6, pp. 1447-1466, 2020.

[12] S. Jerritta, M. Murugappan, R. Nagarajan and K. Wan, "Physiological signals based human emotion Recognition: a review", 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, pp. 410-415, 2011.

[13] D. Sen, S. Datta and R. Balasubramanian, "Facial emotion classification using concatenated geometric and textural features", Multimedia Tools and Applications, vol. 78, no. 8, pp. 10287-10323, 2018.

[14] D. R. Faria, M. Vieira, F. C. C. Faria and C. Premebida, "Affective facial expressions recognition for human-robot interaction", 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 805-810, 2017.