Dissertation on

DETECTION OF FACIAL EMOTION OF LEARNERS' USING FEATURE EXTRACTION IN DESIGNING OF AN ADAPTIVE TUTORING SYSTEM

Thesis submitted towards partial fulfilment
of the requirements for the degree of

Master of Technology in IT (Courseware Engineering)

Submitted by
MahendraNath Murmu

EXAMINATION ROLL NO.: M4CWE22012
UNIVERSITY REGISTRATION NO.: 136201 of 2016-2017

Under the guidance of
DR. SASWATI MUKHERJEE

School of Education Technology
Jadavpur University

Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032
India

2022

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **"DETECTION OF FACIAL EMOTION OF LEARNERS' USING FEATURE EXTRACTION IN DESIGNING OF AN ADAPTIVE TUTORING SYSTEM"** is a bonafide work carried out by **MahendraNath Murmu** under our supervision and guidance for partial fulfillment of the requirements for the degree of Master of Technology in IT (Courseware Engineering) in School of Education Technology , during the academic session 2021-2022.

SUPERVISOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DIRECTOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DEAN - FISLM
Jadavpur University,
Kolkata-700 032

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF APPROVAL **

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

Committee of final examination for evaluation of Thesis

** Only in case the thesis is approved.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of his **Master of Technology in IT (Courseware Engineering)** studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

NAME : MAHENDRANATH MURMU

EXAMINATION ROLL NUMBER : M4CWE22012

THESIS TITLE : DETECTION OF FACIAL EMOTION OF LEARNERS' USING FEATURE EXTRACTION IN DESIGNING OF AN ADAPTIVE TUTORING SYSTEM

SIGNATURE: DATE:

ACKNOWLEDGEMENT

I feel fortunate while presenting this dissertation at **School of Education Technology, Jadavpur University, Kolkata,** in the partial fulfillment of the requirement for the degree of **M.Tech in IT (Courseware Engineering).**

I hereby take this opportunity to show my gratitude towards my mentor, **Dr. Saswati Mukherjee,** who has guided and helped me with all possible suggestions, support, aspiring advice and constructive criticism along with illuminating views on different issues of this dissertation which helped me throughout my work.

I would like to express my warm thanks to **Prof. Ranjan Parekh, Director of School of Education Technology** for his timely encouragement, support and advice. I would also like to thank **Prof. Matangini Chattopadhyay and Mr. Joydeep Mukherjee** for their constant support during my entire course of work. My thanks and appreciation goes to my classmates from M.Tech in IT (Courseware Engineering) and Master in Multimedia Development. I do wish to thank all the departmental support staffs and everyone else who has different contributions to this dissertation. Finally, my special gratitude to my parents who have invariably sacrificed and supported me and made me achieves this height.

Regards

…….

MAHENDRANATH MURMU
Examination Roll No : M4CWE22012
Registration No: 136201 of 2016 – 2017
M.Tech in IT (Courseware Engineering)
School of Education Technology
Jadavpur University
Kolkata :
700032

Date:

Contents

Executive Summary

The present work proposes to design an adaptive tutoring system using different types of feature extraction. In this model, images are captured through a webcam, which are sent to the Facial Emotion Recognition (FER) system for detection. Different types of feature extraction methods are used to predict the emotion, improve accuracy. Based on the predicted outcome, the adaptive tutoring system is designed in such a way that the learning materials can be modified and adapted as per the learners' emotions.

The proposed work has successfully predicted the learners' various emotions using different types of feature extraction methods.

1. Introduction

1.1 Problem statement

DETECTION OF FACIAL EMOTION OF LEARNERS' USING FEATURE EXTRACTION IN DESIGNING OF AN ADAPTIVE TUTORING SYSTEM

1.2 Objective

The objectives of proposed work are:

i. Design an adaptive tutoring system that would detect the facial emotions of learners and for changing the learning materials as per the emotion for better comprehension.

ii. Different types of feature extraction methods are used in emotion detection for analysing the detection accuracy.

2. Background concept

2.1. Intelligent Tutoring System

It is a computer based teaching system that provides feedback automatically according to a learner's performance. It doesn't require any human teachers.
The system mostly has four essential components [1]. Fig.1. depicts the essential components of ITS.

a. The Expert Knowledge Model
b. The Student Model
c. The Tutoring Model
d. The User Interface Model

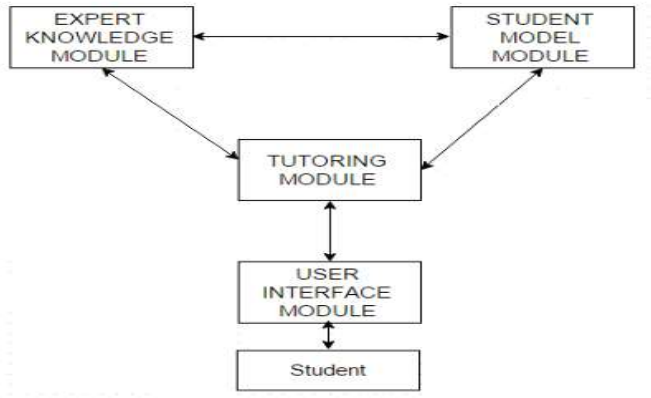


Fig.1.The essential components of ITS

a. **The Expert Knowledge Model -** The Expert Knowledge, also known as the expert model or domain expert, reflects the concepts, facts, problem-solving techniques, and rules of the specific area to be taught and provides ITSs with an understanding of what they are teaching.

b. **The Student Model -** The student model keeps information about the student. The student submits some assignments. It knows about the student's strong and weak knowledge zones in that topic based on the submitted assignment. It is updated throughout the learning process.
c. **The Tutoring Model -** It analyses data from the domain and student models to determine the tutoring system's methods and actions.

During the problem-solving process, if any student requests to get help, for the next step, the system gives hints for the next step. The system detects when the learner deviates from the learning process. It also provides quick feedback to the student.

d. **The User Interface model -** That model represents the interaction between user and device and also how the system responds.

Designing a new model for the tutoring system. It also keeps track of a student's emotional condition. According to the emotion prediction, the system will change its teaching contents.

2.2. Facial Emotion Recognition

Facial Emotion Recognition (FER) is a technology that analyses facial expressions in static images and videos to reveal information about an individual's emotional state. It is consists of several parts, like -

2.2.1. Preprocessing

This technique is applied to an image to reduce its noise. In this process, the unwanted pixels are removed from the image by using a cropping operation. The unwanted parts are the hair and neck, and the essential parts are the lips and eyes.

2.2.2. Convolution Neural Network

CNNs [2] are a form of deep learning algorithm that uses a grid-like structure to process input. CNNs are particularly useful for finding patterns in images to recognize objects, faces, and scenes. The Convolution Neural Network has five layers for finding the patterns in images. For Emotion Recognition, these layers are detect the face key and facial landmarks. Fig.2. depicts the convolutional neural network structure (CNNs).

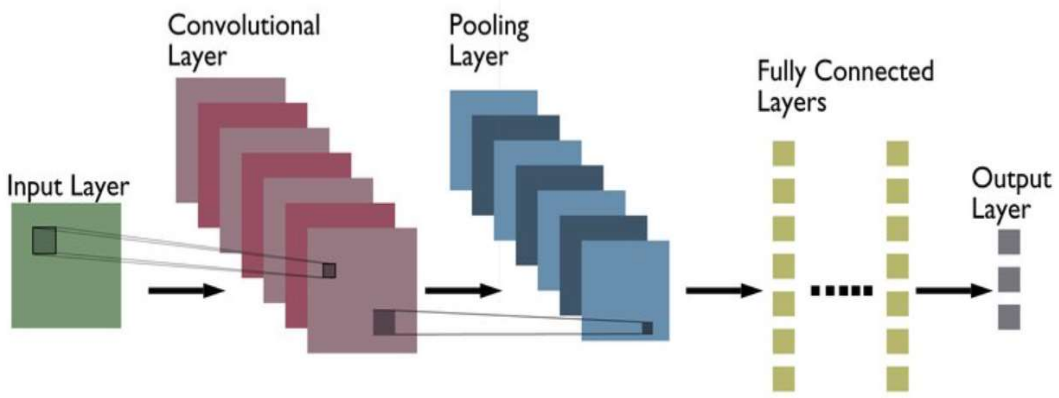


Fig.2. CNN structure

a. **Input layer -** In this layer, input images are placed, and convert it into single-layer 2D image (gray scale), 2D 3-channel image (RGB colour) or 3D.

b. **Convolutional layer -** In convolutional neural networks (CNN), the convolution layer is the main building block. It contains a set of filters. Those sets of filters are called kernels, which are applied to an input image. The convolutional layer produces a component map. The component maps are stacked to generate more complicated models that can learn more intricate information from images. The convolution operation is shown in Fig.3.

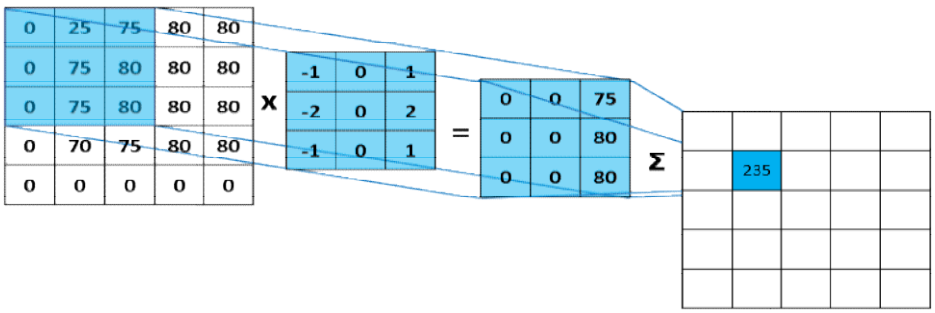


Fig.3. Convolution Operation

c. **ReLu layer -** The ReLU (Rectified Linear Unit) layer is one of the most popular processes in convolution neural networks, which is used in neural networks. The ReLu layer is applied after the convolutional layer and before the pooling layer. It replaces all negative pixel values in the feature map by zero.

Mathematically, it is defined as $y = max(o,x)$ Equation.1

The mathematically graph of ReLu layer is shown in Fig. 4. It gives zero output for all negative value and keeps the positive value.

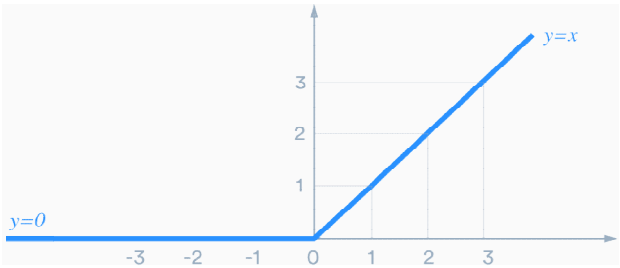


Fig.4. ReLu Operation

d. Pooling layer - The spatial dimension of the input images is reduced by pooling layers, making them easier to analyse and needing less memory. Pooling layers also help to reduce the number of parameters.

Mainly two types of pooling are present: **maximum pooling** and **average pooling**.

Maximum pooling extracts the greatest value from each feature map, where average pooling takes the average value. Pooling layers are generally worked after ReLu layer to minimize the input images size. The operation of maximum pooling and average pooling is shown in Fig. 5.

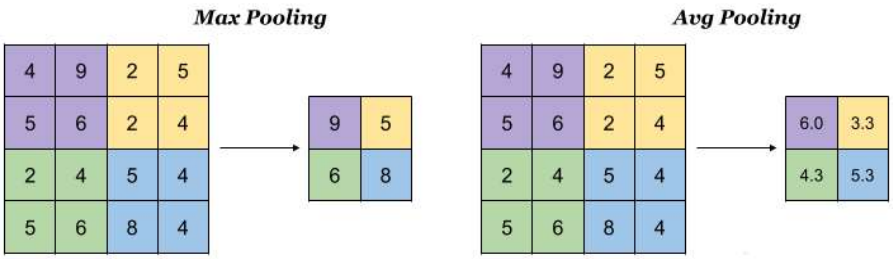


Fig.5. Operation of Max pooling & Average pooling

e. **Fully connected layer -** Fully connected layers are used after the pooling layer. When the fattened matrix goes through a fully connected layer, it classifies the images and produces the predictions. It may use the attributes learned by the preceding layers to classify an image as containing a dog, cat, bird, human, face, or other objects.

f. **Loss layer -** During training, the loss layer compares the fully-connected layer's actual values in order to minimise the difference between the guess and the true value. The weights in the convolution and fully-connected layers are adjusted by the loss layer.

g. **Softmax function**

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$ .....Equation. 2.

The Softmax function, that determines the probability of each class and gives the output between 0 and 1. It is applied just before the output layer. The softmax function formula is shown in Eq. 2.

h. **Output layer -**That is the final layer in the convolutional neural networks, that produces the desired final prediction.

2.2.3. Feature Descriptor

i. **HOG -**The histogram of oriented gradients (HOG) [3] is based on first-order picture gradients that are densely clustered into overlapping orientations.
HOG descriptor focuses on the structure or the shape of an object and it visualises the main image. The HOG based feature is shown in Fig.6.

Fig.6 Simple image and after HOG visualization

ii. LBP - Local Binary Patterns (LBPs) [4] convert a grayscale image at the pixel level to an integer number matrix. This label matrix explains the original image. It computes the texture's local representation. The LBP feature descriptor is a powerful texture classification feature. Fig. 7 depicts the LBP operations.

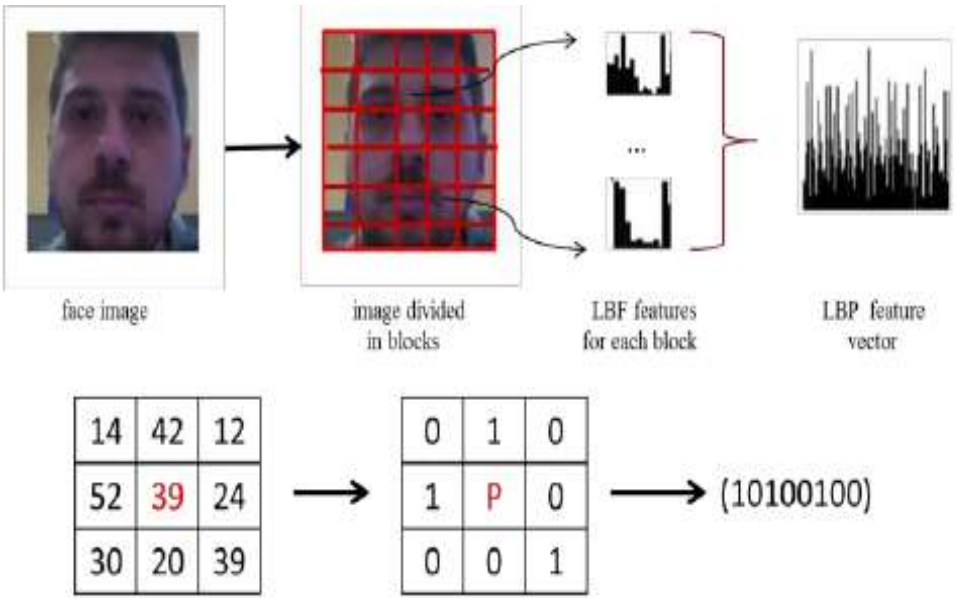


Fig. 7. LBP Operation

iii. SVM - SVM [5] is a popular and successful supervised learning classifier that may be used for classification and regression problems. The Support Vector Machine has been used successfully in pattern recognition applications such as face recognition and word recognition. Fig. 8 depicts the SVM process.

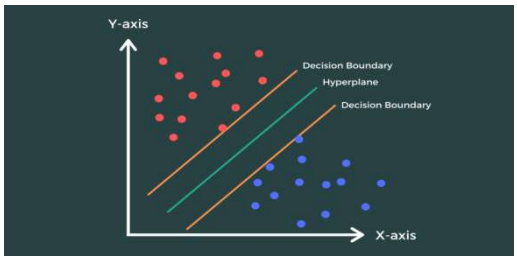


Fig.8. SVM Process

iv. **LDA -** Linear Discriminant Analysis (LDA) is a dimensionality reduction approach often used in machine learning to address issues with more than two classes. LDA considers the labelling of two distance points. The goal is to reduce the distance between data points in different categories. It is more important after projection, and the distance between data points of the same class is more compact. Before the LDA process and after the LDA process are shown in Fig.9.

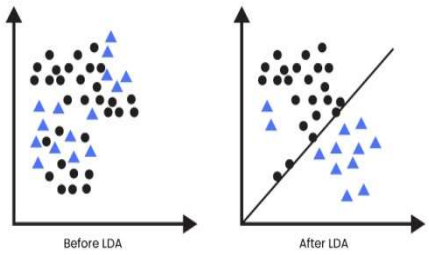


Fig.9. Before LDA and After LDA process

2.2.4. Parameter

The model parameters represent a model in ML or DL. In machine learning and deep learning, parameters are the values of learning algorithms. The learning algorithms can change independently.

KERNEL SIZE - Deep neural networks, or convolutional neural networks (CNN), are essentially a stack of layers defined by the actions of a number of filters on the input. Kernels are the common name for these filters. Convolutional filters are the kernels in the convolutional layer. In reality, no convolution is performed, but rather a cross-correlation. The kernel size refers to the width and height of the filter mask.

PADDING - Padding is a term used in CNN to describe the number of pixels added to an image before the CNN kernel processes it. When the padding in a CNN is set to zero, the value of each pixel added is zero.

STRIDE - Stride is a component of convolutional neural networks optimised for image and video data compression. Stride is a neural network filter parameter that affects the amount of movement over an image or video. When the stride of a neural network is set to one, the filter moves one pixel, or unit, at a time.

FILTER - Filters are in charge of extracting a certain feature from a raw image. CNNs attempt to learn such filters; that is, the filters parametrized in CNNs are learned during CNN training. Each filter in a Conv2D is applied to each input channel, and the results are combined to produce output channels. As a result, the number of filters and output channels are both the same.

3. Literature Survey

Wang et al. [6] proposed an algorithm that uses an integrated expression algorithm to detect learners' faces from images. This is done by detecting the emotional state of learners. The exiting algorithm has four models. Part combination model, Eyes model, Mouth model and Eyebrows model. Through those four models, they got 76.9% prediction accuracy.

Shen et al. [7] proposed an affective e-learning model based on facial emotion. In their model, the learner emotions were detected and fed into the affective e-learning model in real-time so that proper learning content could be delivered depending on the learners' current emotional states and other learning contexts. They achieved a 68.7% accuracy rate.

Zhou et al. [8] proposed a model for a distance education system that estimates the affective state of the learners in the modern distance education system. The support vector machine (SVM) algorithm was applied to recognize facial expressions in real-time. They claimed that the proposed system could achieve a high recognition rate.

Brunelli et al. [9] designed a model based on facial element extraction and it has three classifications, specifically **feature-based** [9], **appearance-based** [10-11], and **template-based** [12].
The appearance-based method converts face images into a small collection of typical feature images known as "**eigenfaces**," which are key components of the first training set of face images [13]. Eigenfaces are nothing more than a collection of orthonormal basis matrices. All of these eigenvectors may be represented as a ghostly face, also known as a "**eigenface**" [14]. They trained their model up to 100 epochs and got highest 76.62% accuracy.

Wprke et al. [15] designed a FER system employing SVMs and local region-specific features. In their FER system, they have converted the whole input face image into domain-specific local regions. The local region's LBP and neutrosophic c-means (NCM) features are retrieved and fed into the SVM classifier. CK+ dataset is used in their proposed work.

Elagoz et al. [16] proposed a method for deleting unnecessary pixels by integrating three datasets (KDEF, JAFFE, and their custom dataset) and using the Haar cascade function. This CNN model achieved an accuracy of 91.81% for the classification of seven different emotions. This model was

more accurate in predicting angry, fear, surprise and happy emotional states, but less accurate in predicting sad.

Chan et al. [17] designed a new deep neural network system for emotion recognition with the help of two convolutional layers followed by max-pooling and four inception layers. They used seven different types of datasets that achieved better accuracy. In their system the got 68.2% accurate prediction rate.

Deep learning for emotion recognition was proposed by J. Singh et al. [18]. They used a facial landmark-based technique with the free source Dlib package. A landmark model was used in this deep metric learning technique to recognise faces. Facial landmarks are the points that are utilised to distinguish the difference between faces. A human face has 68 landmarks, according to the Dlib library. These primary landmarks are eye contour points, eyebrow contour points, pupils, nostrils, nose tips, mouth contour points, etc. Their suggested CNN-based facial landmark-based technique has three types of layers: convolutional, max-polling, and fully-connected.

Pramerdorfer et al. [19] designed a CNN model where modern deep CNNs were used and exclusively tested on the FER2013 dataset. They achieved an accuracy of 75.2% on the FER2013 dataset. They used current architectures to increase the performance of face emotion recognition and were able to outperform the shallow and basic CNN architectures.

Zhan et al. [20] proposed a facial expression classification method. In this method, face key points and land marks are detected. Distance classification and LBP algorithms are used to extract geometric and textural features from key point locations. Finally, they classified the face expression using SVM classification. The CK+ dataset was used in this work and achieved 60% accuracy.

Based on it, introduce a new tutoring system architecture where affective computing is applied. Face emotion recognition is used to construct an emotion recognition system that detects the learner's emotional state and, depending on it, automatically adjusts the level of content that is appropriate for the learner.

4. Proposed Methodology

In this proposed work, an adaptive tutoring system is designed so that the content materials in the tutoring system can be changed as per the learner's emotions.

To capture the learner's facial emotions, a Facial Emotion Recognition (FER) model is implemented, where different types of emotions are identified. Mostly disgust, angry, fear, happy, neutral, sad, and surprise.

If the detected emotion is sad, the learning materials are changed so that students better comprehend. If the predicted emotion is happy, then more enhanced same-version learning materials are provided.

In this proposed adaptive tutoring system, the images of the learner's facial expression are captured during the learning activity. The captured images are sent to the Facial Emotion Recognition (FER) Model to predict the emotions of the learners. Based on the facial expression, the learning content is changed. Different types of features extracted have been used to identify which method is given better accuracy in detecting learner emotions.

For the developing the facial emotion model CNN architecture is used and the forming feature descriptor is used. Here HOG, LBP, LDA feature descriptor are used.

For designing the tutoring system, HTML, CSS, and Javascript are used, and, for the database segment, PhpMyAdmin has been used.

In the proposed adapting tutoring system, the following components are used in the Facial Emotion Recognition (FER) Model:

i. Image prepossessing
ii. Feature extraction
iii. Image classification

Overall FER block diagram are shown in Fig. 10.

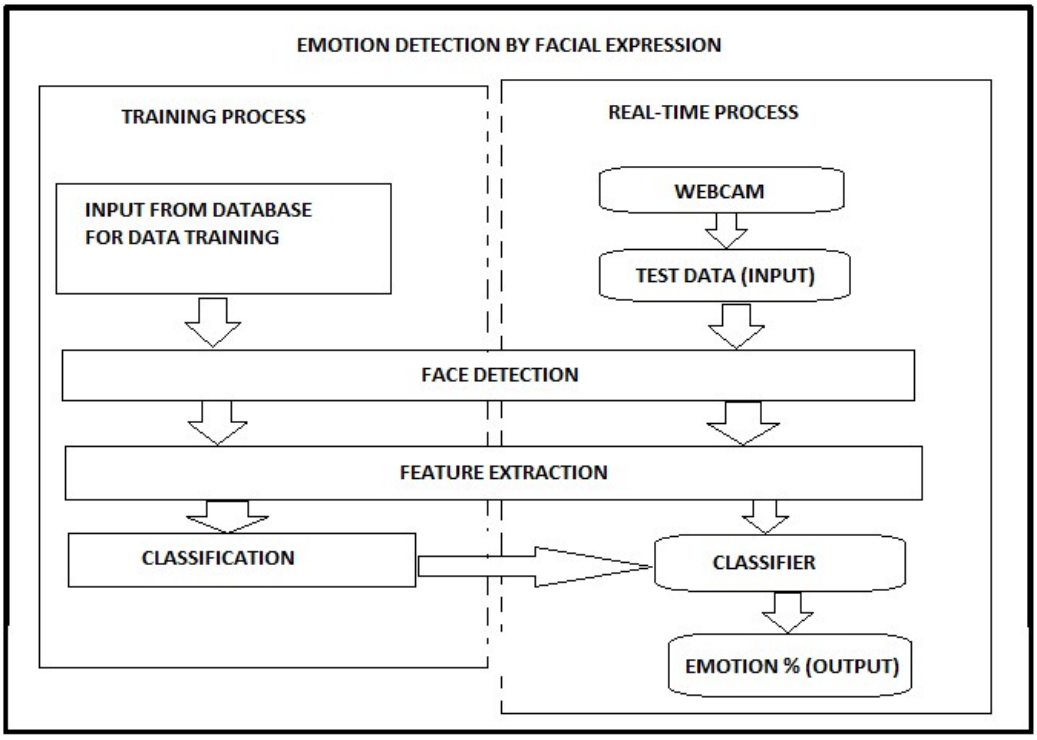


Fig.10. Overall Block diagram of the FER system.

i. Image prepossessing

The images are captured through the webcam, and the pixels are removed from the image for the cropping operation. The unwanted parts are the hair and neck, and the essential parts are the lips and eyes. Image preprocessing is a critical task since it decreases misclassification and increases the recognition rate.

After the images are prepared, they are fed into the Model for extract.

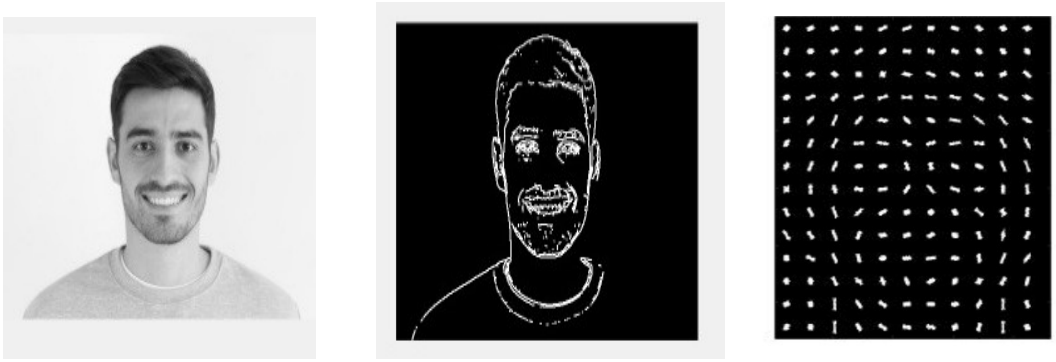
ii. Features Extraction

With the help of feature extraction, it can reduce the amount of redundant data in a data set. Finally, data reduction allows the model to be built with less machine effort while also increasing the speed of the learning and generalization steps in the machine learning process.

Three different types of feature descriptors are used.

a. Histogram of Oriented Gradients (HOG)

HOG feature descriptor is used that divides the image into little cells and computes the edge directions as given in Fig.11.



(a) (b) (c)

Fig. 11 Simple face image(a), Gradient Magnitude image(b), After used HOG application(c)

Here is the simple image, divided into cells of size 64x128 pixels.

Histograms are ordered into a unique HOG histogram, (Fig. 11 c) that is the final outcome of this algorithmic step.

b. Local Binary Pattern (LBP)

LBP converts a grayscale image at the pixel level to an integer number matrix. This label matrix explains the original image. The LBP feature extraction process depicts in Fig. 12

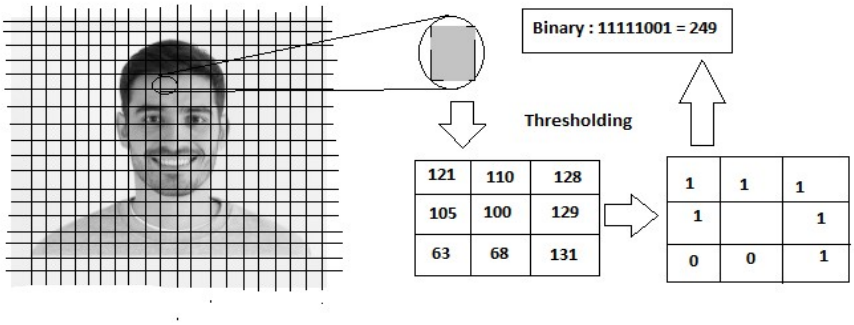


Fig.12. Feature extraction using LBP

There are differences between neighbour and centre pixels here. A binary code is generated for each pixel value in the image by thresholding its neighbourhood with the value of the centre pixel. A binary pattern may be made out of this binary code.
The centre pixel value acts as a threshold for the 3x3 neighbour pixels.
Threshold creates a binary pattern that represents a textural characteristic.

Here the threshold value is 100 for the centre pixel. If the values of the neighbouring pixels are less than the threshold, they are assigned to 0. If the neighbour pixels are more or equal to the threshold, it becomes 1. The LBP value is calculated via scalar multiplication of the binary and weight matrices.
Finally, getting the LBP value represented by the total multiplication of all the results.

c. Linear Discriminant Analysis (LDA)

LDA employs projections of training images onto subspace, a subspace defined by the fisher faces. Recognition is accomplished by projecting a new face onto the fisher space, after which the KNN algorithm is used for identification. Block diagram of LDA model is shown in Fig.13. LDA takes for labelling the two distance point. The process of LDA is shown in Fig.14, and Distance point is depicts in Fig 15.

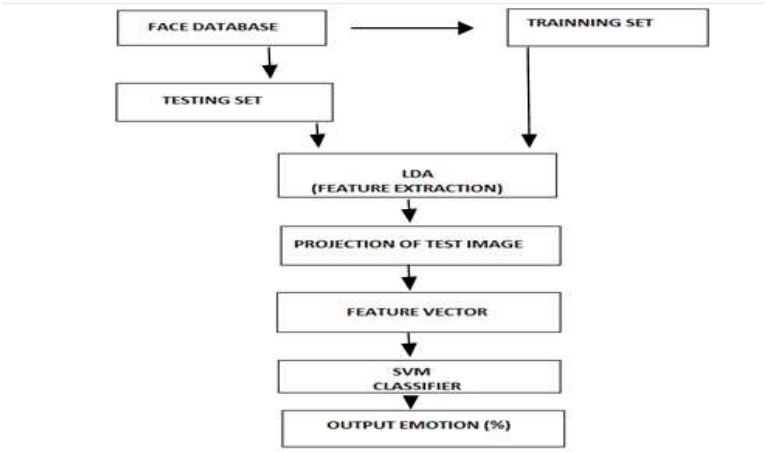


Fig. 13 Block diagram of LDA model

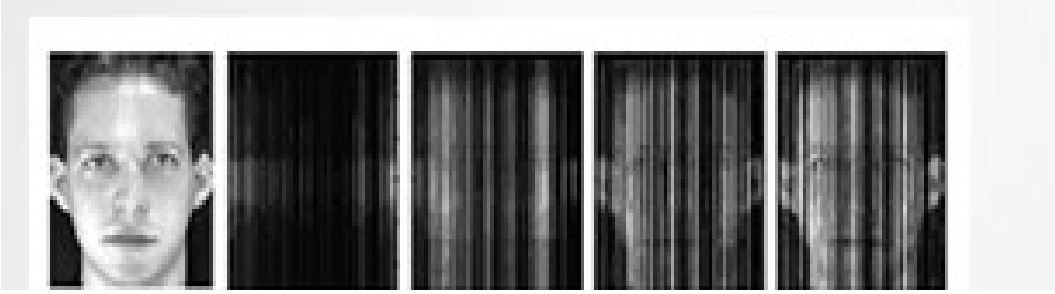


Fig. 14 The LDA process

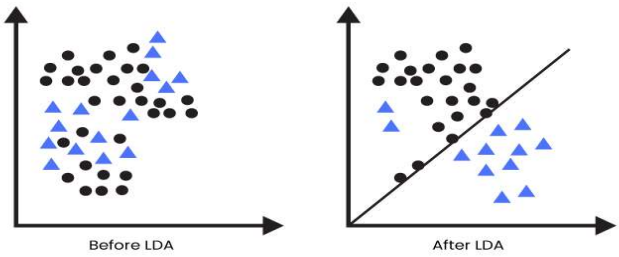


Fig.15.Before LDA process and After LDA process

iii. Image Classification

SVM is used for classification and regression for different states of learning emotions.

So, in this proposed work, firstly recognise the face key point and face landmarks, then extract it by using different feature extraction methods and classified by SVM classifier.

5. Experimentation and Result

DATASET

The learners' images are capture through webcam with various expressions which has send to the Facial Emotion Recognition (FER) system for detection the emotional expression. In proposed method CK+ datasets [21] is use for training and 50 images for testing to accurately predict the emotion.

CK+ Database

The CK+ database is a publicly available dataset for recognizing action units and emotions. It comprises expressions that are both posed and unposed. Images from the CK+ dataset are stored in distinct files labeled according to their emotional degree, and all images have been converted to gray scale. Here we used total 1750 images for training.

The average accuracy is determined to summarise overall accuracy performance.
Eq. 3. shows the formula for accurate calculation.

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of all cases}} * 100\% \quad \ldots\ldots\ldots \text{ Equation. 3}$$

The overall accuracy result :

Method	Accuracy
HOG + SVM	86.3%
HOG + Face Landmarks	56.4%
HOG + CNN	97.6%
HOG + LBP	87.4%
LDA + KNN	96.4%

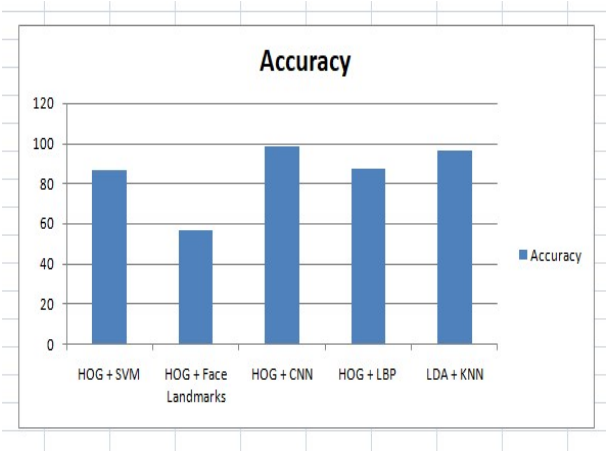


Fig.16. The accuracy result using different methods

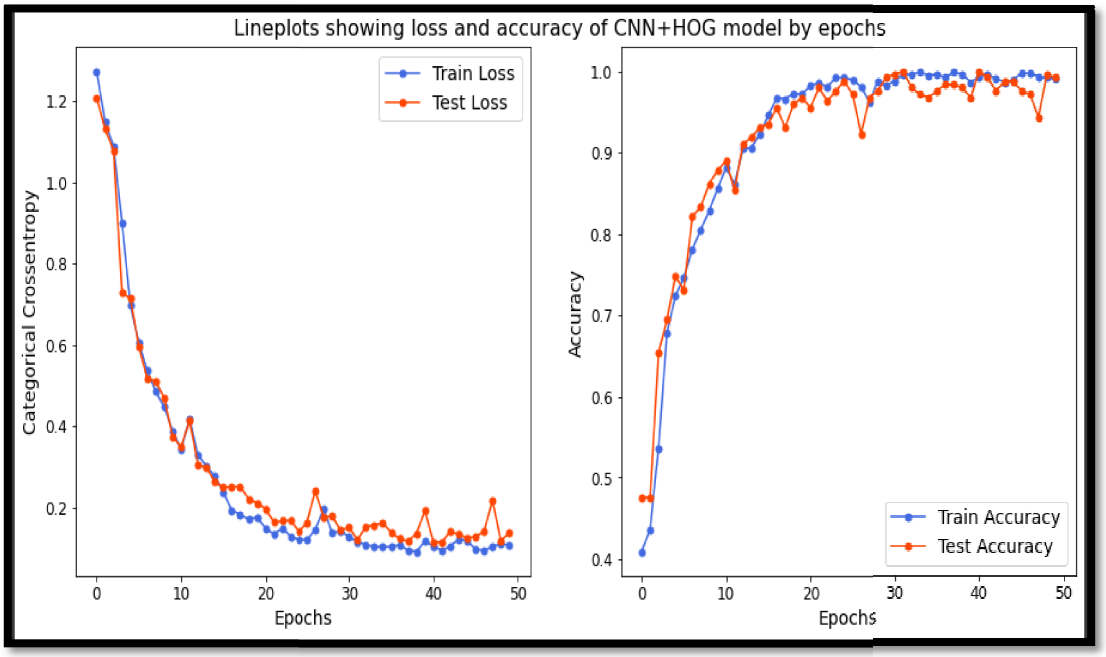


Fig. 17 (The training accuracy and test accuracy graphs)

The outcome of prediction is shown in Fig.18.

Fig. 18. (The Prediction Result- Happy)

6. Conclusion

This work is based on detecting various facial expressions using different feature extraction methods. Here we analysed the facial expressions of seven different emotions in various learners.

The captured images are fed in to CNN and different feature extraction methods are applied, followed by an SVM classifier for classifying the emotions. After applying different feature extraction methods, the CNN model using HOG descriptor gives better accuracy than LBP, LDA, and other methods. The 97.6% accuracy was achieved in the CNN model using HOG.

7. Future scope

The adaptive tutoring system is designed in such a way that the learning materials can be modified and adapted as per the learner's emotions. This learner's emotions are detected using a Face Emotion Recognition (FER) model, and several feature extractor methods are used to compare the detection accuracy.

In the future, this model can be extended by adding more emotion–sensing methods like speech recognition, text analysis, voice analysis, etc. to confirm the emotion sensed through the different methods.

8. Reference

[1] H. S. Nwana, "Intelligent tutoring systems: an overview," Artificial Intelligence Review, vol. 4, no. 4, pp. 251–277, 1990.

[2] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: a review," Journal of medical systems, vol. 42, no. 11,
pp. 1–13, 2018.

[3] A. Nandi, P. Dutta, and M. Nasir, "Automatic facial expression recognition using histogram oriented gradients (hog) of shape information matrix," in International Conference on Intelligent Computing and Communication. pp. 343–351, 2019.

[4] L. Yuan, C.-m. Wu, and Y. Zhang, "Facial expression feature extraction using hybrid pca and lbp," The Journal of China Universities of Posts and Telecommunications, vol. 20, no. 2, pp. 120–124, 2013.

[5] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in Proceedings of the 2015 ACM on international conference on multimodal interaction,
pp. 435–442, 2015.

[6] C.-H. Wu, Y.-M. Huang, and J.-P. Hwang, "Review of affective computing in education/learning: Trends and challenges," British Journal of Educational Technology, vol. 47, no. 6, pp. 1304–1323, 2016.

[7] L. Shen, M. Wang, and R. Shen, "Affective e-learning: Using "emotional" data to improve learning in pervasive learning environment," Journal of Educational Technology & Society, vol. 12, no. 2, pp. 176–189, 2009.

[8] J.-M. Sun, X.-S. Pei, and S.-S. Zhou, "Facial emotion recognition in modern distant education system using svm," in 2008 International Conference on Machine Learning and Cybernetics, vol. 6. pp. 3545–3548, 2008.

[9] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," IEEE transactions on pattern analysis and machine intelligence, vol. 15, no. 10, pp. 1042–1052, 1993.

[10] D. Shah and S. Marshall, "Statistical coding method for facial features," IEE Proceedings-Vision, Image and Signal Processing, vol. 145, no. 3, pp. 187–192, 1998.

[11] I. King and L. Xu, "Localized principal component analysis learning for face feature extraction and recognition," in Proceedings to the Workshop on 3D Computer Vision. Shatin Hong Kong, pp. 124–128, 1997.

[12] R. J. Baron, "Mechanisms of human facial recognition," International Journal of Man-Machine Studies, vol. 15, no. 2, pp. 137–178, 1981.

[13] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of cognitive neuroscience, vol. 3, no. 1, pp. 71–86, 1991.

[14] H. Wu, Q. Chen, and M. Yachida, "Facial feature extraction and face verification," in Proceedings of 13th International Conference on Pattern Recognition, vol. 3. pp. 484–488, 1996.

[15] D. Ghimire, S. Jeong, J. Lee, and S. H. Park, "Facial expression recognition based on local region specific features and support vector machines," Multimedia Tools and Applications, vol. 76, no. 6, pp. 7803–7821, 2017.

[16] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh, and A. Akan, "Real time emotion recognition from facial expressions using cnn architecture," in 2019 medical technologies congress (tiptekno). pp. 1–4, 2019.

[17] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in 2016 IEEE Winter conference on applications of computer vision (WACV). pp.1–10, 2016.

[18] R. Gill and J. Singh, "A deep learning approach for real time facial emotion recognition," in 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART). pp. 497–501, 2021.

[19] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: state of the art," arXiv preprint arXiv:1612.02903, 2016.

[20] Q. Li, S. Zhan, L. Xu, and C. Wu, "Facial micro-expression recognition based on the fusion of deep learning and enhanced optical flow," Multimedia Tools and Applications, vol. 78, no. 20, pp. 29 307–29 322, 2019.

[21] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in 2010 ieee computer society conference on computer vision and pattern recognition-workshops. pp. 94–101, 2010.

9. Appendix

The overview of ITS as in Fig [19, 20, 21]

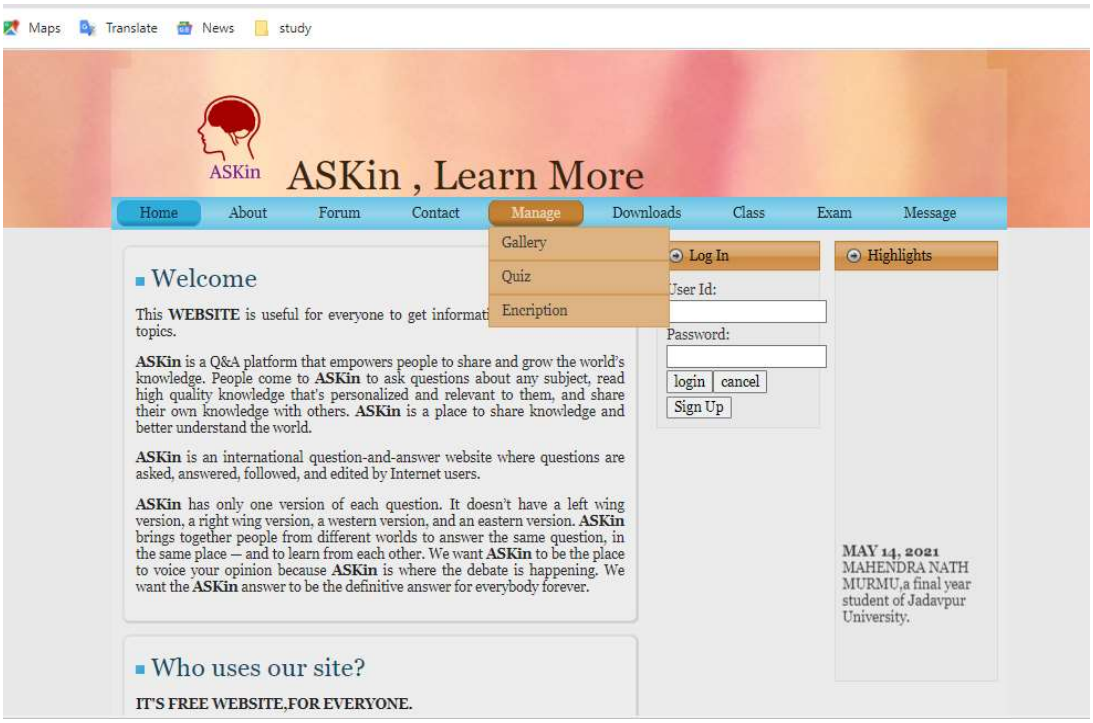


Fig.19. Overview of ITS

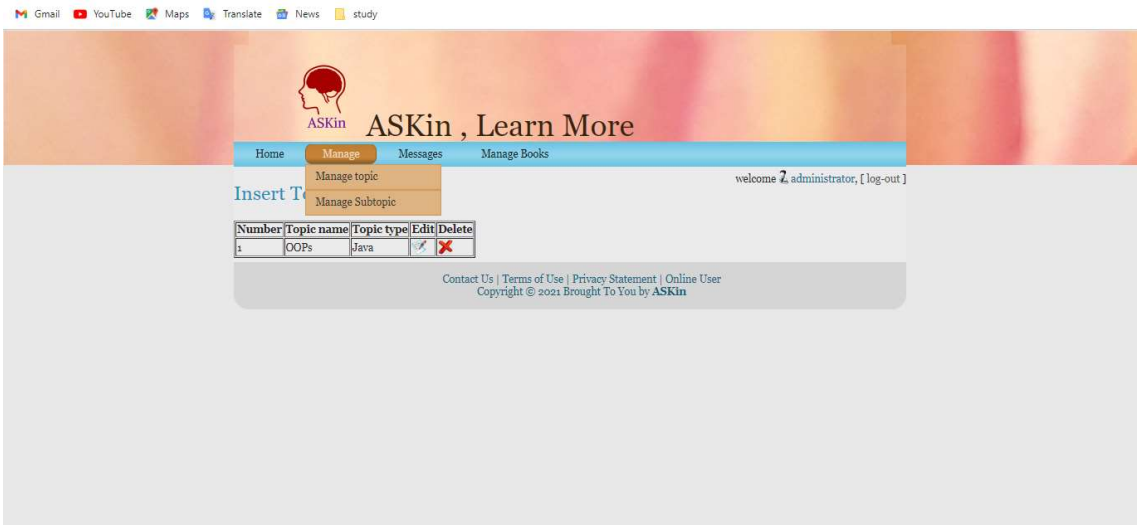


Fig.20. Overview of ITS

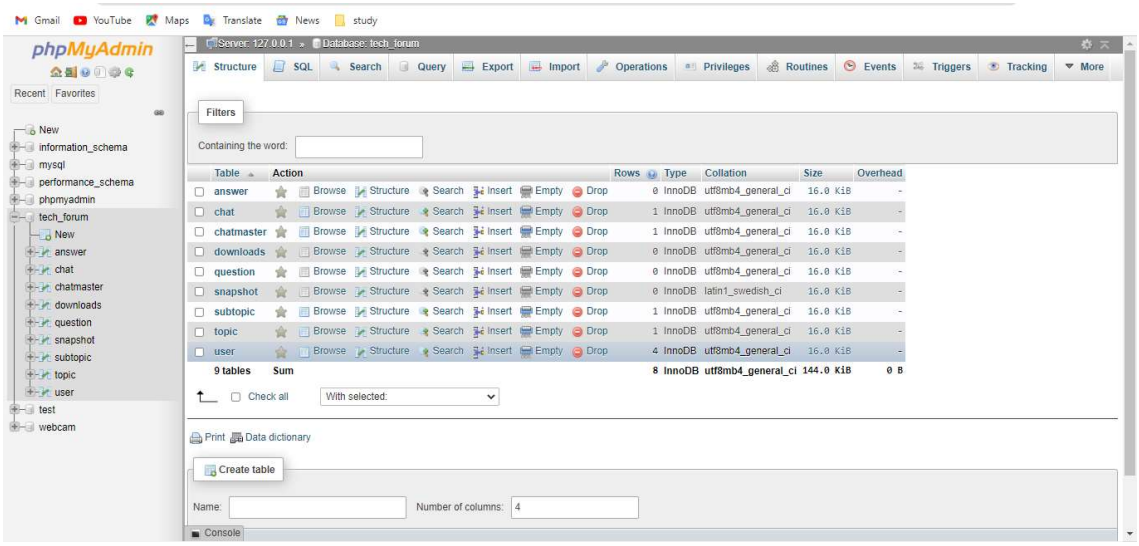


Fig.21. Database

Uhome.php

```
<?php
     session_start();
     require("header.php");
     require("checkUser.php");
?>
<script type="text/javascript">
     document.getElementById("auhome").className="active";
</script>

<h4><a href="que.php">My Questions</a>
     
<a href="ans1.php">My Answers </a></h4>
<?php

          //----------------- Storing values for SELECT
statement -------------------
          $questionID = "";
          $heading = "";
          $uimg = "";
          $fullname = "";
          $question_detail = "";
          $datetime = "";


          $host = "localhost";
         $dbUsername = "root";
```

```
        $dbPassword = "";
        $dbname = "tech_forum";

        //------------------------------ creating
connection ----------------------
        $conn = mysqli_connect($host, $dbUsername,
$dbPassword, $dbname);

              //------------------------------- STATE
Statement -------------------------------
                   $SELECT = "SELECT question_id,
heading, uimg, fullname, question_detail, datetime FROM
question,user WHERE question.user_id=user.user_id ORDER BY
datetime desc";

                $stmt = mysqli_prepare($conn, $SELECT);
                if($stmt == false) {//    <-------------
If something wrong in statement ---------------
                   echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
                if (mysqli_stmt_execute($stmt)) {//    <--
---------------------------- If statement executed -
                   mysqli_stmt_store_result($stmt);//
<---------------------------- Storing the result in $stmt

                   //--------------------------- Binding
the executed values in the variables -------------------
                   mysqli_stmt_bind_result($stmt,
$question_id, $heading, $uimg, $fullname,
$question_detail, $datetime);

                   //------------------------------ For
all the $stmt values ------------------------------------
--
                   while(mysqli_stmt_fetch($stmt))
                        {

                             echo "<span class='box2'>";
                             echo "<span class='head'><a
href='questionview.php? qid=" . $question_id . "'
>$heading</a></span>";

                             echo "<table>";
                             echo "<tr><td valign='top'
width='100px'>
                                  <img src='" . $uimg .
"' alt='' class='uimg'/>
                                  <br/>
```

```
                                    " . $fullname ."
                                    <td valign='top'>
                                    " . $question_detail .
"<br/><br/>
                                    " . $datetime . "<br/><br/>
                                    </td></tr>";


                                    echo "</table></span><div
class='h10'></div>";
                            }

                            //------------------ Closing the
connection ----------------------
                        mysqli_stmt_close($stmt);
                        mysqli_close($conn);

                    } else {//    <---------------------------
---- If STATE statement not executed -------------
                        echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                    }
?>

<?php require("footer.php");?>
```

utility.php

```
<?php
error_reporting(1);
     function ExecuteQuery ($SQL)
     {
          $con=mysqli_connect ("localhost", "root","");
          mysqli_select_db ($con,"tech_forum");

          $rows = mysqli_query ($con,$SQL);

          mysqli_close ($con);

          return $rows;
     }

     function ExecuteNonQuery ($SQL)
     {
          $con=mysqli_connect ("localhost", "root","");
          mysqli_select_db ($con,"tech_forum");
```

```
          $result = mysqli_query ($con,$SQL);

          mysqli_close ($con);

          return $result;
     }
?>
```

checkUser.php

```
<?php
     if(!isset($_SESSION["fn"]))
          header("location:index.php");
?>

<span style="text-align:right ;width:90%; display:block;
margin-bottom:5px;">
     welcome <a href="uedit.php"><img
src="res/images/1.jpg" class="imagedel"/><?php echo
$_SESSION["fn"];
     ?></a>, [ <a href="logout.php">log-out</a> ]
</span>
```

forum.php

```
<?php session_start();
 require("header.php");

 if ($_SESSION["fn"] == null){
     header("location:unreg.php");
     exit();
 }

 require("checkUser.php");
 ?>

 <script type="text/javascript">
     document.getElementById("aforum").className="active";
</script>

<?php
     $topic = ExecuteQuery ("SELECT * FROM topic");

     while ($r1 = mysqli_fetch_array($topic))
     {
```

```
                echo "<div
class='heading'>$r1[topic_name]</div>";

                $stopic = ExecuteQuery ("SELECT * FROM
subtopic WHERE topic_id=$r1[topic_id]");

                while ($r2 = mysqli_fetch_array ($stopic) )
                {
                     echo "<div class='box'>";
                     echo "<div class='sub-heading'>
                                <a
href='questions.php?id=$r2[subtopic_id]'>
$r2[subtopic_name]</a>

                           </div>";
                     echo
"<p>$r2[subtopic_description]</p>";
                     echo "</div>";
                }
     }


?>

<?php require("footer.php"); ?>
```

Questionview.php

```
<?php session_start();
require("header.php");
require("checkUser.php");
?>


<?php
          //----------------- Variables for UPDATE and
SELECT statements -------------------
          $heading = '';
          $uimg = '';
          $fullname = '';
          $datetime = '';
          $question_detail = '';

          //----------------- Extra Variables for Second
SELECT statement -------------------
          $answer_id = '';
          $likes = '';
          $answer_detail = '';
```

```
          $host = "localhost";
         $dbUsername = "root";
         $dbPassword = "";
         $dbname = "tech_forum";



//_____________________________________________________________
_____________________________________________
//============================================ Updating
the views ============================================
//_____________________________________________________________
_____________________________________________________

//----------------------------------- creating connection
for UPDATE -----------------------------
        $conn = mysqli_connect($host, $dbUsername,
$dbPassword, $dbname);

              //------------------------------- UPDATE
Statement -------------------------------
                  $UPDATE = "UPDATE question SET
views=views+1 WHERE question_id=?";

                $stmt = mysqli_prepare($conn, $UPDATE);
                if($stmt == false) {//    <-------------
If something wrong in statement ---------------
                  echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
                mysqli_stmt_bind_param($stmt, "i",
$_GET[qid]);//      <------------------------ Binding
UPDATE with values
                if (mysqli_stmt_execute($stmt)) {//    <--
---------------------------- If statement executed -

                        //------------------ Closing the
connection ---------------------
                  mysqli_stmt_close($stmt);
                  mysqli_close($conn);

                } else {//    <--------------------------
---- If UPDATE statement not executed -------------
                  echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
```

```
                }


//_____________________________________________________________
_____________________________________________
//================================ Accessing the data the
from question and userr tables ====================
//_____________________________________________________________
_____________________________________________________

//----------------------------------- creating connection
for SELECT ----------------------------
        $conn = mysqli_connect($host, $dbUsername,
$dbPassword, $dbname);

               //-------------------------------- SELECT
Statement ------------------------------
                    $SELECT = "SELECT question_id,
heading, uimg, fullname, question_detail, datetime FROM
question, user WHERE  user.user_id=question.user_id AND
question_id=?";

                $stmt = mysqli_prepare($conn, $SELECT);
                if($stmt == false) {//    <-------------
If something wrong in statement ---------------
                    echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
                mysqli_stmt_bind_param($stmt, "i",
$_GET[qid]);//      <------------------------ Binding
SELECT with values
                if (mysqli_stmt_execute($stmt)) {//    <--
----------------------------- If statement executed -
                    mysqli_stmt_store_result($stmt);//
<---------------------------- Storing the result in $stmt

                    //--------------------------- Binding
the executed values in the variables ------------------
                    mysqli_stmt_bind_result($stmt,
$question_id, $heading, $uimg, $fullname,
$question_detail, $datetime);

                    //------------------------------ For
all the binded $stmt values ------------------------------
---------
                    while(mysqli_stmt_fetch($stmt))
                         {

                              $head = $heading;
```

```
                                echo "<h4>";
                                echo $head;
                                echo "</h4>";
                                echo "<span class='box2'>";
                                echo "<span class='head'><a
href='answer.php?id=$_GET[qid]'>REPLY</a></span>";


                                echo "<table>";
                                echo "<tr><td valign='top'
width='100px'>
                                     <img src='$uimg' alt=''
class='uimg'/>
                                     <br/>
                                $fullname
                                <td valign='top'>
                                <b>$head</b><br/>
                                $datetime<br/><br/>
                                $question_detail</tr>";


                                echo "</table></span><div
class='h10'></div>";
                           }
                           //------------------ Closing the
connection ----------------------
                     mysqli_stmt_close($stmt);
                     mysqli_close($conn);

                } else {//    <----------------------------
---- If SELECT statement not executed -------------
                     echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }




//_____________________________________________________
_____________________
//================================ Accessing the data the
from answer and userr tables ====================
//_____________________________________________________
_____________________________________________________

//---------------------------------- creating connection
for SELECT ----------------------------
```

```
        $conn = mysqli_connect($host, $dbUsername,
$dbPassword, $dbname);

              //------------------------------- SELECT
Statement -------------------------------
                  $SELECT = "SELECT answer_id, likes,
uimg, fullname, answer_detail, datetime FROM answer,user
WHERE question_id=? AND answer.user_id=user.user_id ORDER
BY  datetime DESC";

                $stmt = mysqli_prepare($conn, $SELECT);
                if($stmt == false) {//    <-------------
If something wrong in statement ---------------
                  echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
                mysqli_stmt_bind_param($stmt, "i",
$_GET[qid]);//      <------------------------ Binding
SELECT with values
                if (mysqli_stmt_execute($stmt)) {//    <--
---------------------------- If statement executed -
                  mysqli_stmt_store_result($stmt);//
<--------------------------- Storing the result in $stmt

                  //--------------------------- Binding
the executed values in the variables -------------------
                  mysqli_stmt_bind_result($stmt,
$answer_id, $likes, $uimg, $fullname, $answer_detail,
$datetime);

                  //------------------------------ For
all the binded $stmt values ------------------------------
---------
                  while(mysqli_stmt_fetch($stmt))
                       {

                            echo "<span class='box2'>";
                            echo "<span class='head'><a
href='answer.php?id=$_GET[qid]'>REPLY</a><a
href='like.php?id=$answer_id&qid=$_GET[qid] '
class='view2' >Like $likes</a>
                                 </span>";

                            echo "<table>";
                                       echo "<tr><td
valign='top' width='100px'>
                            <img src='$uimg' alt=''
class='uimg'/>
                                 <br/>
```

```
                                $fullname<td
valign='top'><b>Re :
$head</b><br/>$datetime<br/><br/>$answer_detail</tr>";

                                echo "</table></span><div
class='h10'></div>";
                            }
                            //------------------ Closing the
connection ----------------------
                    mysqli_stmt_close($stmt);
                    mysqli_close($conn);

                } else {//    <----------------------------
---- If SELECT statement not executed -------------
                    echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }

?>

<?php
require("footer.php")?>
```

LoginH.php

```
<?php require("header.php");
?>
<?php

//--------------- Getting values from logon.php ----------
---
$uid=$_POST["uid"];
$pwd=$_POST["pwd"];
$active = 1;//     <-------------------------- if user is
logged in, then active value (isuser in SQL database) is
true, false otherwise
     if ( isset($uid) && isset($pwd)) {
          $host = "localhost";
         $dbUsername = "root";
         $dbPassword = "";
         $dbname = "tech_forum";

         //---------------------------- create connection
for login - collecting user_id, fullname, user_type ------
------
```

```
        $conn = mysqli_connect($host, $dbUsername,
$dbPassword, $dbname);
                    $SELECT = "SELECT user_id, fullname,
user_type FROM user WHERE username=? AND password=?";
                //                      ^-----------
SELECT STATEMENT -----<

                $stmt = mysqli_prepare($conn, $SELECT);

                if($stmt == false) { //    <----- If
something wrong in statement
                    echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
                mysqli_stmt_bind_param($stmt, "ss", $uid,
$pwd);//      <----------------------- Binding SELECT
with values
                if (mysqli_stmt_execute($stmt)) {//    <--
---------------------------- If statement executed -
                    mysqli_stmt_store_result($stmt);//
<---------------------------------- Storing results in
variable $stmt
                    //echo "<h10
style='color:MediumSeaGreen;''>Statement executed</h10>";
                    //echo "num rows = " .
mysqli_stmt_num_rows($stmt);
                    if (mysqli_stmt_num_rows($stmt) == 1)
{//   <------------ If number of rows = 1, means one user
is selected
                        //echo "one row only";
                        mysqli_stmt_bind_result($stmt,
$uid, $fn, $utype);//    <---------- Binding the values
with the result
                        mysqli_stmt_fetch($stmt);//   <--
---------------------------------- Storing values in the
variables
                        session_start();//   <-----------
-------------------------- Season for login created

                       //------------ Assigning the
session variables ----------
                            $_SESSION["uid"]= $uid;
                            $_SESSION["fn"] = $fn;

                       //------------ Closing the
connection for SELECT statement
                        mysqli_stmt_close($stmt);
                        mysqli_close($conn);
```

```
                        //---- creating new connection ---
--------------------------------
                        $conn = mysqli_connect($host,
$dbUsername, $dbPassword, $dbname);
                        $update = "UPDATE user SET
isuser=? WHERE user_id=?";//   <---------------- UPDATE
statement ---
                        $stmt = mysqli_prepare($conn,
$update);

                        if($stmt == false) {//   <--------
-------- If something wrong in statement
                            echo "dying.........";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                        }
                        mysqli_stmt_bind_param($stmt,
"is", $active, $_SESSION["uid"]); //   <---------- Binding
UPDATE with values
                        if (mysqli_stmt_execute($stmt))
{//    <------------------------------ If statement
executed -
                            //------------ Closing the
connection for UPDATE statement
                            mysqli_stmt_close($stmt);
                            mysqli_close($conn);
                            //echo "statement exwcuted";
                        } else {//    <-------------------
------------ If UPDATE statement not executed ------------
-
                            echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                        }


                             if($utype=="admin")//   <---
-------------- User is admin or not ------ options may be
(admin/ordinary)
                             {
                            ob_start();

     header("Location:admin/home.php");
                             }
                             else
                             {
                                  header("location:
uhome.php");
                             }
```

```
                    } else {
                        //echo
"fffffffffffffffffffffffff";
                                header("location:
index.php?act=invalid");
                    }


                    //echo "statement exwcuted";
                } else {//    <----------------------------
---- If SELECT statement not executed ----------------
                    //echo "<h10
style='color:Tomato;'>Something went wrong</h10>";

die("<pre>".mysqli_error($conn).PHP_EOL.$query."</pre>");
                }
    }
?>
<?php require("footer.php");?>
```

```
import numpy as np
import tensorflow as tf

from sklearn.model_selection import train_test_split

import os
import cv2
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt

from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten,BatchNormalization
from tensorflow.keras.layers import Dense, MaxPooling2D,Conv2D
from tensorflow.keras.layers import Input,Activation,Add
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint

import pandas as pd

from keras.models import Sequential,load_model,Model
from keras.layers import Conv2D,MaxPool2D,Dense,Dropout,BatchNormalization,Flatten,Input
from sklearn.model_selection import train_test_split
```

Import Dataset

```
[ ] dataset_folder='/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48'
    sub_folders=os.listdir(dataset_folder)

    sub_folders

    ['anger', 'contempt', 'disgust', 'happy', 'sadness', 'surprise', 'fear']
```

```
[ ] # Reading folder names as labels and images underneath
    i=0
    last=[]
    images=[]
    labels=[]
    temp = sub_folders

    # reading folders in the main dataset folder, one at a time
    for sub_folder in sub_folders:
      sub_folder_index = temp.index(sub_folder)
      label = sub_folder_index

      # Define labels basis use case. We are using positive:0, negative:1, neutral:2
      # for our use case of predicting emotions of visitors entering a retail store
```

```
    # Define labels basis use case. We are using positive:0, negative:1, neutral:2
    # for our use case of predicting emotions of visitors entering a retail store
    if  label in [4, 6]:     # label in ['happy', 'surprise']
      new_label=0            # changed to label = positive emotion
    elif label in [0,5]:       # label in ['anger','sadness']
      new_label=1            # changed to label = negative emotion
    else:                    # label in ["contempt', 'disgust', 'fear']
      new_label=2            # changed to label = neutral emotion


    path = dataset_folder+'/'+sub_folder
    sub_folder_images= os.listdir(path)

    # reading images in the sub folder, one at a time
    for image in sub_folder_images:
      image_path = path+'/'+image
      print(image_path+"\t"+str(new_label))

      image = cv2.imread(image_path)
      image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
      image= cv2.resize(image,(48,48))
      images.append(image)
      labels.append(new_label)
      i+=1
    last.append(i)
```

```
last.append(i)
```

```
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S010_004_00000017.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S011_004_00000020.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S022_005_00000032.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S010_004_00000018.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S010_004_00000019.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S011_004_00000021.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S011_004_00000019.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S014_003_00000029.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S014_003_00000028.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S022_005_00000031.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S026_003_00000013.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S022_005_00000030.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S014_003_00000030.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S028_001_00000024.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S028_001_00000022.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S028_001_00000023.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S026_003_00000015.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S026_003_00000014.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S032_003_00000016.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S029_001_00000017.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S034_003_00000027.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S034_003_00000026.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S037_003_00000020.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S034_003_00000025.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S032_003_00000017.png 1
/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/input/CK+48/anger/S029_001_00000019.png 1
```

Split into Train / Test

```
X_train, X_test, Y_train, Y_test= train_test_split(images_x, labels_y_encoded,test_size=0.25, random_state=10)
```

```
len(X_train)
```

```
735
```

```
len(X_test)
```

```
246
```

```
input = Input(shape = (48,48,1))

conv1 = Conv2D(32,(3, 3), padding = 'same', strides=(1, 1), kernel_regularizer=l2(0.001))(input)
conv1 = Dropout(0.1)(conv1)
conv1 = Activation('relu')(conv1)
pool1 = MaxPooling2D(pool_size = (2,2)) (conv1)

conv2 = Conv2D(64,(3, 3), padding = 'same', strides=(1, 1), kernel_regularizer=l2(0.001))(pool1)
conv2 = Dropout(0.1)(conv2)
conv2 = Activation('relu')(conv2)
```

Model Training

```
fle_s='/content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/output/emotion_model.h5'
checkpointer = ModelCheckpoint(fle_s, monitor='loss',verbose=1,save_best_only=True,
                               save_weights_only=False, mode='auto',save_freq='epoch')
callback_list=[checkpointer]
```

```
save = model.fit(X_train,Y_train,batch_size=32,validation_data=(X_test,Y_test),epochs=50,callbacks=[callback_list])
```

```
Epoch 1/50
23/23 [==============================] - ETA: 0s - loss: 1.2731 - accuracy: 0.4095
Epoch 1: loss improved from inf to 1.27315, saving model to /content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/output/emotion_model.h5
23/23 [==============================] - 13s 57ms/step - loss: 1.2731 - accuracy: 0.4095 - val_loss: 1.2085 - val_accuracy: 0.4756
Epoch 2/50
17/23 [=====================>........] - ETA: 0s - loss: 1.1723 - accuracy: 0.4265
Epoch 2: loss improved from 1.27315 to 1.14833, saving model to /content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/output/emotion_model.h5
23/23 [==============================] - 0s 13ms/step - loss: 1.1483 - accuracy: 0.4367 - val_loss: 1.1309 - val_accuracy: 0.4756
Epoch 3/50
16/23 [===================>..........] - ETA: 0s - loss: 1.1018 - accuracy: 0.4863
Epoch 3: loss improved from 1.14833 to 1.08874, saving model to /content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/output/emotion_model.h5
23/23 [==============================] - 0s 12ms/step - loss: 1.0887 - accuracy: 0.5374 - val_loss: 1.0785 - val_accuracy: 0.6545
Epoch 4/50
17/23 [=====================>........] - ETA: 0s - loss: 0.9331 - accuracy: 0.6765
Epoch 4: loss improved from 1.08874 to 0.90115, saving model to /content/drive/MyDrive/SM Maam/emotion/1.3_emotion_input_output/output/emotion_model.h5
```

Model Performance

```
# Checking the train and test loss and accuracy values from the neural network above.

train_loss = save.history['loss']
test_loss = save.history['val_loss']
train_accuracy = save.history['accuracy']
test_accuracy = save.history['val_accuracy']
```

```
# Plotting a line chart to visualize the loss and accuracy values by epochs.

fig, ax = plt.subplots(ncols=2, figsize=(15,7))

ax = ax.ravel()

ax[0].plot(train_loss, label='Train Loss', color='royalblue', marker='o', markersize=5)
ax[0].plot(test_loss, label='Test Loss', color = 'orangered', marker='o', markersize=5)

ax[0].set_xlabel('Epochs', fontsize=14)
ax[0].set_ylabel('Categorical Crossentropy', fontsize=14)

ax[0].legend(fontsize=14)
ax[0].tick_params(axis='both', labelsize=12)
```

----------X----------