"FACIAL EMOTION RECOGNITION USING CNN"

Submitted in Partial Fulfilment of the Academic

Requirement for the Award of Degree of

BACHELOR OF TECHNOLOGY

In

Department Of Computer Science

Submitted

By

I. Sumanth (19H51A05K4)

Under the esteemed guidance of

Ms. Suhasini

(Assistant Professor)

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Dist., Hyderabad.

2021-22

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad) Kandlakoya, Medchal Dist, Hyderabad.

CERTIFICATE

This is to certify that an Industry oriented Mini Project entitled with: **"FACIAL EMOTION RECOGNITION USING CNN "** is being

Submitted By

I. Sumanth (19H51A05K4)

In partial fulfilment of the requirement for the award of the degree of B. Tech in IT to the JNTUH, Hyderabad is a record of a bonafide work carried out under our guidance and supervision.

The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for the award of any other degree or diploma.

Signature of Guide	**Signature of HOD**
Ms.Suhasini	Dr.K.Vijaya Kumar
(Assistant Professor)	(HOD)

ACKNOWLEDGEMENT

We are extremely grateful to **Major Dr.VA.Narayana**, **Principal,** and **Dr.K.Vijaya Kumar**, **Head of the Department**, Dept of Computer Science, CMR College of Engineering & Technology for their inspiration and valuable guidance during the entire duration.

We are extremely thankful to internal guide **Ms.Suhasini** Dept of Computer science, CMR College Of Engineering & Technology for their constant guidance, encouragement, and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literature referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided us directly or indirectly for every step towards success.

I.Sumanth (19H51A05K4)

ABSTRACT

Emotions analysis play an important role in modern life era. The information about what a user is feeling as a reaction to a product, or to an advertisement, is a very important aspect. From response towards advertisements to making an enhanced and personalized user interface, requires a solution pertaining to the field of finding the emotional state of the user. This project aims at predicting the emotions. These days, profound learning procedures know a major accomplishment in different fields including PC vision. Without a doubt, a convolutional neural network (CNN) model can be prepared to break down pictures and recognize face feeling. we make a framework that perceives understudies' feelings from their appearances. Our framework comprises of three stages: face identification utilizing using opencv, standardization and feeling acknowledgment utilizing CNN on FER 2013 database with seven sorts of emotions. Acquired outcomes show that face feeling acknowledgment is plausible in training, thus, it can assist educators with modifying their introduction as per the understudies' feelings.

Keywords: Deep learning, convolution neural network, opencv, framework, User Interface, dataset

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

1. INTRODUCTION

human being are able to transmit many emotions without saying a word. Facial expression recognition identifies emotion from face image, it is a manifestation of the activity and personality of a human. In the 20th century, the American psychologists Ekman and Friesen defined six basics emotions (anger, fear, disgust, sadness, surprise and happiness), which are the same across cultures. Facial expression recognition has brought much attention in the past years due to its impact in clinical practice, sociable robotics and education. According to diverse research, emotion plays an important role in education. Currently, a teacher use exams, questionnaires and observations as sources of feedback but these classical methods often come with low efficiency. Using facial expression of students the teacher can adjust their strategy and their instructional materials to help foster learning of students. The purpose of this article isto implement emotion recognition in education by realizing an automatic system that analyze students' facial expressions based on Convolutional Neural Network (CNN), which is a deep learning algorithm that are widely used in images classification. It consist of a multistage image processing to extract feature representations. Our systemincludes three phases: face detection, normalization and emotion recognition thatshould be one of these seven emotions: neutral, anger, fear, sadness, happiness,surprise and disgust.

1.1 MOTIVATION

Many researchers are interested in improving the learning environment with Face Emotion Recognition (FER). Tang proposed a system which is able to analyze students' facial expressions in order to evaluate classroom teaching effect. The system is composed of five phases: data acquisition, face detection, face recognition, facial expression recognition and post-processing. The approach uses K-nearest neighbor (KNN) for classification and Uniform Local Gabor Binary Pattern Histogram Sequence (ULGBPHS) for pattern analysis. Savva proposed a web application that performs an analysis of students' emotion who participating in active face-to-face classroom instruction. The application uses webcams that are installed in classroomsto collect live recordings, then they applied machine learning algorithms on its.

1.2 Existing System

The authors proposed a system that identifies and monitors student's emotion and gives feedback in real-time in order to improve the e-learning environment for a greater content delivery. The system uses moving pattern of eyes and head to deduce relevant information to understand students' mood in an e-learning environment. Ayvaz developed a Facial Emotion Recognition System (FERS), which recognizesthe emotional states and motivation of students in videoconference type e-learning. The system uses 4 machine learning algorithms (SVM, KNN, Random Forest and Classification & Regression Trees) and the best accuracy rates were obtained using KNN and SVM algorithms. Kim proposed a system which is able of producing real-time recommendation to the teacher in order to enhance the memorability and the quality of their lecture by granting the teacher to make modification in real-time to their non-verbal behavior like body language and facial expressions. The authors proposed a model that recognizes emotion in virtual learning environment based on facial emotion recognition with Haar Cascades method to identify mouth and eyes on JAFF database in order to detect emotions.China used wireless sensor network technology to create an intelligent classroom management system that aids teachers tomodify instruction modes rapidly to avert wasting of time.

1.3 Limitations of existing system

- No proper prediction of emotion is not high accuracy

1.4 Objectives and Outcomes

The major objective is Facial Emotion Recognition of Students using Convolutional Neural Network The proposed model achieved an accuracy rate of 70% on FER 2013 database. Our facial expression recognition system can help the teacher to recognize students' comprehension towards his presentation.

1.5 Applications

Expression recognition has certain applications in the fields of education, medical care, transportation and communication, and has outstanding advantages in specific fields. • In the field of education, the technology can be applied to assist teachers in

teaching and measuring student learning levels, as well as monitoring the test room environment. The student model built using relevant techniques, is an intelligent counseling system (ITS) consisting of a student model and a diagnostic module.

1.6 STRUCTURE OF PROJECT (SYSTEM ANALYSIS)

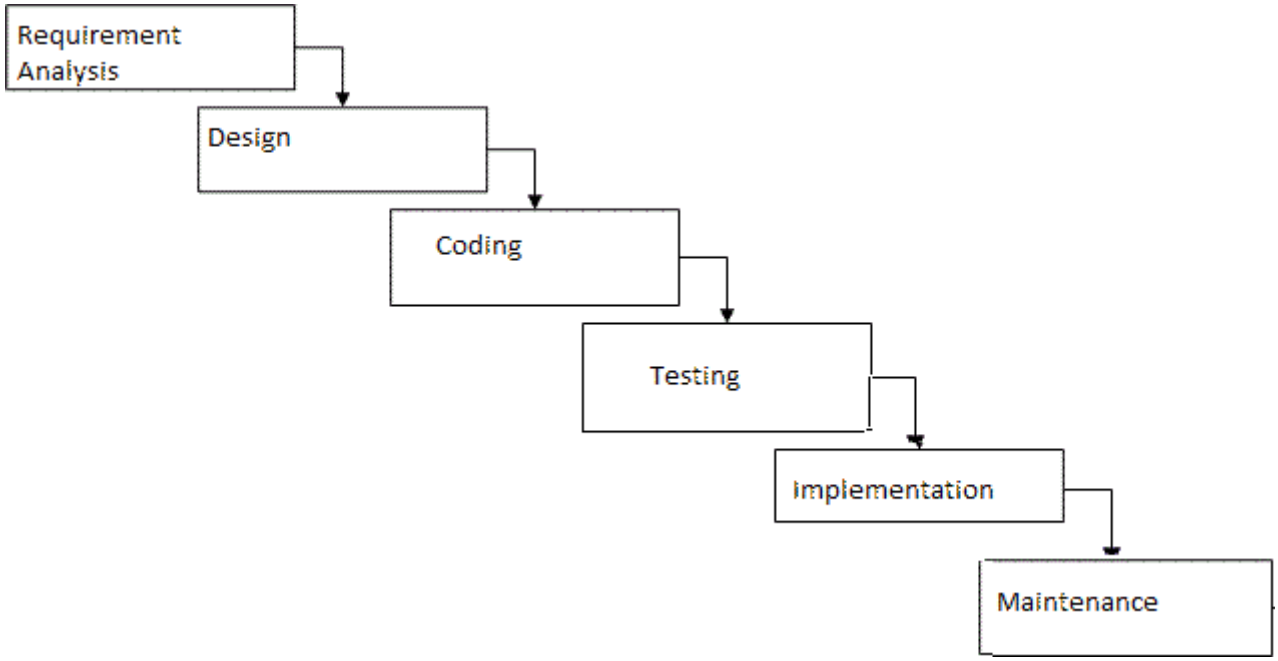

Figure 1.6.1 project SDLC

- Project Accumulating and Analysis
- Application System Design
- Practical coding
- Manual Testing of My Application
- Application Deployment of System
- Maintenance of the Project

2. SYSTEM ANALYSIS

2.1 EXISTING APPROACH:

The authors proposed a system that identifies and monitors student's emotion and gives feedback in real-time in order to improve the e-learning environment for a greater content delivery. The system uses moving pattern of eyes and head to deduce relevant information to understand students' mood in an e-learning environment. Ayvaz developed a Facial Emotion Recognition System (FERS), which recognizes the emotional states and motivation of students in videoconference type e-learning. The system uses 4 machine learning algorithms (SVM, KNN, Random Forest and Classification & Regression Trees) and the best accuracy rates were obtained using KNN and SVM algorithms. Kim proposed a system which is able of producing real-time recommendation to the teacher in order to enhance the memorability and the quality of their lecture by granting the teacher tomake modification in real-time to their non-verbal behavior like body language and facial expressions. The authors proposed a model that recognizes emotion in virtual learning environment based on facial emotion recognition with Haar Cascades method to identify mouth and eyes on JAFF database in order to detect emotions. China used wireless sensor network technology to create an intelligent classroom management system that aids teachers to modify instruction modes rapidly to avert wasting of time.

2.2 Drawbacks

- No proper prediction of emotion and no high accuracy

2.3Proposed System

In this section, we describe our proposed system to analyze students' facial expressions using a Convolutional Neural Network (CNN) architecture. First, the system detects the face from input image

and these detected faces are cropped and normalized to a size of 48×48. Then, these face images are used as input to CNN. Finally, the output is the facial expression recognition results.

2.4 Advantages

- The face emotion detection and prediction is high accurate and accuracy is high

2.5 Modules and Versions required:

tensorflow==1.14.0

ipykernel==5.3.4

scikit-image==0.17.2

scikit-learn==0.23.2

pandas==1.1.1

matplotlib==3.3.1

Keras==2.3.1

Pillow==7.2.0

plotly==4.10.0

opencv-python==4.4.0.42

spacy==2.3.2

lightgbm==3.0.0

mahotas==1.4.11

matplotlib==3.3.1lightgbm==3.0.0

mahotas==1.4.11

nltk==3.5

matplotlib==3.3.1

xgboost==1.2

3. REQUIREMENT SPECIFICATIONS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idel 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google colab**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system : windows, linux**
- **Processor : minimum intel i3**
- **Ram : minimum 4 gb**
- **Hard disk : minimum 250gb**

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.2 Global Use Case Diagrams:

Actor: An actor in use case modeling specifies a role played by a user or any other system that interacts with the subject. An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject.

Graphical representation:

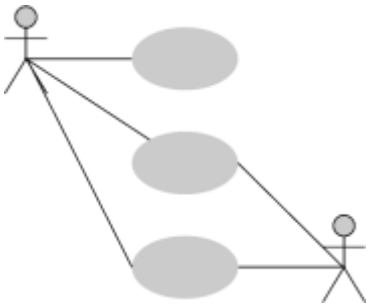

Fig 4.2.1 Actor

• An actor is someone or something that interacts with or uses the system.

• Provides input to and receives information from the system. Is external to the system and has no control over the use cases. Actors are discovered by examining:

• Who directly uses the system

Who is responsible for maintaining the system

4.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction. These internal and external agents are known as actors.Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

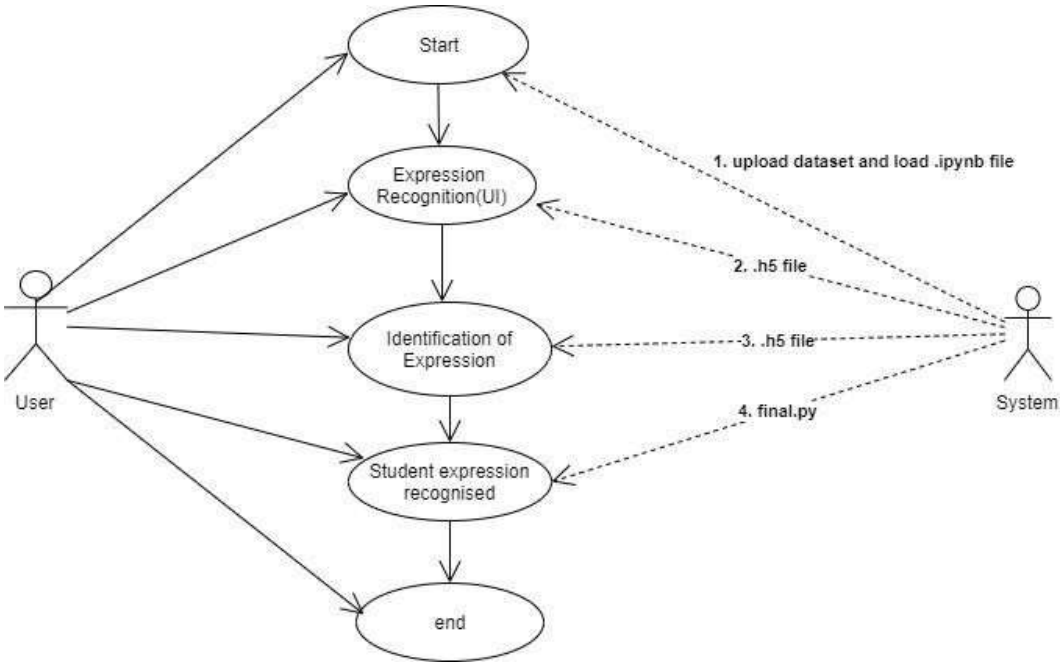

Figure 4.2.1 use case

4.4 Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join.It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows suchas parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

This UML diagram focuses on the execution and flow of the behavior of a system instead of implementation. Activity diagrams consist of activities that are made up of actions that apply to behavioral modeling technology.

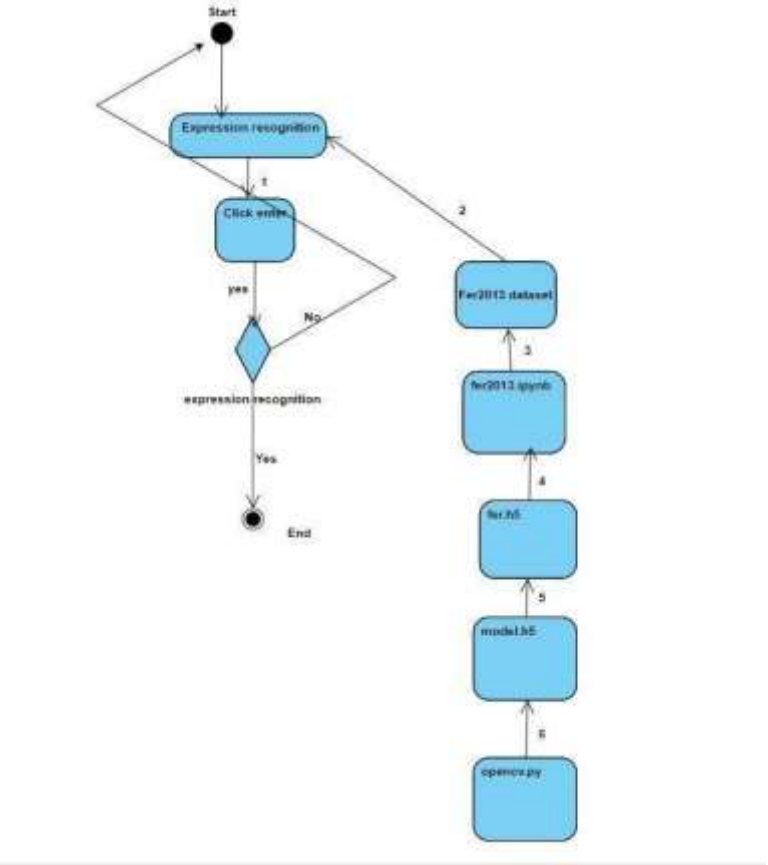

Figure 4.3.1: Activity Diagram

4.5 Architecture:

Convolutional neural networks (CNN) model can be trained to analyze images and identify face emotion. we create a system that recognizes students' emotions from their faces. Our system consists of phases: face detection using Opencv, normalization and emotion recognition using CNN on FER 2013 database with seven types of expressions. System detects the input image using opencv with in build UI, the detected faces will be cropped into 48*48 size and normalize the input image is possible, filter the input image using CNN. This means that network learns to filter and normalize in build available in CNN, the CNN contains layers like convolution , pooling, fully connected layer.

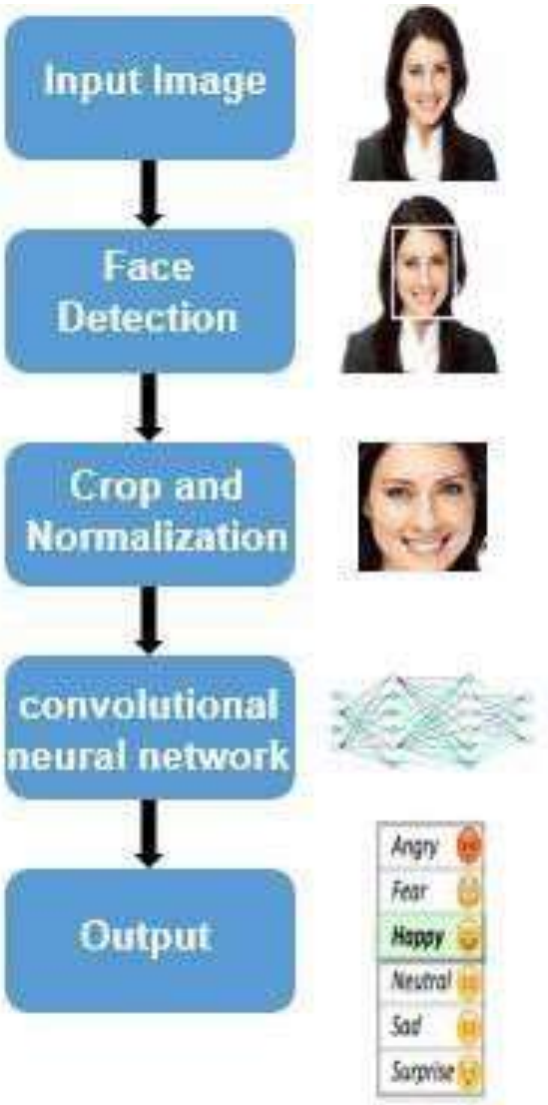

Figure 4.4.1 :Architecture

5. IMPLEMENTATION

5.1 Flow chart:

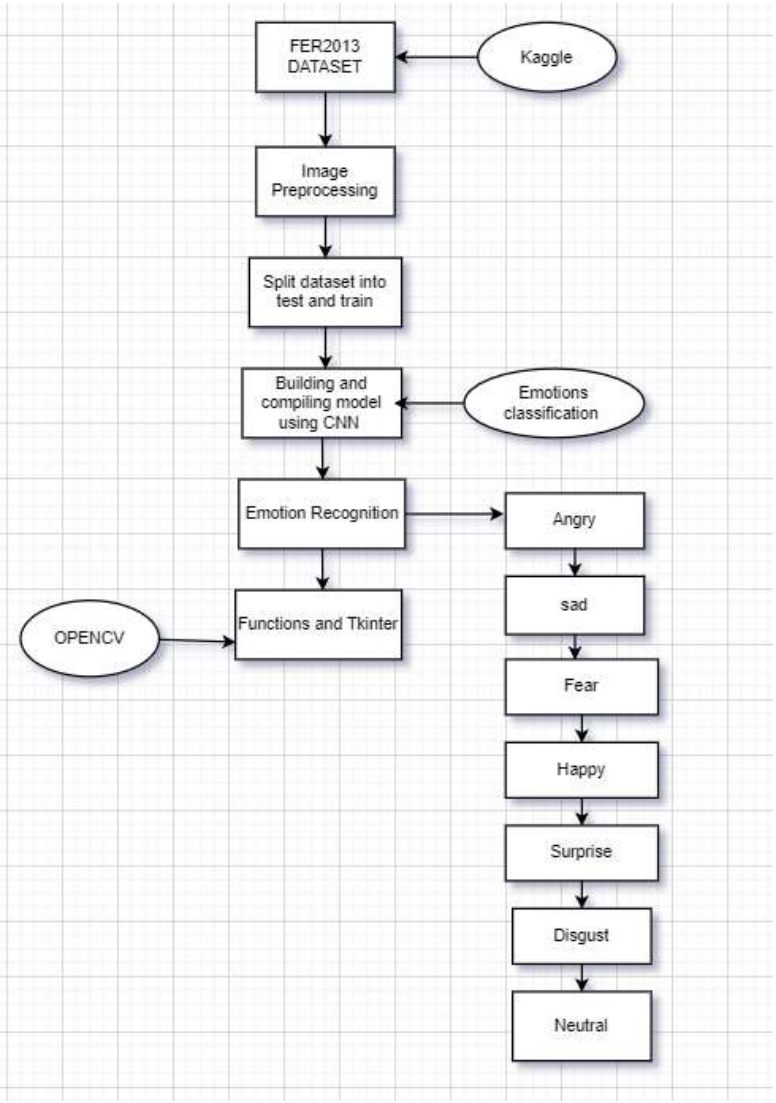

Fig 5.1.1 Flow chart

The architecture contains flow of work. The dataset is taken from internet i.e kaggle. The dataset used is FER2013 which consists of 35887 of seven sorts of emotional classes.

After collecting dataset load the model into platform and perform the preprocessing steps for image. The large dataset in divided into test and train , validate the test sets using the train and divide the images according to their category. CNN model build using tensor flowand keras. After building the model compile the model and accuracy is checked as metric tocheck either model is perfectly compiled or not . The classified images are accessed in test dataset and stored in .h5 file, load functions such as open cv and tkinter to capture and recognize the emotion using classified images in dataset. Thus , facial emotion is recognized.

5.2 Algorithm

The CNN algorithm is used in the facial emotion recognition of student .It is a deep learning algorithm used in form of neural networks, widely used in areas of image classification, object detection and face recognition.

Convolution Neural Network:

5.2.1 In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

5.2.2 CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of4 x 4 x 1 array of matrix of grayscale image.

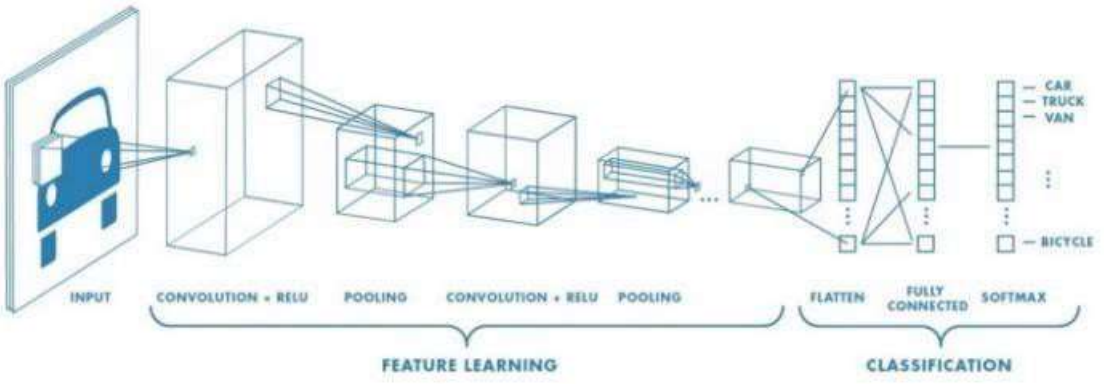

Figure 5.2.1:CNN MODEL

i. Convolution Layer:

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

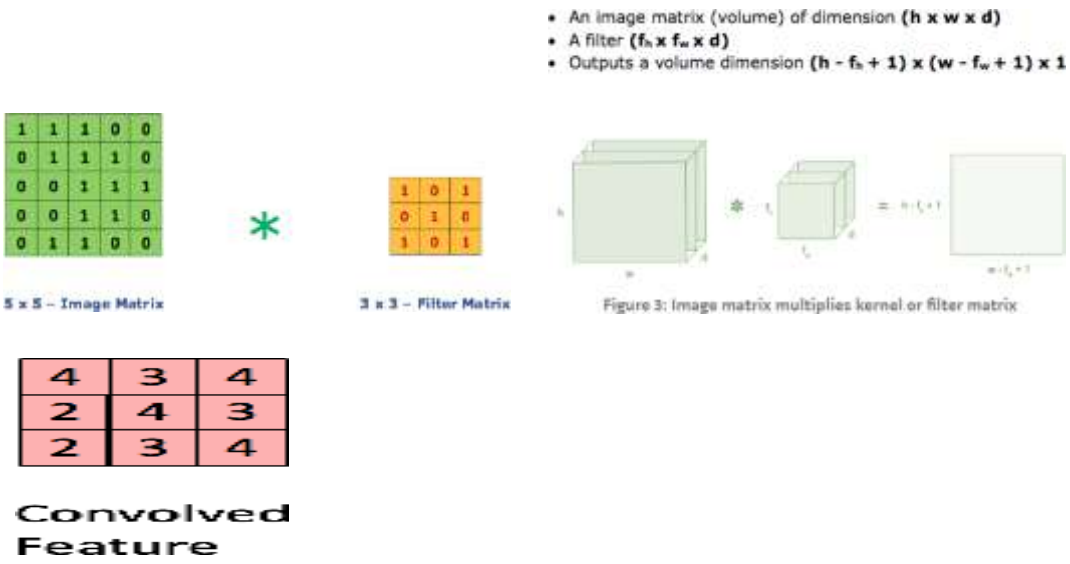

Fig 5.2.2 Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is Feature map

ii. Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

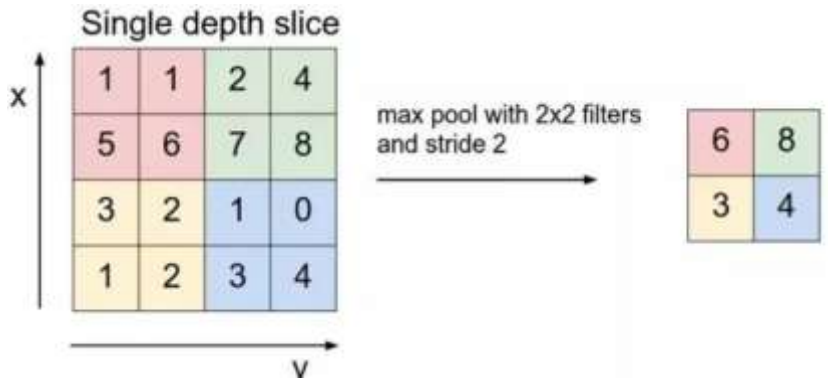

Figure 5.2.3:Max pooling

iii. Fully Connected Layer

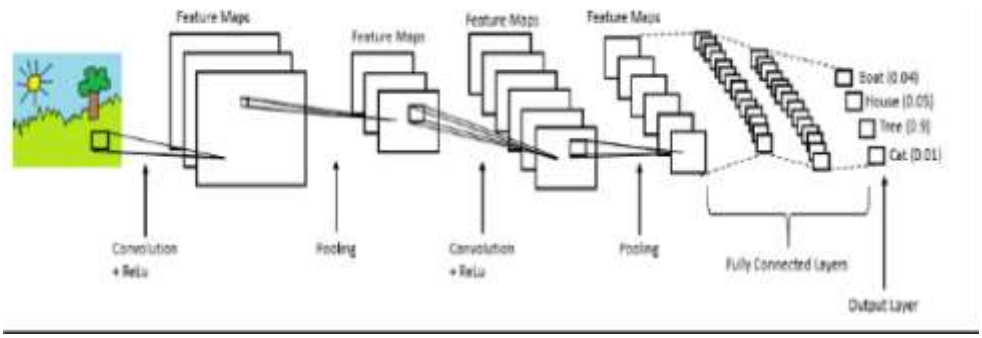

Figure 5.2.4:FC LAYER

- Provide input image into convolution layer
- Choose parameters, apply filters with strides, padding if requires. Perform convolution on the image and apply ReLU activation to the matrix.
- Perform pooling to reduce dimensionality size
- Add as many convolutional layers until satisfied

- Flatten the output and feed into a fully connected layer (FC Layer)
- Output the class using an activation function (Logistic Regression) and classifies images.

```
model = Sequential()
model.add(Conv2D(filters = 64, kernel_size = (5,5), input_shape = (48, 48, 1), padding='same'))
model.add(Conv2D(64, kernel_size=(5,5), padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Activation('relu'))

model.add(Conv2D(filters=128, kernel_size=(5,5), padding='same'))
model.add(Conv2D(128, kernel_size=(5,5), padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Activation('relu'))

model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same'))
model.add(Conv2D(256, kernel_size=(3,3), padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Activation('relu'))

model.add(Flatten())
model.add(Dense(128))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(rate=0.25))
model.add(Dense(7, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Figure 5.2.5:CNN layers

5.3 Source code

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import numpy as np
import argparse
import matplotlib.pyplot as plt
import cv2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
```

```
global model
main = tkinter.Tk()
main.title("Facial Expression Detection")
main.geometry("500x400")
def cnnmodel():
    global model
    model = Sequential()

    model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
    model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))

    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(7, activation='softmax'))
    model.load_weights('model.h5')

    return model

def pred():
    global model
    model = cnnmodel()
    # prevents openCL usage and unnecessary logging messages
    cv2.ocl.setUseOpenCL(False)

    # dictionary which assigns each label an emotion (alphabetical order)
    emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}
    # start the webcam feed
    cap = cv2.VideoCapture(0)
    while True:
        # Find haar cascade to draw bounding box around face
        ret, frame = cap.read()

            if not ret:
                break

            facecasc = cv2.CascadeClassifier('C:/Users/krazy/anaconda3/envs/tf/Lib/site-
        packages/cv2/data/haarcascade_frontalface_default.xml')
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            faces = facecasc.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=5)
```

```
        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
            roi_gray = gray[y:y + h, x:x + w]
            cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray, (48, 48)), -1), 0)
            prediction = model.predict(cropped_img)
            maxindex = int(np.argmax(prediction))
            cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,
255, 255), 2, cv2.LINE_AA)
        cv2.imshow('Video', cv2.resize(frame,(1600,960),interpolation = cv2.INTER_CUBIC))
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()

font = ('times', 16, 'bold')
title = Label(main, text='Facial Emotion Recognition')
title.config(bg='dark salmon', fg='black')
title.config(font=font)
#title.config(height=3, width=120)
title.place(x=100,y=50)

font1 = ('times', 14, 'bold')
upload = Button(main, text="Facial Expression Detection", command=pred)
main.config(bg='seashell3')
main.mainloop()
```

6. TESTING

6.1 SOFTWARE TESTING

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

6.2 White Box Testing

- White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.
- It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.
- White box testing involves the testing of the software code for the following:
- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis
- The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

A major White box testing technique is Code Coverage analysis. Code Coverage analysis eliminates gaps in a Test Case suite. It identifies areas of a program that are not exercised by a set of test cases. Once gaps are identified, you create test cases to verify untested parts of the code, there by increasing tquality of the software product

- There are automated tools available to perform Code coverage analysis.
  Below are a few coverage analysis techniques a box tester can use:
- **Statement Coverage**:- This technique requires every possible statement in the code to be tested at least once during the testing process of software engineering.
- **Branch Coverage** - This technique checks every possible path (if-else and other conditional loops) of a software application.
- Apart from above, there are numerous coverage types such as Condition Coverage, Multiple Condition Coverage, Path Coverage, Function Coverage etc. Each technique has its own merits and attempts to test (cover) all parts of software code. Using Statement and Branch coverage you generally attain 80-90% code coverage which is sufficient.
- **White Box Penetration Testing**: In this testing, the tester/developer has full information of the application's source code, detailed network information, IP addresses involved and all server information the application runs on. The aim is to attack the code from several angles to expose security threats

White Box Mutation Testing: Mutation testing is often used to discover the best coding techniques to use for expanding a software solution.

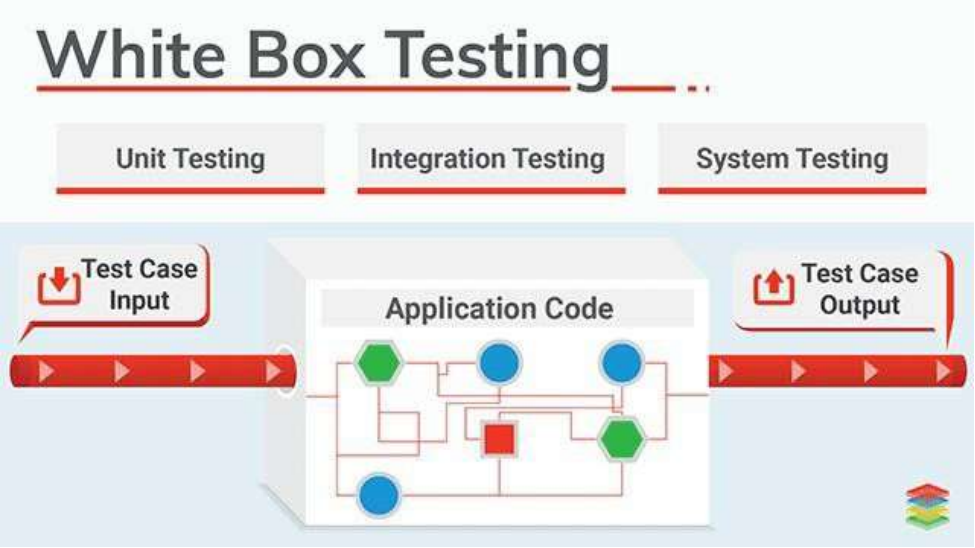

Fig 6.2.1 White Box Testing

6.3 Output Testing

```
C:\Users\krazy>cd C:\Users\krazy\Downloads\Facial Emotion Recognition of Students using Convolutional Neural Network

C:\Users\krazy\Downloads\Facial Emotion Recognition of Students using Convolutional Neural Network>conda activate tf

(tf) C:\Users\krazy\Downloads\Facial Emotion Recognition of Students using Convolutional Neural Network>python final.py
```

Fig 6.3.1 tensorflow backend

- Open the command prompt and go to the directory of the source code python file using tensor flow by activating the tf, run the python file to open tkinter application

Figure 6.3.2 tkinter UI

- Click on facial expression detection to identify the emotions on the tkinter UI

- **Steps to open output:**

1. Open Facial emotion recognition
2. Click on Facial expression detection
3. Above screen will be opened.
4. Which will open live capture using opencv
5. Then emotion is recognized

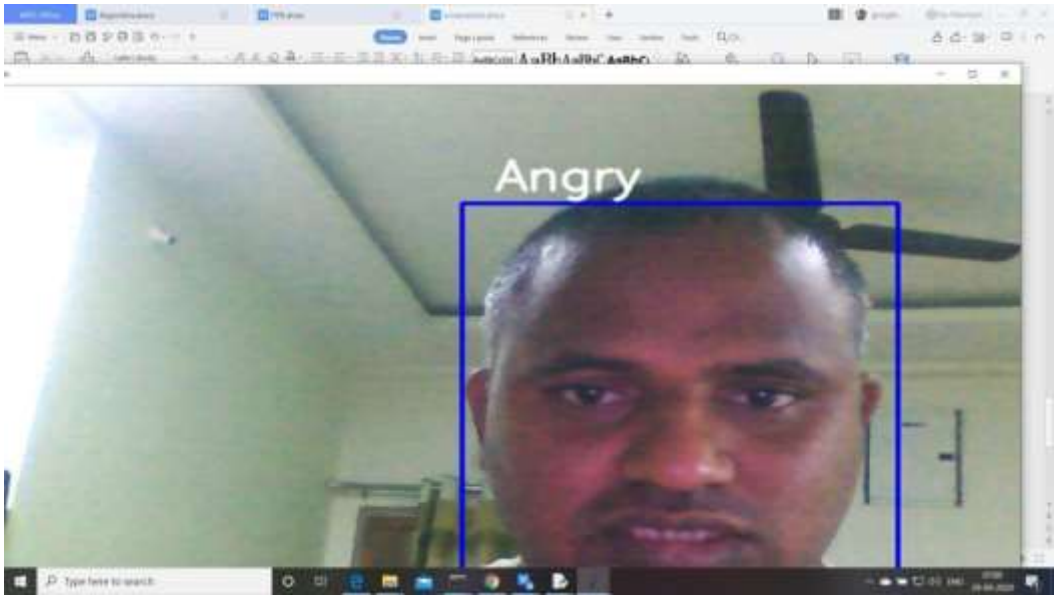

Figure 6.3.3 Emotion Angry

- Emotion Angry is captured live using opencv interface

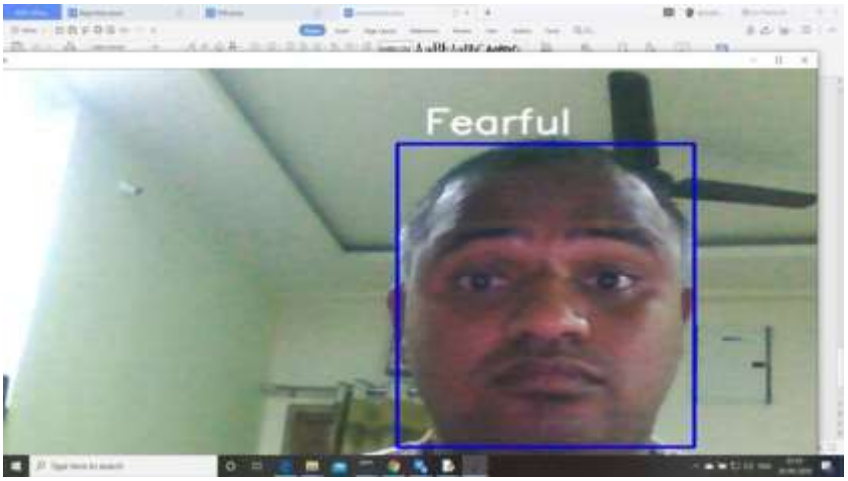

Figure 6.3.4 Emotion Fearful

- Emotion fearful is captured live using opencv

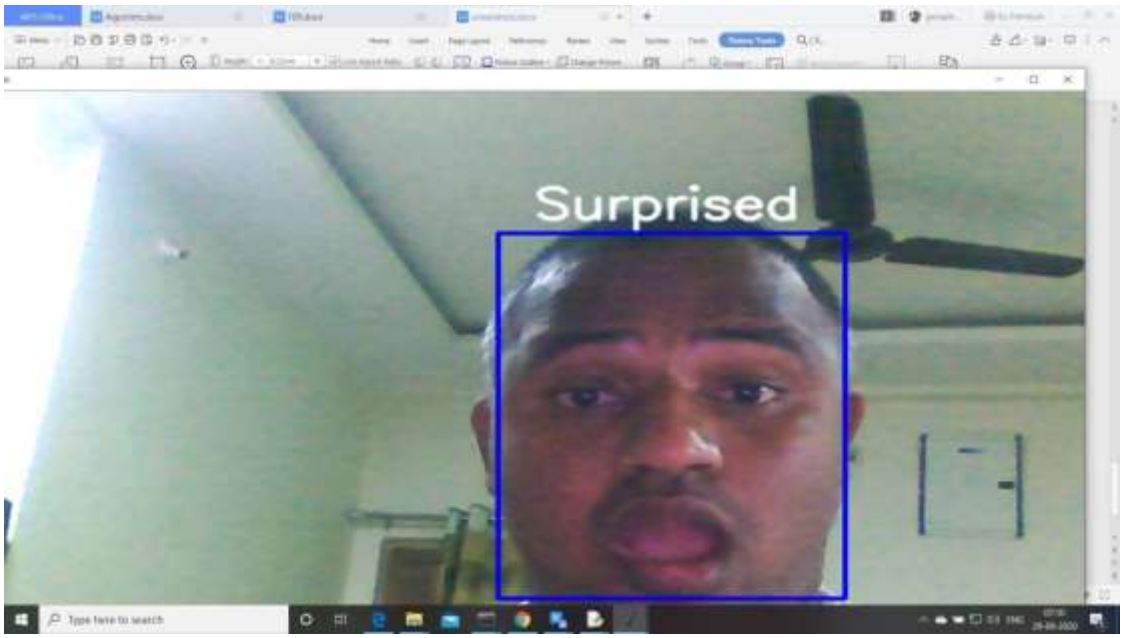

Figure 6.3.5 Emotion Surprised

Figure 6.3.6 Emotion Neutral

7.RESULTS ANALYSIS

7.1 Results

The model is executed with 20 epoch over 32298 of training data and 3589 of testing orvalidation data. For 20 epochs, batch size of 64 is given and learning rate is given as 0.0001. After 20 epochs, we got accuracy as 0.82% and loss as 0.50%. We took accuracy as metric by accuracy the proposed model is getting better results and able to classify emotions and recognize emotions using open cv . The train and test values are converted into categorical values because the data consists of multiple categories and also during the compiling of model to achieve good accuracy and also contains columns equal to number of categorical data. The have used 3 input layer or feature learning and 1 output layer or classification layer in CNN model. The input shape (h*w*d) height ,width
,depth is given as 48,48,1 and padding is given same where the input and filtered image will have same padding so, there is no change dimensionality of the image. The kernel size is given as 5,5 and it changes with every layer. Batch normalization is used to scale and adjust the input values and train deep neural network standardizes input to each batch. Flatten is used for the output layer which converts the input or feature map image into 1- dimensional array to next layer and also creates single long vector to fully connected layer. Dropout is used to drop pixels which has huge weights, over fits and also having bad learning rate. We use RELU activation in input layers and soft max activation in output layer or dense layer to get good accuracy. The dense with 128 is used in the output layer, adam optimizer is used instead of stochastic gradient decent parser to update network weight based in training data and also used for hyper-parameter. The model without using transfer learning shows less accuracy
i.e we used metric as accuracy shows less accuracy compare to CNN model for image classification and also for recognition.

Model	Accuracy
Model using CNN	82.5%
Model without using transfer learning	61.2%

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 64)        1664
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 64)        102464
_________________________________________________________________
batch_normalization (BatchNo (None, 48, 48, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 64)        0
_________________________________________________________________
activation (Activation)      (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 128)       204928
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 128)       409728
_________________________________________________________________
batch_normalization_1 (Batch (None, 24, 24, 128)       512
=================================================================
Total params: 2,787,015
Trainable params: 2,785,863
Non-trainable params: 1,152
```

Figure 7.1.1 Model summary of CNN model

- Here, total parameters is 2787015 in which trainable are 2785863 and non-trainable are 1152
- Model = sequencial 4 layers of CNN algorithm

```
from tensorflow.keras.utils import to_categorical
n_epochs = 20
batch_size = 64
lr = 0.0001
```

```
history = model.fit(X_train32, to_categorical(y_train),
                    batch_size = batch_size, epochs = n_epochs, validation_data= (X_test32, to_categorical(y_test)))
```

```
Train on 32298 samples, validate on 3589 samples
Epoch 1/20
32298/32298 [==============================] - 1662s 51ms/sample - loss: 1.6533 - accuracy: 0.3613 - val_loss: 1.4998 - val_acc
uracy: 0.4461
Epoch 2/20
32298/32298 [==============================] - 1666s 52ms/sample - loss: 1.3013 - accuracy: 0.5006 - val_loss: 1.4150 - val_acc
uracy: 0.4472
```

Figure 7.1.2 Model compilation and Classification Accuracy

- Using tensorflow package, fit the train and test model with batchsize=64, total number of epochs=20 and learning rate is 0.0001
- The model is converted into categories i.e classification of model takes place using to_categorical

```
...] - ETA: 16:34 - loss: 0.5025 - accuracy: 0.8205
```

```
model.save('fer.h5')
```

Figure 7.1.3 Model accuracy and model is saved into fer.h5 file

- The Final accuracy of model is 82.5%
- The model is save into .h5 file i.e hierarical data format version 5(HDF5) which is open source file format supports large ,complex data,
- One HDF5 file can store similar set of data organized in same way that you organize files, one important feature is to attach metaset to everyday in file then it provides powerful searching and accessing, numpy must be installed in HDF5.

8. CONCLUSION

In the paper we proposed deep learning Convolution neural network architecture with training on 32298 images of emotions and testing on 3589 images of emotions and correctly classifies 82.5% from the proposed model in the paper we are able to say that CNN algorithms are producing good results for image dataset. The number of epochs used are 20.The proposed method can directly takes image values as input through training sample image data. The proposed method can automatically learn pattern features through CNN and reduce the incompleteness effected by artificial design features. By improvingthe training images we achieve better results. Proposed algorithm can improve the recognition rate of student facial expressions in complex background also.

In, future autonomous learning also applied to extract the facial emotions and expressions more accurately and also new type of expressions can also be added in the dataset. The teacher can easily identify the student's current emotion and can enact or deal according to the situation by proposed algorithm. Autonomous learning can implicitly acquire more abstract feature expression of the image. We can also develop android and windows based GUI using tkinter which is consists of open cv frameworks built in it. We can develop and implement application which is easy to use and carry, provided by tkinter GUI and open cv framework to the classrooms for the teachers or instructors to recognise the students emotions.