**Project-I Report on**

# FACIAL EXPRESSION RECOGNITION

**Submitted to partial fulfillment of the academic requirements of**
**Jawaharlal Nehru Technological University Hyderabad**
**For the award of the degree of**

**Bachelor of Technology**

**in**

**Electronics and Computer Engineering**

**(2019 – 2023)**
**Submitted By**


**G.SHIRISHA (19311A1914)**
**A.MANI CHANADANA (19311A1927)**
**B.ANKITHA (19311A1956)**


**Under the Guidance of**


**Dr. MANU GUPTA**
**Internal Guide**
**Assistant Professor, Department of ECM**




**DEPARTMENT OF**

**ELECTRONICS AND COMPUTER ENGINEERING**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**
*(Autonomous)*
Yamnampet, Ghatkesar Mandal, Medchal District, Hyderabad - 501301

SNIST
Sreenidhi Institute of Science and Technology
*Because, Life is all about taking the right decisions*
AUTONOMOUS

# CERTIFICATE

This is to certify that this Group Project report on "**FACIAL EXPRESSION RECOGNITION**", submitted by G.SHIRISHA (19311A1914), A.MANI CHANDANA (19311A1927) and B.ANKITHA (19311A1956) towards partial fulfillment for the award of Bachelors degree in Electronics and Computer Engineering from Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad,is a record of bonafide work that has been done by them. This report has not been submitted to any other institute or university for the award of any degree.

| | | |
|---|---|---|
| **Dr.Manu Gupta** | **Mrs. K.Aruna Kumari** | **Dr. Mohan** |
| **Assistant Professor** | **Assistant Professor** | **Professor & HOD** |
| **Department of ECM** | **Department of ECM** | **Department of ECM** |
| **Internal Guide** | **Project Coordinator** | |

**Date:** **External Examinar:**

# DECLARATION

We, G.SHIRISHA (19311A1914), A.MANI CHANDANA (19311A1927) and B. ANKITHA(19311A1956),students of SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, Yamnampet, Ghatkesar, studying IV year 1st semester, Electronics and Computer Engineering solemnly declare that the Project-I work, titled "FACIAL EXPRESSION RECOGNITION USING NEURAL NETWORK" is submitted to SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY for partial fulfillment for the award of degree of Bachelor of technology in ELECTRONICS AND COMPUTER ENGINEERING.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

# ACKNOWLEDGEMENT

We would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

We would like to express our heart-felt gratitude to our parents without whom we would not have been privileged to achieve and fulfill our dreams. We are grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably runs the institution and has had the major hand in enabling us to do this project.

We profoundly thank **Dr**. **Mohan**, Head of the Department of Electronics & Computer Engineering who has been an excellent guide and also a great source of inspiration to our work.

We would like to thank our internal guide **Dr.Manu Gupta** for her technical guidance, constant encouragement and support in carrying out our project at college.

We also extend our gratitude to our project coordinator **Mrs.K.Aruna Kumari**, who lent us valuable guidance and led us towards the staged completion of the project.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**G.SHIRISHA 19311A1914**
**A.MANI CHANDANA 19311A1927**
**B.ANKITHA 19311A1956**

# FACIAL EXPRESSION RECOGNITION USING CONVOULTIONAL NEURAL NETWORKS


## Abstract

Various Facial expressions contribute highly in conveying the feelings of a person. In this project we took the facial expression image dataset consists of 7 different basic emotions of the human beings for building a Deep Learning based model using computer vision model architectures for classification of facial expressions.

Preprocessed images using augmentation techniques were trained our custom built Convolutional Neural Network (CNN) layers of each image of all categories by passing their weights. Unlike using networks like VGG16, Xception and ResNet50, our custom sequential layered model gave the training accuracy and validation accuracy which are far better than other networks. Feature extraction is used to extract the most prominent parts of the face, including the jaw, mouth, eyes, nose, and eyebrows. We obtained a test accuracy of 68.84% on FER2013 in a seven-classes classification task compared to 82.33% in state-of-the-art classification.



Keywords—Facial Expression Recognition (FER), Convolutional Neural Networks (CNNs), Artificial Intelligence (AI), Facial Action Coding System (FACS), Pre-processing, Feature Extraction.

# INDEX

## LIST OF FIGURES

# 1.INTRODUCTION

As we are stepping forward from one generation to another, innumerous technologies are abiding us according to our necessities. Thus, we are thoroughly depending on these technologies as a part of human-computer interaction. And one of them is facial expression recognition. Face plays an important role in social communication, equally facial expressions are vital. Facial expressions not only exposes the sensitivity or feelings of any person but can also be used to judge his/her mental views. Facial expression recognition is a method to recognize expressions on one's face. A wide range of techniques have been proposed to detect expressions like happy, sad, fear, disgust, angry, neutral, surprise but others are difficult to be implemented. Facial expression recognition is composed of three major steps: Face detection and preprocessing of image, Feature extraction and Expression classification. This project is organized in six sections and the second section includes the basic terminologies which are essential to understand for both face recognition and facial expression recognition. The third section of this project includes the difference between the face recognition and facial expression recognition. The fourth section explains about the procedure being followed for the recognition of facial expressions. The fifth section includes a review of ten previous researches in the expression recognition using various techniques. The sixth section is conclusion and it is about acknowledging the facial expression rate above 82%, calculated from the collected review.

## 1.1 Scope

1. In order to prevent the frauds of ATM in India, it is recommended to prepare the database of all ATM customers with the banks in India & deployment of high resolution camera and face recognition software at all ATMs. So, whenever user will enter in ATM his photograph will be taken to permit the access after it is being matched with stored photo from the database.
2. Duplicate voter are being reported in India. To prevent this, a database of all voters, of course, of all constituencies, is recommended to be prepared. Then at the time of voting the resolution camera and face recognition equipped of voting site will accept a subject face 100% and generates the recognition for voting if match is found.
3. Passport and visa verification can also be done using face recognition technology as explained above.
4. Driving license verification can also be exercised face recognition technology as mentioned earlier.
5. To identify and verify terrorists at airports, railway stations and malls the face recognition technology will be the best choice in India as compared with other biometric technologies since other technologies cannot be helpful in crowdy places.
6. In defense ministry and all other important places the face technology can be deployed for better security.

7. This technology can also be used effectively in various important examinations such as SSC, HSC, Medical, Engineering, MCA, MBA, B- Pharmacy, Nursing courses etc. The examinee can be identified and verified using Face Recognition Technique.

8. In all government and private offices this system can be deployed for identification, verification and attendance.

9. It can also be deployed in police station to identify and verify the criminals.

10. It can also be deployed vaults and lockers in banks for access control verification and identification of authentic users.

## 1.2 Existing System

Though the existing systems are computer based, they are not connected onto the Real world through applications i.e., classroom monitoring. Even if they are employed in the different usecases, they provide the significant accuracy on the real time hence accurate predictions are very less.

The following are the drawbacks of the existing manual System:

1. Scope for redundancy
2. Time Delay
3. No accuracy
4. Age old technique ie, Facial landmark detection and Haar features.

## 1.3 Proposed System

Proposed system works with Convolutional Neural Network and custom build computer vision architecture.The model is trained with augmentation methods and generate good accuracy among all transfer learning and predefined architectures.

We can make a web application to upload an image and predict the of emotion in it.

## 2. LITERATURE SURVEY

A.Zhiding Yu and Cha Zhang,2010.[1] This paper The proposed method contains a face detection module based on the ensemble of three state-of-the-art face detectors, followed by a classification module with the ensemble of multiple deep convolutional neural networks.

B.P. Viola and M. J. Jones, [3] This paper describes a face detection framework that is capable of processing images extremely rapidly while achieving high detection rates. A set of experiments in the domain of face detection is presented. The system yields face detection performance comparable to the best previous systems.

C.Pushpaja V. Saudagare,[4] Classification of face detection and token matching can be carried out any neural network for recognizing the facial expression. This paper reviews various techniques of facial expression recognition systems using MATLAB (neural network) toolbox.

D.M. Xiaoxi, L. Weisi, H. Dongyan, D. Minghui and H. Li,[6] This paper describes the fundamental phases involved in the emotion detection process. It explores the various face emotion detection schemes like machine learning, deep learning and others employed in literature and compares the diverse emotion recognition approaches in terms of FER techniques, datasets, emotions, number of emotions and publication years.

E.Neha Jain, Shishir Kumar, Amit Kumar, Pourya Shamsolmoali, Masoumeh Zareapoor,[7] This work proposes a Hybrid Convolution-Recurrent Neural Network method for FER in Images. The proposed network architecture consists of Convolution layers followed by Recurrent Neural Network which the combined model extracts the relations within facial images and by using the recurrent network the temporal dependencies which exist in the images can be considered during the classification.

F.A. Fathallah, L. Abdi and A. Douik, [8] Although there is a way to use machine learning and manual recognition of expressions intelligence technology, this paper tries to point out deep learning and image classification methods to recognize expressions and categorize expressions based on images.

# 3.SYSTEM ANALYSIS

This System Analysis is closely related to requirements analysis. It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made." This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

## 3.1 Functional Requirement Specification

The System after careful analysis has been identified to be present with the following modules.

### 3.1.1 User Module:

The user uploads the image that asked by the model which is in suitable form in terms of png, jpg, other image forms. The model takes this image and stores in its respective places.

### 3.1.2 Model Module:

This module maintains all the information to preprocess it according to the training data and displays it to the user for confirmation. The trained model get imported in-order to predict.

### 3.1.2 Display Module:

This module is an interface between user and the model. The model gives the accuracy of the different classes that the given image belongs. The module finds the maximum support from that and maps to the corresponding feeling.

## 3.2 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

1. The system should be able to interface with the existing system

2. The system should be accurate
3. The system should be better than the existing system
4. The existing system is completely dependent on the user to perform all the duties.

## 3.3 Software Requirements:

Operating System: Microsoft Windows XP.
Language : Python
Frameworks : Tensorflow
Libraries : OpenCV, Scikit-Learn
IDE : Google Colab

## 3.4 Hardware Requirements:

Processor : Intel P-IV based system

RAM : Min. 512 MB

# 4.SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.

## 4.1 Architectural Design

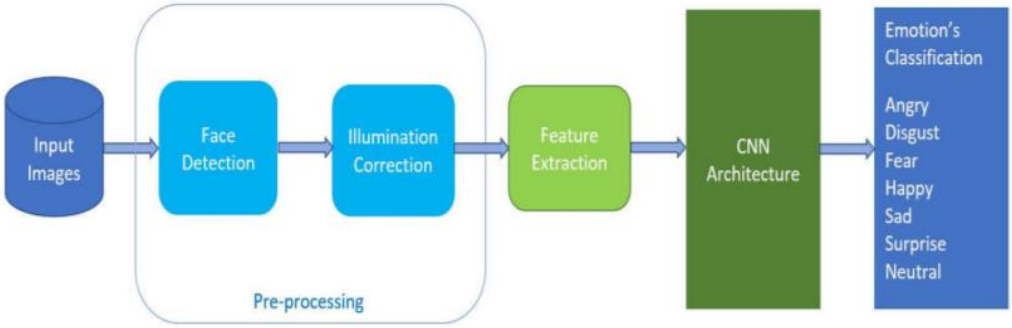


Fig 4.1 : Architectural Design

CNNs have been widely used in a variety of computer vision applications, including FER. Early in the 21st century, several studies of FER literature, determined that CNNs work well on changes of face location, as well as variations in scale. They were also found to work better than multilayer perceptron (MLP) when looking at face pose variations not seen previously. Researchers used CNN to help solve various facial expression recognition problems, such as translation, rotation, subject independence, and scale invariance. Fig shows that finding an optimized architecture or network structure is a very challenging task. Our model was trained using the following characteristics:
1.six convolutional layers using "RELU" as an activation function;
2.three max-pooling: out of which the first two using pool size (3,3) and stride (2,2), and the third using pool size (2,2) and stride (2,2); every max pooling is followed by every two convolutional layers.

## 4.2. Modules

### 4.2.1.User Module:

The user uploads the image that asked by the model which is in suitable form in terms of png, jpg, other image forms. The model takes this image and stores in its respective places.

- Import libraries
- Upload image of right extension
- Avoid using high quality

### 4.2.2. Model Module:

This module maintains all the information to preprocess it according to the training data and displays it to the user for confirmation. The trained model get imported in-order to predict.

- Import the trained model.
- Preprocess the image.
- Display it for confirmation.

### 4.2.3. Display Module:

This module is an interface between user and the model. The model gives the accuracy of the different classes that the given image belongs. The module finds the maximum support from that and maps to the corresponding feeling.

- Display the confidence and support
- Maps the emotion and display with picture.

## 4.3 UML Diagrams

UML Diagrams for our application are as follows:

### 4.3.1 Use Case Diagrams

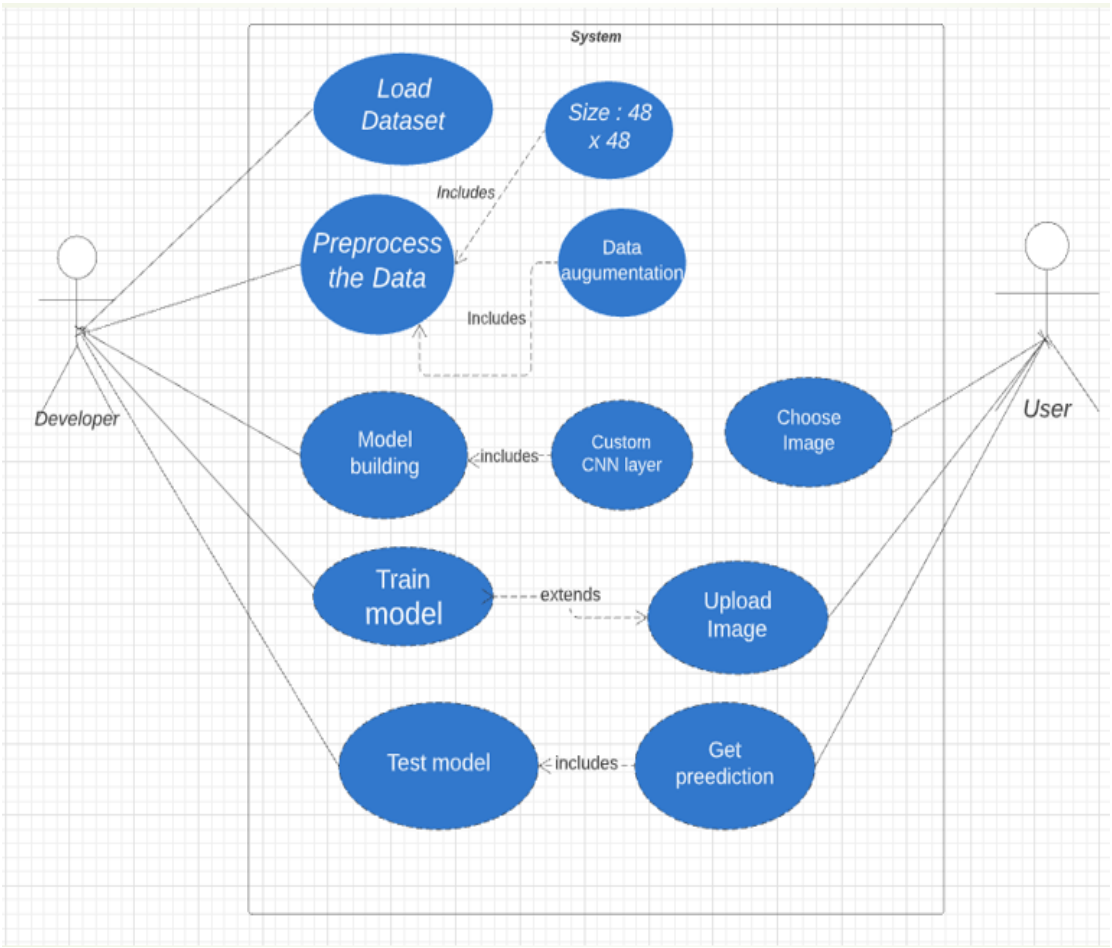


Fig 4.2: Usecase Diagram for FER

This UseCase Diagram depicts the functionality of all the modules in the project. The user interacts with the model by uploading the image with right extention. The developer initially developed a model and train it with the dataset and gain a suitable accuracy and save the model and further load it for the predictions. The model predicts the image into the respective class of feeling.

### 4.3.2 Class Diagram

:

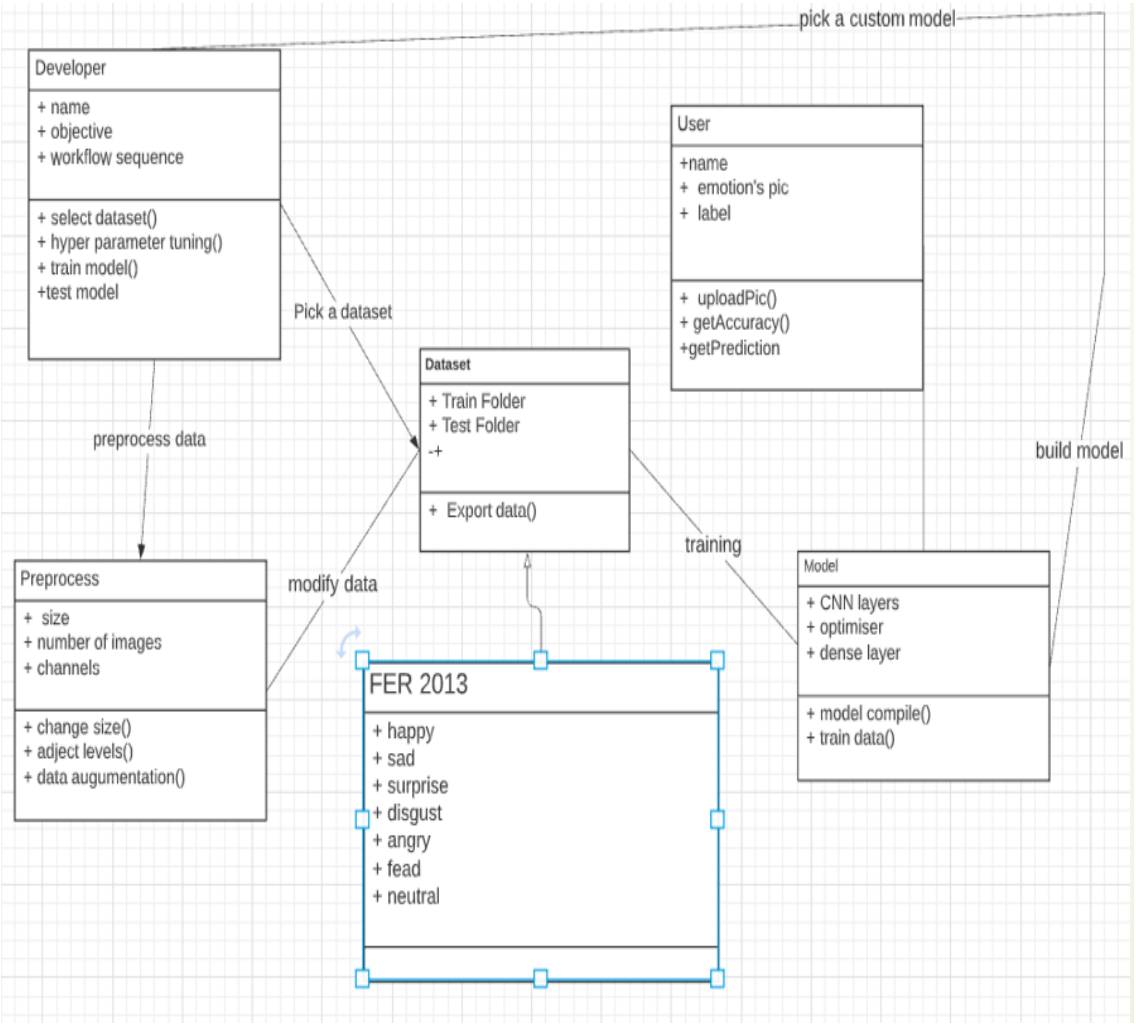


Fig 4.3 class diagram

In class diagram, we have the nouns which interacts with each other and provides the functionality to the user. We have the classes like Dataset, User, Model, Preprocess, Developer, FER model. Each have the different properties and behavior and connect with the other classes to provide the integrity.

### 4.3.3 Sequence Diagrams

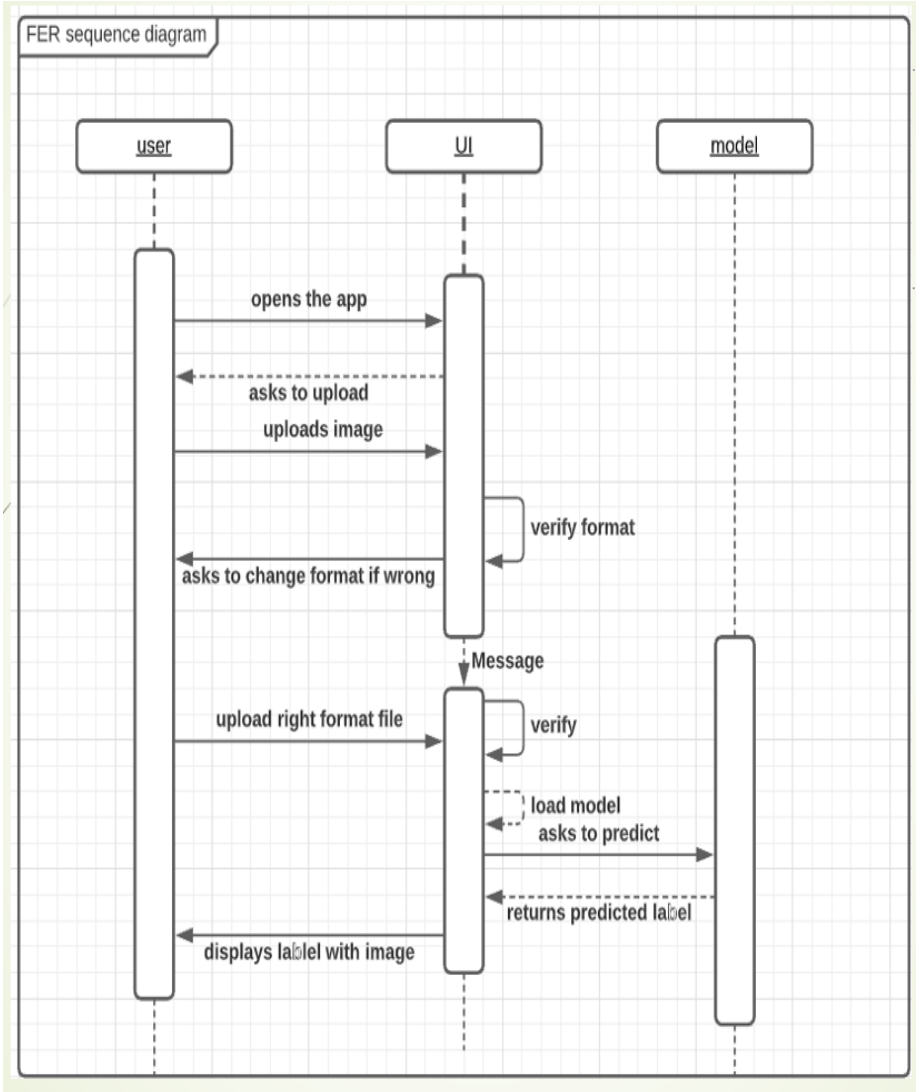


Fig 4.4 Sequence Diagram for Administrator's Module

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

### 4.3.4 Activity Diagrams

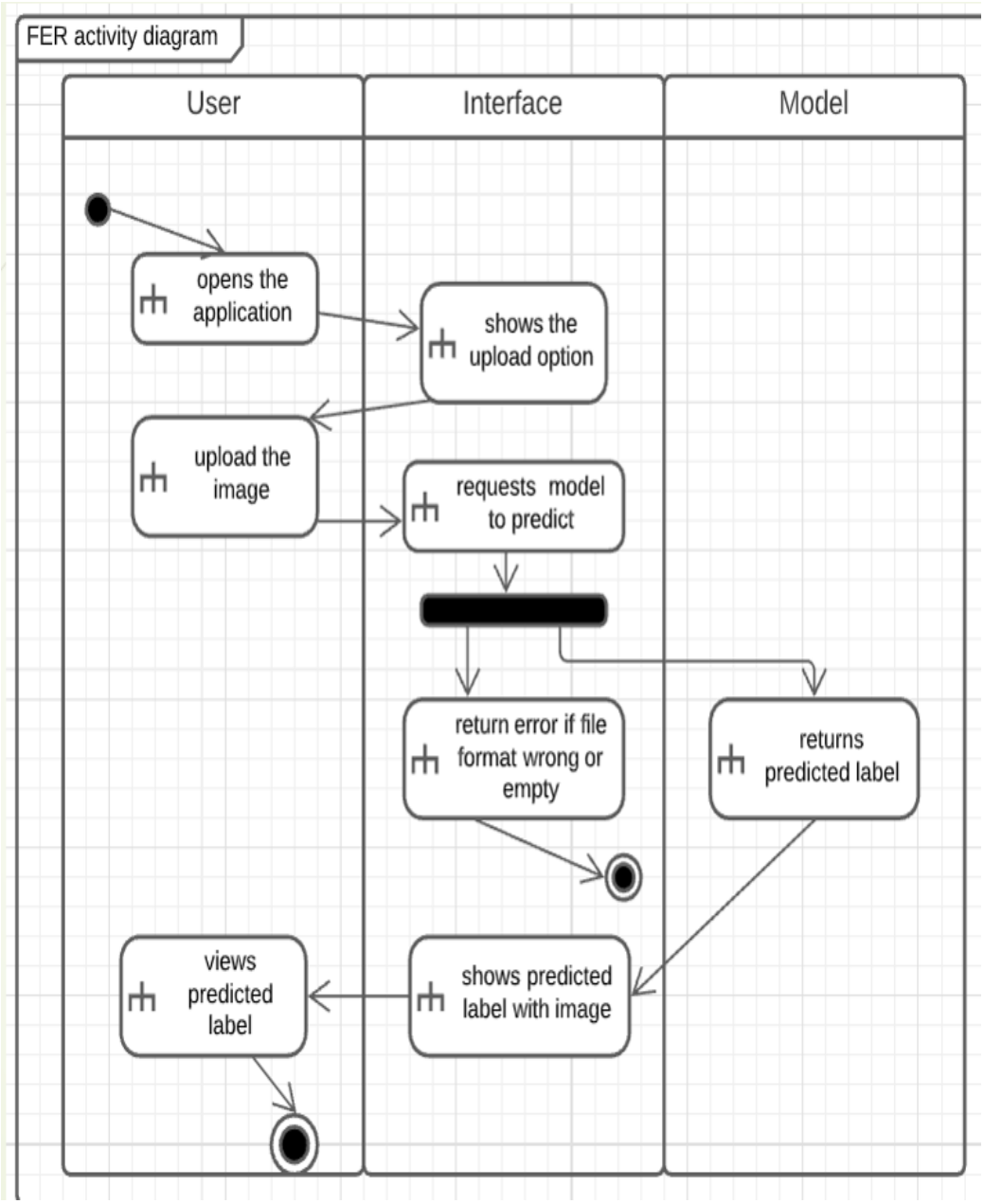


Fig 4.5: Activity Diagram

# 5.SYSTEM IMPLEMENTATION

The implementation stage of any project is a true display of the defining moments that make a project a success or a failure. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

## 5.1 Algorithm and Work Flow

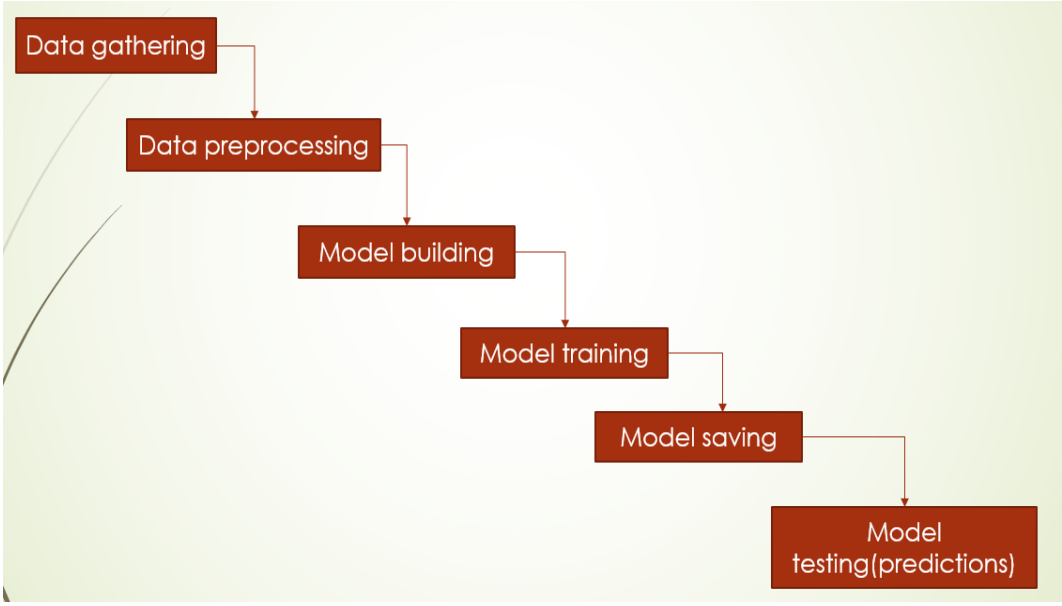


Fig 5.1 Work flow

## 5.2 Data preprocessing

Steps Involved in Data Preprocessing:
Step-1:DataCleaning:
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.
Step-2:Missing Data:
This situation arises when some data is missing in the data. It can be handled in various ways.
Some of them are:
1.Ignore the tuples:
This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

2.Fill the Missing values: There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

Step-3:Noisy Data:
Noisy data is a meaningless data that can't be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

- **BinningMethod:** This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
- **Regression:** Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).
- **Clustering:** This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

Step-4:Data Transformation

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

- **Normalization:** It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
- **Attribute Selection:** In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
- **Discretization:** This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
- **Concept Hierarchy Generation:**Here attributes are converted from lower level to higher level in hierarchy.For Example-The attribute "city" can be converted to "country".

Step-5.Data Reduction:
Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.
The various steps to data reduction are:

**Data Cube Aggregation:** Aggregation operation is applied to data for the construction of the data cube.

**Attribute Subset Selection:** The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value

of the attribute.the attribute having p-value greater than significance level can be discarded.

**Numerosity Reduction:**This enable to store the model of data instead of whole data, for example: Regression Models.

**Dimensionality Reduction:**This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Component Analysis).

## 5.3 Data Augmentation

The prediction accuracy of the Supervised Deep Learning models is largely reliant on the amount and the diversity of data available during training. The relation between deep learning models and amount of training data required is analogous to that of the relation between rocket engines (deep learning models) and the huge amount of fuel (huge amounts of data) required for the rocket to complete its mission (success of the deep learning model).

DL models trained to achieve high performance on complex tasks generally have a large number of hidden neurons. As the number of hidden neurons increases, the number of trainable parameters also increases. The number of parameters in the State-of-the-art Computer Vision models such as RESNET (60M) and Inception-V3 (24M) is of the order of ten million.

In Natural Language Processing models such as BERT (340M), it is of the order of a hundred million. These deep learning models trained to perform complex tasks such as object detection or language translation with high accuracy have a large number of tunable parameters. They need a large amount of data to learn the values for a large number of parameters during the training phase.

In simple terms, the amount of data required is proportional to the number of learnable parameters in the model. The number of parameters is proportional to the complexity of the task.

Oftentimes, when working on specific complex tasks such as classifying a weed from a crop, or identifying the novelty of a patient, it is very hard to get large amounts of data required to train the models. Though transfer learning techniques could be used to great

effect, the challenges involved in making a pre-trained model to work for specific tasks are tough.

Another way to deal with the problem of limited data is to apply different transformations on the available data to synthesize new data. This approach of synthesizing new data from the available data is referred to as 'Data Augmentation'.

Data augmentation can be used to address both the requirements, the diversity of the training data, and the amount of data. Besides these two, augmented data can also be used to address the class imbalance problem in classification tasks.

The questions that come to the mind are; Does data augmentation work? Do we really get better performance from the models when we use augmented data for training? Table 1 shows a few case studies indicating the effect of data augmentation on model performance for different applications.

## 5.4 Training the model

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

This iterative process is called "model fitting". The accuracy of the training dataset or the validation dataset is critical for the precision of the model.

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning.

Supervised learning is possible when the training data contains both the input and output values. Each set of data that has the inputs and the expected output is called a supervisory signal. The training is done based on the deviation of the processed result from the documented result when the inputs are fed into the model.

Unsupervised learning involves determining patterns in the data. Additional data is then used to fit patterns or clusters. This is also an iterative process that improves the accuracy based on the correlation to the expected patterns or clusters. There is no reference output dataset in this method.

## 5.5 Google Colab

**Google Colaboratory** is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs.

Here's what I truly love about Colab. It does not matter which computer you have, what it's configuration is, and how ancient it might be. You can still use Google Colab! All you need is a Google account and a web browser. And here's the cherry on top – you get access to GPUs like **Tesla K80** and even a TPU, for free!

TPUs are much more expensive than a GPU, and you can use it for free on Colab. It's worth repeating again and again – it's an offering like no other.

Are you are still using that same old Jupyter notebook on your system for training models? Trust me, you're going to love Google Colab.

GPUs and TPUs on Google Colab

Ask anyone who uses Colab why they love it. The answer is unanimous – the availability of free GPUs and TPUs. Training models, especially deep learning ones, takes numerous hours on a CPU. We've all faced this issue on our local machines. GPUs and TPUs, on the other hand, can train these models in a matter of minutes or seconds.

Whether it is a Data science hackathon or a deep learning project, I always prefer a GPU over any other CPU because of the sheer computational power and speed of execution. But, not everyone can afford a GPU because they are expensive. That's where Google Colab comes into play.

It gives you a decent GPU for free, which you can continuously run for 12 hours. For most data science folks, this is sufficient to meet their computation needs. Especially if you are a beginner, then I would highly recommend you start using Google Colab.

Google Colab gives us three types of runtime for our notebooks:

- CPUs,
- GPUs, and
- TPUs

As I mentioned, Colab gives us 12 hours of continuous execution time. After that, the whole virtual machine is cleared and we have to start again**.** We can run multiple CPU, GPU, and TPU instances simultaneously, but our resources are shared between these instances.

## 5.6 CONVOULTIONAL NEURAL NETWORKS

CNN's are a special type of ANN which accepts images as inputs. Below is the representation of a basic neuron of an ANN which takes as input X vector. The values in the X vector is then multiplied by corresponding weights to form a linear combination. To thus, a non-linearity function or an activation function is imposed so as to get the final output.

### 5.6.1. Use of CNN

Talking about grayscale images, they have pixel ranges from 0 to 255 i.e. 8-bit pixel values. If the size of the image is NxM, then the size of the input vector will be N*M. For RGB images, it would be N*M*3. Consider an RGB image with size 30x30. This would require 2700 neurons. An RGB image of size 256x256 would require over 100000 neurons. ANN takes a vector of inputs and gives a product as a vector from another hidden layer that is fully connected to the input. The number of weights, parameters for 224x224x3 is very high. A single neuron in the output layer will have 224x224x3 weights coming into it. This would require more computation, memory, and data. CNN exploits the structure of images leading to a sparse connection between input and output neurons. Each layer performs convolution on CNN. CNN takes input as an image volume for the RGB image. Basically, an image is taken as an input and we apply kernel/filter on the image to get the output. CNN also enables parameter sharing between the output neurons which means that a feature detector (for example horizontal edge detector) that's useful in one part of the image is probably useful in another part of the image.

### 5.6.2. Convolutions

Every output neuron is connected to a small neighborhood in the input through a weight matrix also referred to as a kernel or a weight matrix. We can define multiple kernels for every convolution layer each giving rise to an output. Each filter is moved around the input image giving rise to a 2nd output. The outputs corresponding to each filter are stacked giving rise to an output volume.

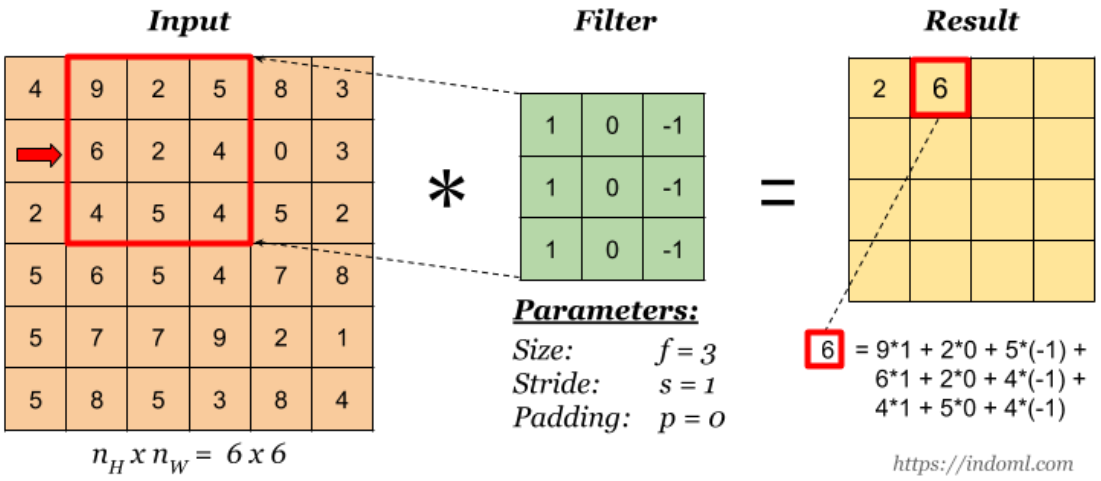


Fig 5.2 convolution

```
import numpy as np
import pandas as pd
import os
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator, load_img
from keras.layers import Conv2D, Dense, BatchNormalization, Activation, Dropout, MaxPooling2D, Flatten
from keras.optimizers import Adam, RMSprop, SGD
from keras import regularizers
from keras.callbacks import ModelCheckpoint, CSVLogger, TensorBoard, EarlyStopping, ReduceLROnPlateau
import datetime
import matplotlib.pyplot as plt
from keras.utils.vis_utils import plot_model
```

```
os.chdir("/content/drive/MyDrive/group project/FER")
os.getcwd()
```

5.3. Importing the libraries

```
train_dir = 'train/'
test_dir = 'test/'
row, col = 48, 48
classes = 7

def count_exp(path, set):
    dict = {}
    for expression in os.listdir(path):
        dir = path + expression
        dict[expression] = len(os.listdir(dir))
    df = pd.DataFrame(dict, index=[set])
    return df
train_count = count_exp(train_dir, 'train')
test_count = count_exp(test_dir, 'test')
print(train_count)
print(test_count)
print()
```

```
       angry  disgust  fear  happy  neutral   sad  surprise
train   3995      436  4097   7215     4965  4830      3171
      angry  disgust  fear  happy  neutral   sad  surprise
test    958      111  1024   1774     1233  1247       831
```

Fig 5.4 Data visualization

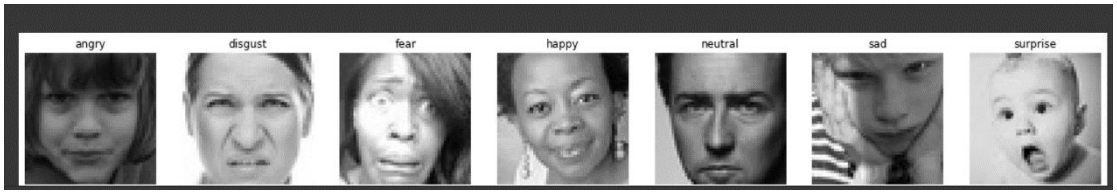


Fig 5.5 Sample emotions

```
train_datagen = ImageDataGenerator(rescale=1./255.0,
                                   zoom_range=0.3,
                                   horizontal_flip=True)

training_set = train_datagen.flow_from_directory(train_dir,
                                                 batch_size=64,
                                                 target_size=(48,48),
                                                 shuffle=True,
                                                 color_mode='grayscale',
                                                 class_mode='categorical')
### size of training_set ==  28709
test_datagen = ImageDataGenerator(rescale=1./255.0)
test_set = test_datagen.flow_from_directory(test_dir,
                                                 batch_size=64,
                                                 target_size=(48,48),
                                                 shuffle=True,
                                                 color_mode='grayscale',
                                                 class_mode='categorical')
```

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

Fig 5.6 Data Augumentation

```
def get_model(input_size, classes=7):
     #Initialising the CNN
    model = tf.keras.models.Sequential()

    model.add(Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', input_shape =input_size))
    model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(2, 2))
    model.add(Dropout(0.25))

    model.add(Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l2(0.01)))
    model.add(Conv2D(256, kernel_size=(3, 3), activation='relu', kernel_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(512, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=regularizers.l2(0.01)))
    model.add(Conv2D(512, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    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(classes, activation='softmax'))

    #Compliling the model
    model.compile(optimizer=Adam(lr=0.0001, decay=1e-6),
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

Fig 5.7 Model Architecture

# 6.OUTPUT AND APPLICATIONS

The proposed system recognizing the facial expressions that classifies the images into seven different emotions as happy, fear, angry, disgust, neutral, sad and surprise. The main advantage of the proposed system is, it has more accuracy. The training and test examples gain more accuracy.

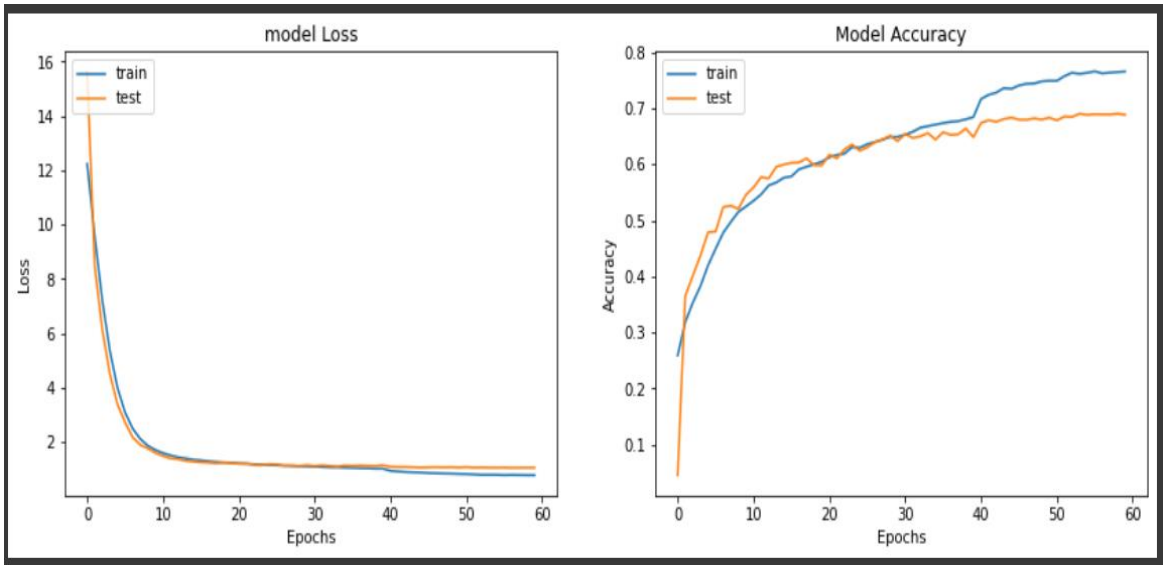


Fig 6.1 Result Curves

**APPLICATIONS:**

- Facial expression provides clues about emotion, intention, alertness, pain, personality, regulates interpersonal behavior, and communicates psychiatric and biomedical status among other functions
- Companies can use it for marketing, sending targeted ads to consumers. Law enforcement agencies use it to identify suspects or track down missing persons. And tech companies use it to allow consumers to easily unlock their devices.
- Facial Expression Recognition (FER) can be widely applied to various research areas, such as mental diseases diagnosis and human social/physiological interaction detection.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 48, 48, 32)        320
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 48, 48, 64)        18496
_________________________________________________________________
batch_normalization_11 (Batc (None, 48, 48, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 24, 24, 64)        0
_________________________________________________________________
dropout_10 (Dropout)         (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 24, 24, 128)       73856
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 22, 22, 256)       295168
_________________________________________________________________
batch_normalization_12 (Batc (None, 22, 22, 256)       1024
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 11, 11, 256)       0
_________________________________________________________________
dropout_11 (Dropout)         (None, 11, 11, 256)       0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 11, 11, 512)       1180160
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 11, 11, 512)       2359808
_________________________________________________________________
batch_normalization_13 (Batc (None, 11, 11, 512)       2048
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 5, 5, 512)         0
_________________________________________________________________
dropout_12 (Dropout)         (None, 5, 5, 512)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 12800)             0
_________________________________________________________________
dense_10 (Dense)             (None, 1024)              13108224
_________________________________________________________________
dropout_13 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_11 (Dense)             (None, 7)                 7175
=================================================================
Total params: 17,046,535
Trainable params: 17,044,871
Non-trainable params: 1,664
_________________________________________________________________
```

Fig 6.2 Model Testing

```
import os
os.chdir("/content/drive/MyDrive/group project/FER")
os.getcwd()
```

```
'/content/drive/MyDrive/group project/FER'
```

```
import tensorflow as tf
model = tf.keras.models.load_model("model.h5")
```

```
from google.colab import files
from io import BytesIO
from PIL import Image

uploaded = files.upload()
#im = Image.open(BytesIO(uploaded["Training_87867.jpg"]))
```

```
Choose Files No file chosen  Upload widget is only available when the cell
Saving Training_109676.jpg to Training_109676.jpg
```

```
import cv2
import matplotlib.pyplot as plt
im = cv2.imread("/content/drive/MyDrive/group project/FER/Training_109676.jpg")
im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
plt.imshow(im)
plt.show()
```

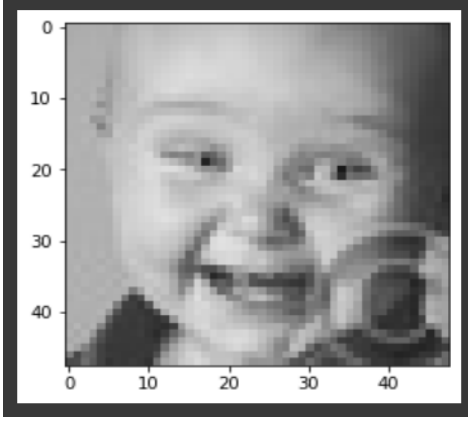


Fig 6.3 Uploading the model

```
def preprocess(filepath):
  IMG_SIZE = 48  # 50 in txt-based
  img_array = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)  # read in the image, convert to grayscale
  new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))  # resize image to match model's expected sizing
  return new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 1)
filepath = "/content/drive/MyDrive/group project/FER/Training_123362.jpg"
prediction = model.predict([preprocess(filepath)])
print(prediction)
classes = np.argmax(prediction, axis = 1)
cat = ["angry","disgust","fear","happy","neutral","sad","surprise"]
print()
print(cat[classes[0]])
print()
im = cv2.imread("/content/drive/MyDrive/group project/FER/Training_123362.jpg")
im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
plt.imshow(im)
plt.show()
```

```
[[0. 0. 0. 1. 0. 0. 0.]]
```

Fig 6.4 Preprocessing the uploaded images

```
[[0. 0. 0. 1. 0. 0. 0.]]

happy
```

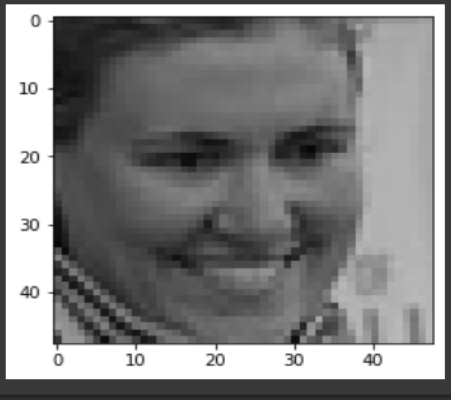


Fig 6.5 Mapping the results

```
train_loss, train_accu = model.evaluate(training_set)
test_loss, test_accu = model.evaluate(test_set)
print("final train accuracy = {:.2f} , validation accuracy = {:.2f}".format(train_accu*100, test_accu*100))
```

```
449/449 [==============================] - 39s 86ms/step - loss: 0.6576 - accuracy: 0.8233
113/113 [==============================] - 8s 70ms/step - loss: 1.0590 - accuracy: 0.6884
final train accuracy = 82.33 , validation accuracy = 68.84
```

```
model.save_weights('model_bestweight.h5')
```

Fig 6.6 Model accuracy

# 7.CONCLUSION

- We developed various CNNs for a facial expression recognition problem and evaluated their performances using different pre and post-processing and visualization techniques.
- The results demonstrated that deep CNNs are capable of learning facial characteristics and improving facial emotion detection.
- Hence, we developed a model that can classify the basic emotion with image

# 8.FUTURE SCOPE

- It is not possible to make a complete 100 percent accurate model but the present model can be enhanced with other different techniques.
- Facial expression recognition can be used to detect the pain of the patient in case of emergency.
- It can be used to express the feelings of dumb people.
- We can use Haar Cascade classifier or DNN to detect the face in the image and crop the face part that useful for expression recognition.
- The cropped image is well preprocessed inorder to classify it’s emotion.
- Our proposed model will classify the image and gives output as the name of emotion.
- We can use opencv library to get live web stream data to be processed and in this way we can classify emotion using webcam too.

# 9.REFERENCES


[1] Zhiding Yu and Cha Zhang, “Image based Static Facial Expression Recognition with Multiple Deep Network Learning”, ICMI '15 Proceedings of the ACM on International Conference on Multimodal Interaction, 2015, pp. 435-442.

[2] S. M. Lajevardi and M. Lech, “Facial expression recognition from image sequences using optimized feature selection,” in Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference, IEEE, 2008, pp. 1–6.

[3] P. Viola and M. J. Jones,“Robust real-time face detection,” International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[4] Pushpaja V. Saudagare, D.S. Chaudhari, “Facial Expression Recognition using Neural Network –An Overview,” International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 1, 2012

[5] Jung, Heechul & Lee, Sihaeng & Park, Sung Il & Kim, Byungju & Kim, Junmo & Lee, Injae & Ahn, Chunghyun, “Development of deep learning-based facial expression recognition system,” 2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV), Mokpo, 2015, pp. 1- 4.

[6] M. Xiaoxi, L. Weisi, H. Dongyan, D. Minghui and H. Li, "Facial emotion recognition," 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 2017, pp. 77-81.

[7] Neha Jain, Shishir Kumar, Amit Kumar, Pourya Shamsolmoali, Masoumeh Zareapoor, “Hybrid deep neural networks for face emotion recognition,” Pattern Recognition Letters, vol. 115, 2018, pp. 101-106.

[8] A.Fathallah, L. Abdi and A. Douik, "Facial Expression Recognition via Deep Learning," 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, 2017, pp. 745-750.

1. https://www.kaggle.com/msambare/fer2013
2. https://www.sciencedirect.com/science/article/pii/S1319157818303379
3. https://docs.streamlit.io/en/stable/
4. https://www.youtube.com/results?search_query=data+professor
5. https://www.youtube.com/results?search_query=krish+naik
6. https://arxiv.org/ftp/arxiv/papers/1005/1005.4263.pdf
7. https://www.ijana.in/Special%20Issue/C28.pdf
8. https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/
9. https://algorithmia.com/blog/introduction-to-emotion-recognition#:~:text=Facial%20emotion%20recognition%20is%20the,can%20recognize%20emotions%20as%20well