IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 9, September 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

Facial Emotional Recognition Using Deep Convolutional Neural Networks

Hemish Veeraboina[1,*], Surapur Sai Teja[2], Y Sai Sameer[3], Mavuluri Vamsi Krishna Reddy[4], Mohammed Danish Hussain[5]

Department of Computer Science & Engineering, Maturi Venkata Subba Rao Engineering College, affiliated to Osmania University, India[1,2,3,4]

Department of Electronics and Communications Engineering, Maturi Venkata Subba Rao Engineering College, affiliated to Osmania University, India[5]

ABSTRACT: A face reveals a lot of information about a person's identity, age, sex, race, and emotional as well as psychological state. Facial expressions are often used in the behavioral interpretation of emotions and play a key role in social interactions. Due to its potential applications such as HCI, behavioral science, automatic facial emotion detection is one of the most intriguing and challenging areas in computer vision.

Our Facial Emotion Recognition system performs detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification.

Numerous models were experimented such as decision trees and SVM's before arriving at a Convolutional Neural Networks (CNN) model. Due to large number of filters, CNNs are efficient for image recognition tasks since they can capture spatial features of the inputs. After observing unbelievable performance with deep learning models, we used deep convolution features to better represent the given image instead of using the traditional handcrafted features.

We have proposed and developed a Deep Convolutional Neural Network for classifying human emotions from dynamic facial expressions in real-time. Kaggle's FER-2013 dataset with seven facial emotions labels as happy, sad, surprise, fear, anger, disgust, and neutral is used to train the model. An overall training accuracy of 92.86% and test accuracy of 64.42% are achieved. Finally, a live video stream connected to a face detector feeds images to the neural network. The network subsequently classifies an arbitrary number of faces per image in real-time simutaneously, wherein appropriate emotions are displayed over the subject's faces and their probabilities are displayed over a real-time bar graph on a separate window.

KEYWORDS: Facial Expression Recognition, Convolutional Neural Network, Deep Learning, Facial Action Coding System.

I. INTRODUCTION

A person's affective state, cognitive activity, intention, personality, and psychopathology are observable manifestations of their facial expression, which plays a huge role in interpersonal relationships. It has been researched for many years and has made significant development in recent decades. Generally, human beings can convey expressions and emotions through non-verbal ways such as gestures, facial expressions, and involuntary language. The important thing is how efficiently the system detects or extracts the facial expression from an image. The important thing is how well the system detects or extracts the facial emotions from a given image. The technology is gaining popularity because it has the potential to be applied in numerous domains, including lie detection, medical assessment, and HCI. Facial Action Coding System (FACS), introduced by Ekman in 1978 and modified in 2002, is a widely used technique for analyzing facial expressions.

Humans perceive emotions on a daily basis through characteristic features expressed as part of a facial expression. Happiness, for example, is inextricably linked to a smile or upward movement of the lips' corners. Similarly, other emotions are defined by deformations that are unique to that expression.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

The approach divides a person's facial expression into basic emotions (disgust, anger, happiness, fear,sadness, and surprise). The major goal of this system is to enable efficient interaction between humans and machines through the use of eye gazing, facial expressions, cognitive modeling, and other techniques.

As the intensity of the system differs from person to person, it also depends on the change in features such as age, gender, facial size, and shape. And even within the same person, reactions do not remain consistent over time. However, due to the intrinsic variety of facial images caused by various aspects such as lighting, position, alignment, and occlusions, recognizing expressions is a complicated process [1]. Some surveys on facial feature representations for face recognition and expression analysis addressed these challenges and proposed appropriate solutions in detail.

1.1 PROBLEM STATEMENT

Human expressions and emotions are expressed through facial expressions andfacial expression system's key component is to determine an effective and efficientfeature[2].Face recognition is critical for interpreting facial expressions in applications including man-machine interfaces and communication between them, real-time animation from live motion images, intelligent visual surveillance, and teleconferencing. The facial expressions are useful for efficient interaction Most research and system in facial expression recognition are limited to seven basic expressions wherein each number represents the respective emotion i.e emotions such as anger, disgust, fear, happy, sad, surprise and neutral are represented by 0,1,2,3,4,5,6 respectively. It isn't possible to describe all facial emotions and expressions by categorizing them into these categories [2]. Detecting the face and recognizing the facial expression is a very complicated task when it is vital to pay attention to primary components like face configuration, orientation, the location where the face is set [4]

1.2 OBJECTIVES

- Developing robust computer vision techniques for the analysis and recognition of facial expressions from Real-Time video feed.
- To experiment with machine learning algorithms in computer vision fields.
- TodetectemotionthusfacilitatingIntelligentHuman-ComputerInteraction
- Improving and optimizing both feature extraction and selection methods for facial expression recognition.
- To overcome some of the disadvantages and limitations of existing facial expression recognition methods.

1.3 SCOPE

The scope of this system is to tackle the problems that can arise in day-to-day life. Some of the scopes are:

- The system can be used to detect and track a user's state of mind.
- The system can be used in mini-marts, shopping malls to view the feedback of the customers to enhance the business.
- The system can be installed at busy places like airports, railway stations, or bus stations for detecting human faces and facial expressions of each person. If any faces appeared suspicious like angry or fearful, the system might set an internal alarm.
- The system can also be used for educational purposes such as one can get feedback on how the student is reacting during the class.
- During interrogation, this system can be used to detect lies amongst criminal suspects
- Clever marketing is feasible using the emotional knowledge of a person which can be identified by this system.
- This system can help people in emotion-related research to improve the processing of emotional data [5]

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

1.4 PROPOSED SYSTEM

This paper aims to propose and develop a CNN for classifying facial emotions from a live-video feed in real time. The results obtained from this proposed approach demonstrates the feasibility of implementing this neural network to detect facial emotions in real-time.

II. LITERATURE REVIEW

Processing the facial image as a "whole" or dividing the facial image into separate action units appears to be the primary distinction between the two approaches.

2.1 EXISTING SYSTEM

The proposed approach makes use of the whole face image and performs processing, finally narrowing it down to end up with the classifications of 7 universal facial expressions.

It is assumed that each of the above-mentioned emotions has a unique characteristic associated with it that's why recognition is necessary and sufficient. The existing system operates on Geometric Based Parameterization, which is a traditional way that involves tracking and processing the motions of some spots on images and video sequences. In the existing approach, the extraction in the change of expressions or motion of features are approximated using Bezier Curves [6]

The manual adjustments of contours of the features and components have a huge setback in the existing approach. Difficulties arise in terms of robustness, changes in poses and illumination when tracking is applied is predominant. Given that actions and expressions change both in morphological and dynamical senses, it is complicated to estimate these general parameters for movement and displacement. Thus, ending up with strong decisions for facial expressions can be difficult under these varying conditions.

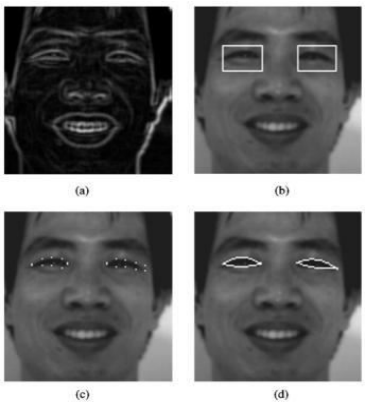

Fig. 2.1 Eye extraction

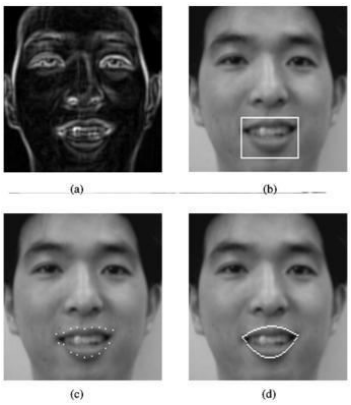

Fig. 2.2 Mouth extraction

Fig. 2.3Face Data (Normal and Happy)

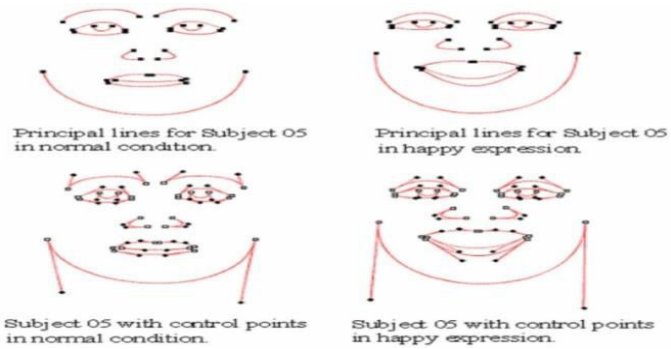

Fig 2.4 Analyzing Control points

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

2.2 PROPOSED SYSTEM

Dividing the facial image into sub-sections for processing forms up the main idea of the proposed approach, rather than using the whole image. Subtle changes in discrete features such as eyes, lips, eyebrows could result in change in emotions; these changes are used to analyze facial emotion. Pixel information in appropriate regions of the face are analyzed and processed in appearance-based parameterizations rather than tracking spatial points and using positional parameters that change by time.

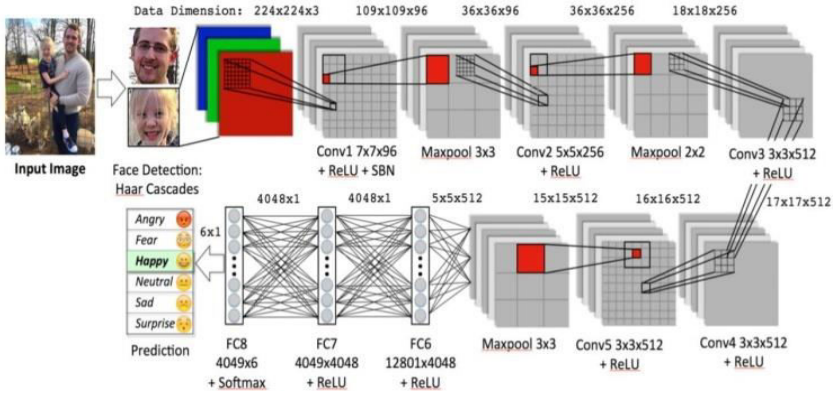

Figure2.5.The architecture of the convolution neural network for Facial Emotional Recognition
Input images and cropped faces are shown here. The images are cropped using Haar-Cascade detector.

2.3 NEURAL NETWORKS

A neural network is defined as a system of interconnected artificial "neurons" which communicate by exchanging messages to each other. The numeric weights between connections are tuned in such a way that a trained network will respond precisely whenan image or pattern is given as an input to recognize. The network consists of numerous layers of "neurons" which exhibit feature-detecting capabilities. Every neuron is trained to respond differently for every input depending on the given the information about present layer an previous layer. In figure 2.6, the layers are built in such a way that the initial layer detects a set of primitive patterns; the second layer detects the patterns associated within those patterns and so on for further layers. Typical convolution neural networks use five to twenty five distinct layers of pattern recognition [7][8]

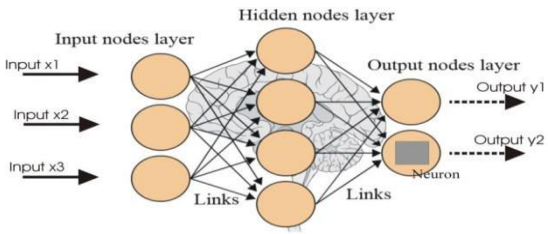

Fig. 2.6 Artificial Neural Network

Neural networks are inspiration taken from biological neural systems. Given that the basic computational unit of the brain is a neuron and they are connected with synapses, neural networks mimics the operation of the brain in a similar way.

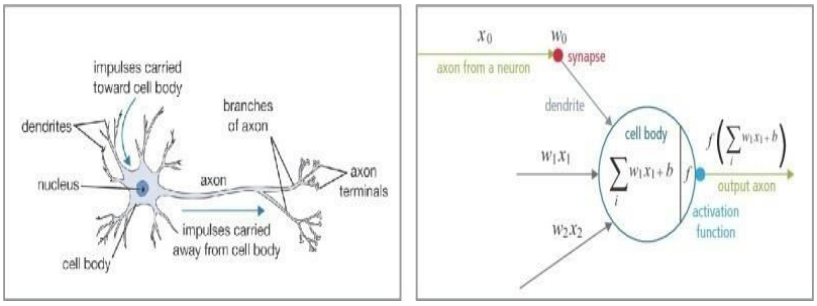

Fig. 2.7 Illustration of a biological neuron model (left) and mathematical model (right)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

2.3.1 Convolutional Neural Network

A convolutional neural network (CNN) is a type of artificial neural network used in processing and image recognition that is specifically designed to process pixel data. Convolution is a mathematical operation on two functions which produce a third function. Convolution neural networks process and ingest images as tensors, and tensors act as the matrices of numbers with additional dimensions [9]

2.3.1.1 Stride

Stride is a component of CNN's or neural networks which is extensively used for compression of video data and images. This parameter's filter is responsible for the modification of the amount of movement over the image or a video sequence. I.e. the number of pixels the window has to moves across.

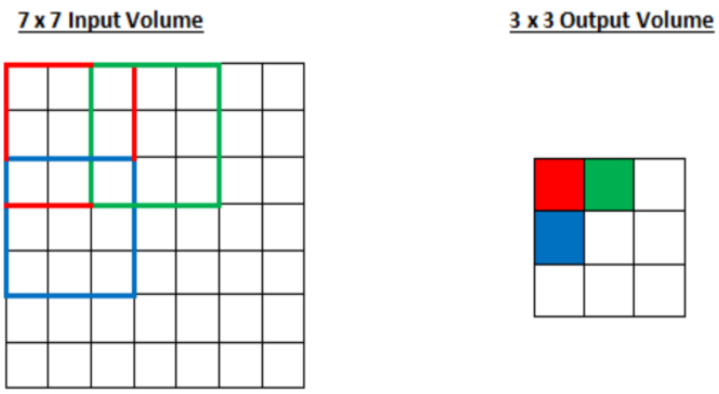

Fig. 2.8 Illustration of Stride

2.3.1.2 Padding

Figure 2.9 displays the image about padding and it is a parameter that works in conjunction with Stride. To allow minimized reduction of size in the output layer, padding operation performs addition of blank or empty pixels to the frame of the image. To counteract the fact that stride reduces the size, padding increases the size of the image. Stride and padding are the foundational parameters of any convolution neural network.

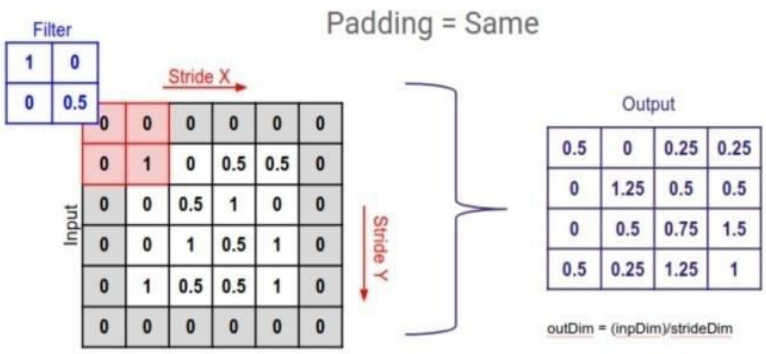

Fig. 2.9 Illustration of padding

2.3.1.3 Pooling

Figure 2.10 displays the pooling illustration and it is the condensation of a feature map. The pooling layer is added after the convolution layer. Dimensionality reduction is performed i.e. reduction in number of parameters and computations thereby controllingover fitting and shortening training time. The pooling technique used in this paper is "**max-**

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

pooling", which decreases the size of the feature map by taking maximum value from each window while keeping the essential information.

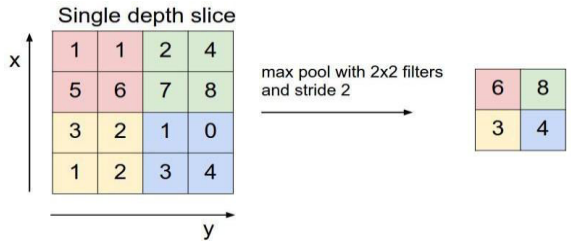

Fig. 2.10 Illustration of Max-Pooling

2.3.1.4 Activation Functions

An **activation function** is a simple mapping of summed weighted input tothe output of the neuron. It mimics the stimulation of a biological neuron. Activation functions are beneficial because they add nonlinearities into neural networks, allowing neural networks to learn powerful operations.

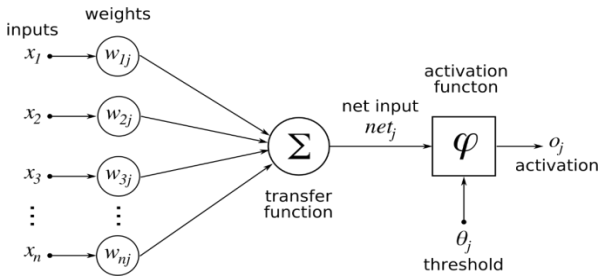

Fig. 2.11Activation Function

2.3.1.4.1 ReLU (Rectified Linear Unit) function

The function y = max(x,0) is implemented by ReLU, which converges input and output sizes to be the same. Non-linear properties of the decision function and the overall network are increased without affecting the receptive fields of the convolution layer. In comparison to other non-linear functions used in CNNs (e.x, sigmoid, hyperbolic tangent and absolute of hyperbolic tangent), the benefit of a ReLU is that the frequency of the network is enhanced. ReLU functionality is illustrated in Figure 2 [10].

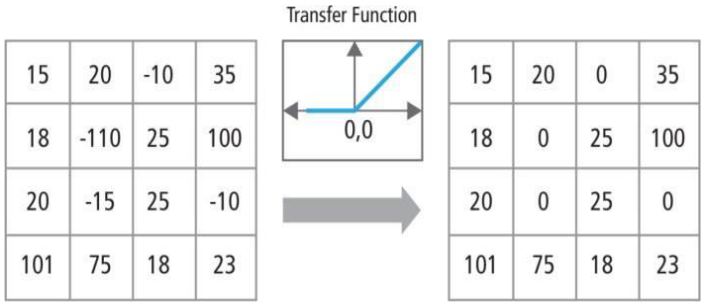

Fig. 2.12 ReLU functionality

2.3.1.5 Call Back Functions

Callback functions are those functions that are called at the end of every epoch and in this model; we use two callback functions – Reduce LROn Plateau and Early Stopping.

a. **Reduce LROn Plateau:** This function monitors a certain variable, in this case, validation loss, and alters the learning rate when the value stops significantly changing after some certain number of epochs.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

b. **Early Stopping:** At times our optimizer can land inlocal optima and get stuck there. There is no point in continuing the training as there won't be any further improvements.

2.3.1.6 Model Check pointing

This method saves the best version of the model along with its weights so that incase any crash occurs, the model can be recovered.

2.4 DATA SET DESCRIPTION

The dataset contains 35,685 examples of 48x48 pixel grayscale images offaces. The faces are automatically registered in such a way that the face is more or less centered and occupies about the same amount of space in each image. Facial emotions are characterized into any one of these emotions.

The dataset has two columns, "emotion" and "pixels". The "emotion"column contains a number code ranging from 0 to 7, exclusive, representing each emotion of the image.(0 is Angry, 1 is Disgust, 2 is Fear, 3 is Happy, 4 is Sad, 5 is Surprise,6 is Neutral). The "pixels" column contains a string which contains space-separated pixel values in row-major order, surrounded by quotes for each image.

The dataset is prepared by Pierre-Luc Carrier and Aaron Courville.

Table 2.1 maps emotion with the respective emotion label and several images for that particular emotion.

EmotionLabel	NumberofImages	Emotion
0	4593	Angry
1	547	Disgust
2	5121	Fear
3	8989	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

Table 2.1Description of FER dataset

Table 2.2 Sample images of FER dataset

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

III. SYSTEM DESIGN

3.1 DATAFLOW DIAGRAM

A data-flow diagram is a representation of flow of the data of a process or a system. Figure 3.1 is the DFD for the system proposed where a series of events occur in a flow. Firstly, data acquisition takes place and it is pre-processed.

Based on the relationship between variables, various models are selected. The data is then split into training and testing set which is passed through a prediction model. Finally, the accuracy of the model and prediction is obtained.

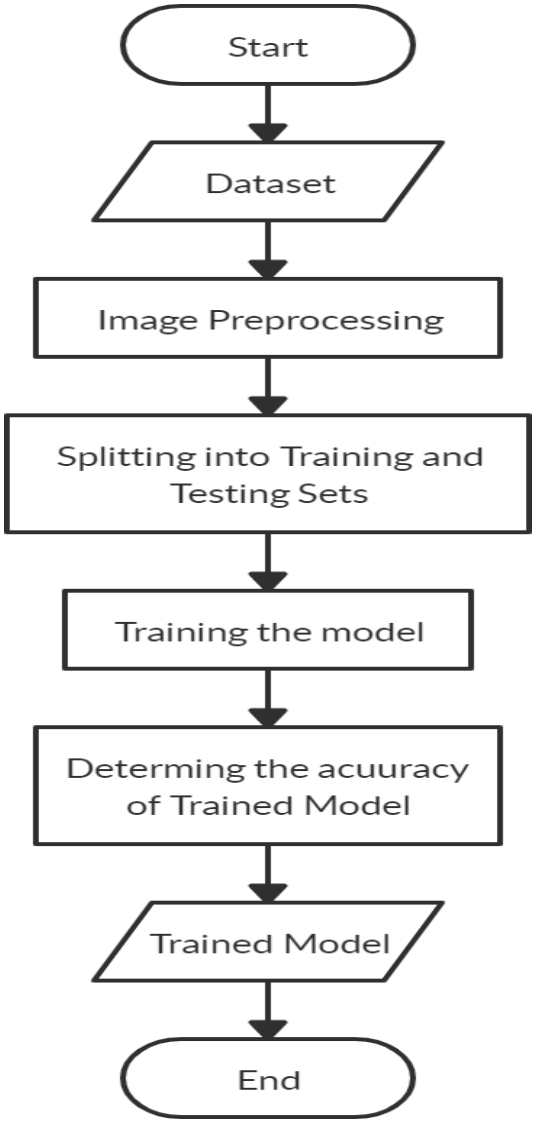

Fig.3.1 Data Flow Diagram of Training Phase

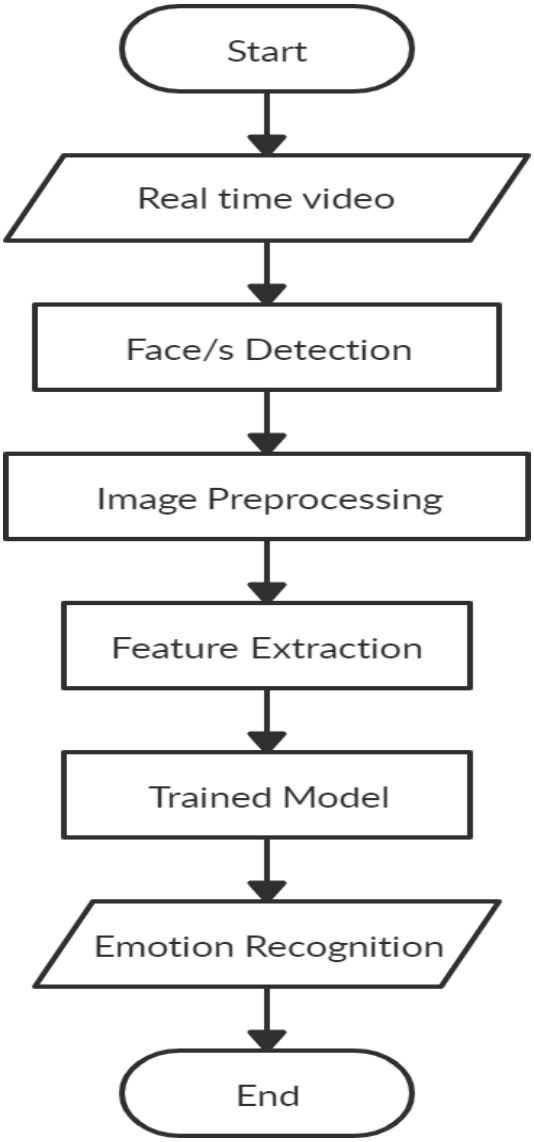

Fig.3.2 Data Flow Diagram of FER executable file

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

3.2 USECASEDIAGRAM

A Use Case Diagram is a simple representation of a user's interactionwith the system that depicts the relationship between the user and different use cases the user is involved in.
Different users who interacted with our project are Software System, Data Source,and User which are related to different use cases. Figure 3.2 shows the use case diagram of our proposed system.

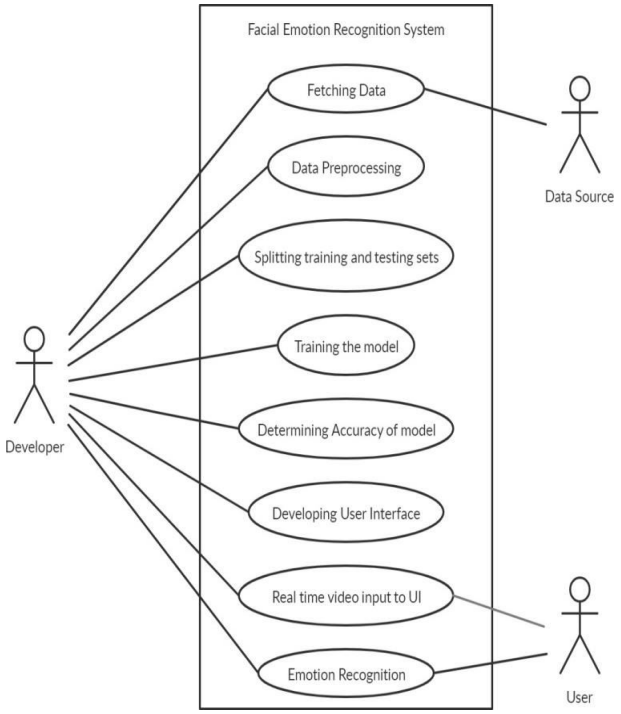

Fig. 3.3Use Case Diagram

3.3 SEQUENCEDIAGRAM

The sequence diagram shows object interactions arranged in a time sequence. It displays the numerous classes and objects involved in the scenario and the sequence in which communication is exchanged between them, which are essential to carry out the functionality required in the scenario. Sequence diagrams are typically associated with use case realizations in the logical aspect of the system which is under development.

The parallel vertical lines (lifelines) display different processes or objects that live simultaneously, and the horizontal arrows represent the messages exchanged between them. This simple runtime scenario is depicted using use case diagram in a graphical manner.

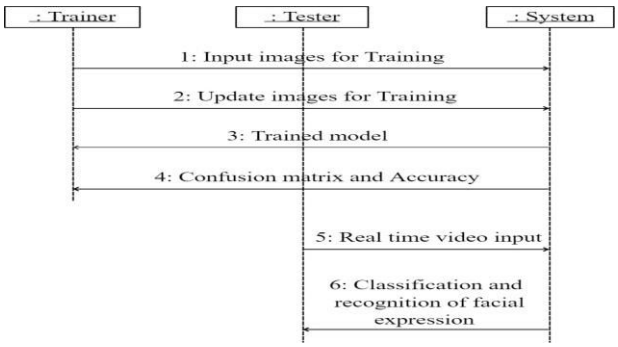

Fig 3.4 Sequence Diagram

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

IV. IMPLEMENTATION

4.1. PREPROCESSING

- Importing all the libraries required and reading the respective CSV file.
- Obtain training features x and labels y from pixels and columns featuring emotions in the CSV and converting them into NumPy arrays.
- Dimension can be increased to this feature vector by using the function np.expand_dims(). This makes it useful to the implementation of our CNN. After we execute the code above, the output is –

Pre-processing Done
Features: 2304(48x48) Number of Labels: 7
Images in the dataset: 35887

4.2. MODEL DEVELOPMENT

- Import necessary libraries for the CNN Implementation
- After declaring required variables, processing will be done in a batch size of 64. The pixel resolution is 48 x 48.
- Next, features are loaded into labels x and y respectively and standardize x by subtracting the means and dividing by the standard deviation.
- Division of data is performed using the function train_test_spilt(). Training data is further sub-divided for validation.

After we execute the code above, the output is –
Number of images in the Training set: 32298
Number of images in Test set: 3589

4.3. DESIGNINGTHECNN:

In this step,we create the CNN through which we will pass our features to train the model and then test it using the test features which is the most critical aspect of the entire process. To construct CNN, we used a combination of several different functions, which we will go through one by one.

a. Sequential():As we progress from the input layer to the output layer, a sequential model is just a linear stack of layers with layers stacked on top of each other.
b. **model.add(Conv2D())** - This is a 2D Convolution layer that performs the convolution process pointed out earlier . “This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs,” according to the Keras documentation. The kernel size is 3x3 and the activation function used is Rectified Linear Unit (ReLU).
c. **model.add(BatchNormalization())** - It performs batch normalization on inputs to the next layer so that our inputs are in a particular scale, such as 0 to 1, rather than being scattered throughout.
d. **model.add(MaxPooling2D())** - This function executes the data pooling procedure outlined in the Literature Review. In this model, we're using a 2x2 pooling window with 2x2 strides.
e. **model.add(Dropout())** - Dropout is a training technique where a sample of neurons are ignored during the training. They are "dropped out" at random. Overfitting is reduced as a result.
f. **model.add(Flatten())** - This simply converts the input from ND to 1D and has no effect on the batch size.
g. **model.add(Dense())** - Dense implements the operation: output = activation(dot(input, kernel), where element-wise activation function applied as the activation argument, and kernel is a weights matrix generated by the layer, according to the Keras documentation. In simple terms, it's the cherry on top, as it applies the features learned from the layers to the label.

The summary representation of the model is printed once the model.summary() method is called.[10]

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

Code Snippet for Training neural network

```
epochs = 100
batch_size = 64
learning_rate = 0.001


model = Sequential()

model.add(Conv2D(64, (3, 3), activation='relu', input_shape=(48, 48, 1),
          kernel_regularizer=l2(0.01)))
model.add(Conv2D(64, (3, 3), padding='same',activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(256, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(256, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(256, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(512, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(512, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(512, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))
adam = optimizers.Adam(lr = learning_rate)
model.compile(optimizer = adam, loss = 'categorical_crossentropy',
              metrics = ['accuracy'])
```

4.4. FACIAL EMOTION RECOGNITION DEMONSTRATION FILE

This file is responsible to display the emotion/s from real-time
Are handle real-time video input, face detection using Haar cascade filter, pre-processing of the images, and successively displaying the results on the screen.

a. **model_to_json()-** This method is defined in keras. models. This method is responsible to load our saved CNN model.
b. **load_weights ()-** This method is responsible to load all the weights from the saved CNN model.
c. **cv2.CascadeClassifier-** This method groups the features of the selected window by applying stages of classifiers one-by-one. A window is considered as a pass if it qualifies all stages as a face region.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

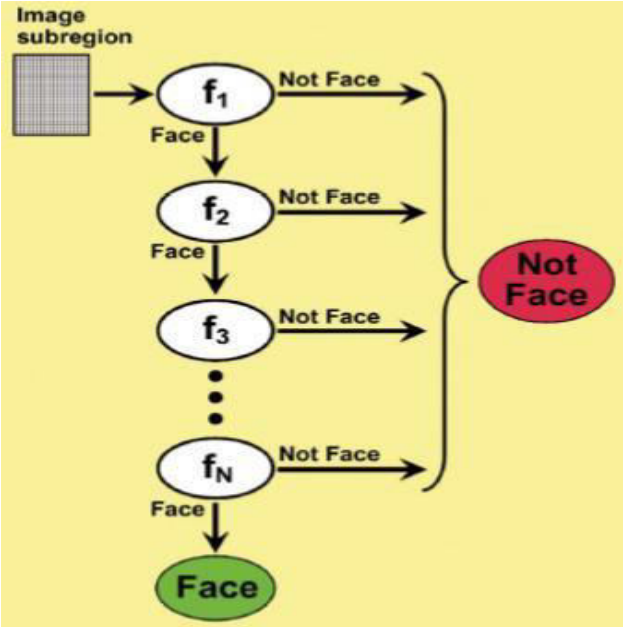

Fig. 4.1Haar Cascade Classifier

d. **cv2.videoCapture()-V**ideo capturing from image sequences, cameras, video files is enabled by this procedure. A C++ API is responsible for capturing video from image sequences or live-feed. This method is responsible to work with the camera and perform the following procedures: Read video, display video, save the video, capture and display the content from camera.

e. **cv2.cvtColor ()-**This method converts an image from one color space to another. There are more than 150 color-space conversion methods available in Open CV.
For this project, we convert the given images to grayscale.

f. **model.predict()-** Given a trained model, predict the label of a new set of data. This method accepts one argument and returns the learned label for each object tin the array.

g. **cv2.imshow ()-**This method is used to display an image in a window.

h. **cv2.resize ()-** Resizing an image means changing the dimensions of it, be it width alone, height alone or both can be performed using this method.

i. **cv2.putText ()-**This method disused to draw a text string on any image.

j. **Destroy All Windows()-** This method is responsible to de-allocate any unnecessary memory usage and close the window.

k. **cap.release()-**This method is used to release software resources and release hard ware resources.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

V. RESULTS AND OBSERVATIONS

5.1 TRAININGMODEL

5.1.1 Summary of Model

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 46, 46, 64)	640
conv2d_2 (Conv2D)	(None, 46, 46, 64)	36928
batch_normalization_1 (Batch	(None, 46, 46, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 64)	0
dropout_1 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 23, 23, 128)	73856
batch_normalization_2 (Batch	(None, 23, 23, 128)	512
conv2d_4 (Conv2D)	(None, 23, 23, 128)	147584
batch_normalization_3 (Batch	(None, 23, 23, 128)	512
conv2d_5 (Conv2D)	(None, 23, 23, 128)	147584
batch_normalization_4 (Batch	(None, 23, 23, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 11, 11, 128)	0
dropout_2 (Dropout)	(None, 11, 11, 128)	0
conv2d_6 (Conv2D)	(None, 11, 11, 256)	295168
batch_normalization_5 (Batch	(None, 11, 11, 256)	1024
conv2d_7 (Conv2D)	(None, 11, 11, 256)	590080
batch_normalization_6 (Batch	(None, 11, 11, 256)	1024
conv2d_8 (Conv2D)	(None, 11, 11, 256)	590080
batch_normalization_7 (Batch	(None, 11, 11, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 256)	0
dropout_3 (Dropout)	(None, 5, 5, 256)	0
conv2d_9 (Conv2D)	(None, 5, 5, 512)	1180160
batch_normalization_8 (Batch	(None, 5, 5, 512)	2048
conv2d_10 (Conv2D)	(None, 5, 5, 512)	2359808
batch_normalization_9 (Batch	(None, 5, 5, 512)	2048
conv2d_11 (Conv2D)	(None, 5, 5, 512)	2359808
batch_normalization_10 (Batc	(None, 5, 5, 512)	2048
max_pooling2d_4 (MaxPooling2	(None, 2, 2, 512)	0
dropout_4 (Dropout)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_6 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dropout_7 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
dropout_8 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 7)	455

Total params: 9,014,727
Trainable params: 9,009,223
Non-trainable params: 5,504

Table 5.1 Summary of Trained model

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

5.1.2 Training Cycles (Epochs)

The following results contain epochs numbered from 1-16 and 91-100.

```
Train on 25838 samples, validate on 6460 samples
Epoch 1/100
25838/25838 [==============================] - 33s 1ms/step - loss: 2.0946 - accuracy: 0.2001 - val_
loss: 1.8647 - val_accuracy: 0.2489

Epoch 00001: val_loss improved from inf to 1.86469, saving model to /content/gdrive/My Drive/Colab N
otebooks/fer/weights.hd5
Epoch 2/100
25838/25838 [==============================] - 24s 936us/step - loss: 1.8626 - accuracy: 0.2379 - va
l_loss: 1.8272 - val_accuracy: 0.2489

Epoch 00002: val_loss improved from 1.86469 to 1.82716, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 3/100
25838/25838 [==============================] - 24s 936us/step - loss: 1.8416 - accuracy: 0.2470 - va
l_loss: 1.8220 - val_accuracy: 0.2489

Epoch 00003: val_loss improved from 1.82716 to 1.82199, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 4/100
25838/25838 [==============================] - 24s 935us/step - loss: 1.8302 - accuracy: 0.2507 - va
l_loss: 1.8203 - val_accuracy: 0.2489

Epoch 00004: val_loss improved from 1.82199 to 1.82032, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 5/100
25838/25838 [==============================] - 24s 934us/step - loss: 1.8240 - accuracy: 0.2500 - va
l_loss: 1.8126 - val_accuracy: 0.2489

Epoch 00005: val_loss improved from 1.82032 to 1.81256, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 6/100
25838/25838 [==============================] - 24s 937us/step - loss: 1.8207 - accuracy: 0.2506 - va
l_loss: 1.8097 - val_accuracy: 0.2489

Epoch 00006: val_loss improved from 1.81256 to 1.80968, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 7/100
25838/25838 [==============================] - 24s 936us/step - loss: 1.8124 - accuracy: 0.2533 - va
l_loss: 1.8066 - val_accuracy: 0.2488

Epoch 00007: val_loss improved from 1.80968 to 1.80657, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 8/100
25838/25838 [==============================] - 24s 932us/step - loss: 1.8055 - accuracy: 0.2526 - va
l_loss: 1.7957 - val_accuracy: 0.2503

Epoch 00008: val_loss improved from 1.80657 to 1.79573, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 9/100
25838/25838 [==============================] - 24s 933us/step - loss: 1.7807 - accuracy: 0.2712 - va
l_loss: 2.2637 - val_accuracy: 0.1412

Epoch 00009: val_loss did not improve from 1.79573
Epoch 10/100
25838/25838 [==============================] - 24s 934us/step - loss: 1.7431 - accuracy: 0.2882 - va
l_loss: 1.8105 - val_accuracy: 0.2611

Epoch 00010: val_loss did not improve from 1.79573
Epoch 11/100
25838/25838 [==============================] - 24s 934us/step - loss: 1.7062 - accuracy: 0.3030 - va
l_loss: 1.8164 - val_accuracy: 0.2590

Epoch 00011: val_loss did not improve from 1.79573
Epoch 12/100
25838/25838 [==============================] - 24s 934us/step - loss: 1.6473 - accuracy: 0.3201 - va
l_loss: 1.7473 - val_accuracy: 0.2757

Epoch 00012: val_loss improved from 1.79573 to 1.74728, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 13/100
25838/25838 [==============================] - 24s 936us/step - loss: 1.6138 - accuracy: 0.3450 - va
l_loss: 1.6562 - val_accuracy: 0.3466

Epoch 00013: val_loss improved from 1.74728 to 1.65622, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 14/100
25838/25838 [==============================] - 24s 938us/step - loss: 1.5559 - accuracy: 0.3848 - va
l_loss: 1.5946 - val_accuracy: 0.3964

Epoch 00014: val_loss improved from 1.65622 to 1.59458, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 15/100
25838/25838 [==============================] - 24s 937us/step - loss: 1.5040 - accuracy: 0.4044 - va
l_loss: 1.4816 - val_accuracy: 0.4217

Epoch 00015: val_loss improved from 1.59458 to 1.48163, saving model to /content/gdrive/My Drive/Col
ab Notebooks/fer/weights.hd5
Epoch 16/100
25838/25838 [==============================] - 24s 939us/step - loss: 1.4721 - accuracy: 0.4128 - va
l_loss: 1.5639 - val_accuracy: 0.3887

Epoch 00016: val_loss did not improve from 1.48163
```

Fig. 5.2 Model Loss

For training, a dataset comprised of 32298 images is provided for the model, from which 20% is again split for validation. Out of the 25838 images used for training the model, 6460 images are used for cross-validation. The above outputs describe training accuracy of 92.86% and validation accuracy of 64.66%. Complete output can be viewed in the conclusion.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

5.1.3 Model Accuracy

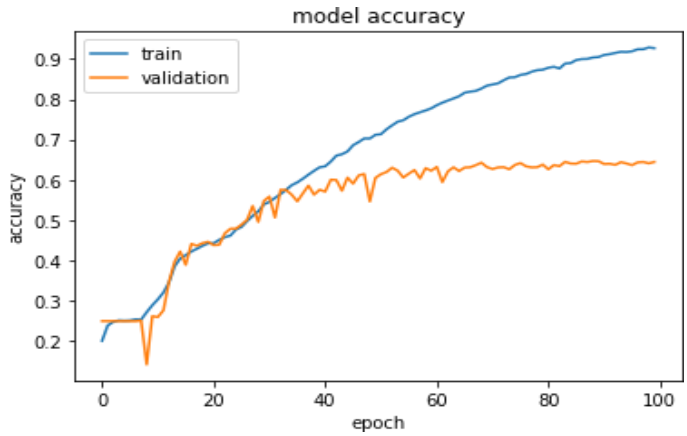

Fig. 5.1ModelAccuracy

5.1.4 Model Loss

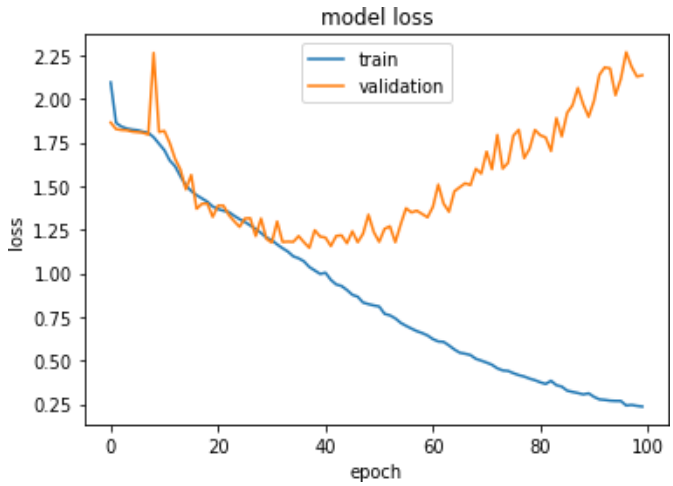

Fig 5.2 Model Loss

5.1.5 Accuracy

```
predicted_test_labels = np.argmax(model.predict(test_data), axis=1)
test_labels = np.argmax(test_labels, axis=1)
print ("Accuracy score = ", accuracy_score(test_labels, predicted_test_labels)*100)
```

```
Accuracy score =  64.41905823349123
```

```
print("Model accuracy  = ",max(model_info.history['accuracy'])*100)
print("Validation accuracy  = ",max(model_info.history['val_accuracy'])*100)
```

```
Model accuracy  =  92.85935163497925
Validation accuracy  =  64.65944051742554
```

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

5.1.6 Confusion Matrix

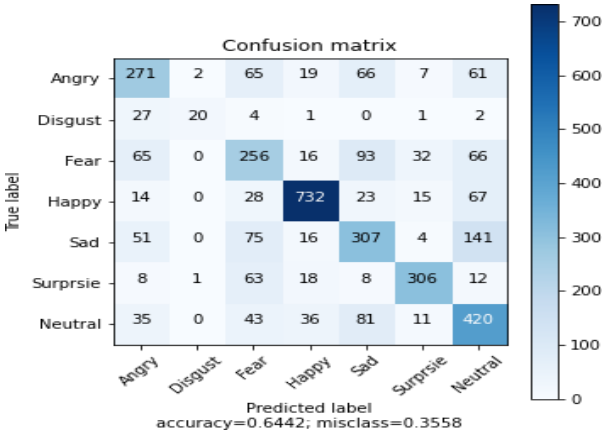

Fig. 5.3 Confusion Matrix

5.1.7 Classification Report

Classification Report

	precision	recall	f1-score	support
0	0.58	0.55	0.56	491
1	0.87	0.36	0.51	55
2	0.48	0.48	0.48	528
3	0.87	0.83	0.85	879
4	0.53	0.52	0.52	594
5	0.81	0.74	0.77	416
6	0.55	0.67	0.60	626
accuracy			0.64	3589
macro avg	0.67	0.59	0.62	3589
weighted avg	0.65	0.64	0.65	3589

Table 5.2 Classification Report

5.2 FERDEMONSTRATION FILE

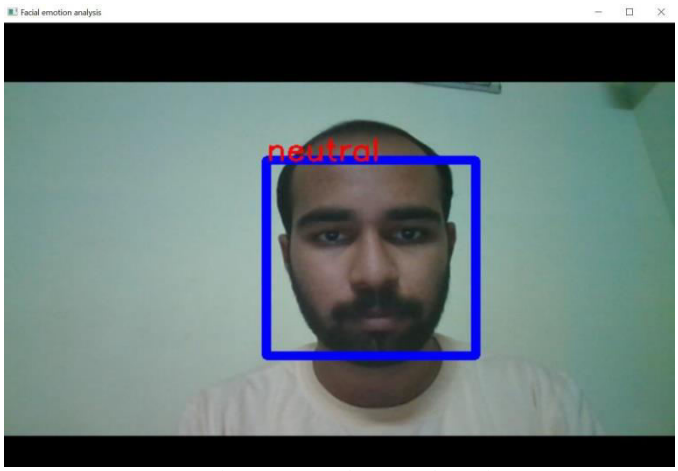

Fig. 5.4 Output for Neutral emotion

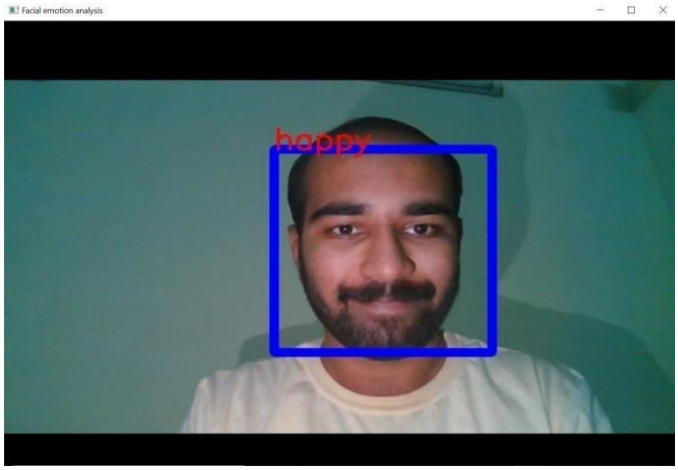

Fig 5.5 Output for Happy emotion

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

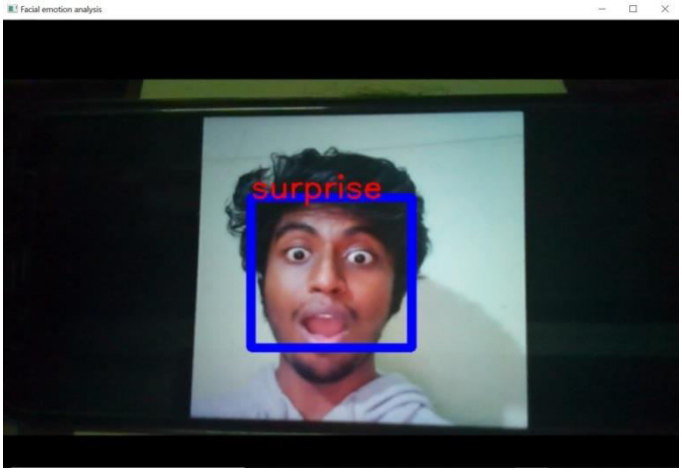

Fig. 5.6 Output for Surprise emotion

Fig. 5.7 Emotions of two persons

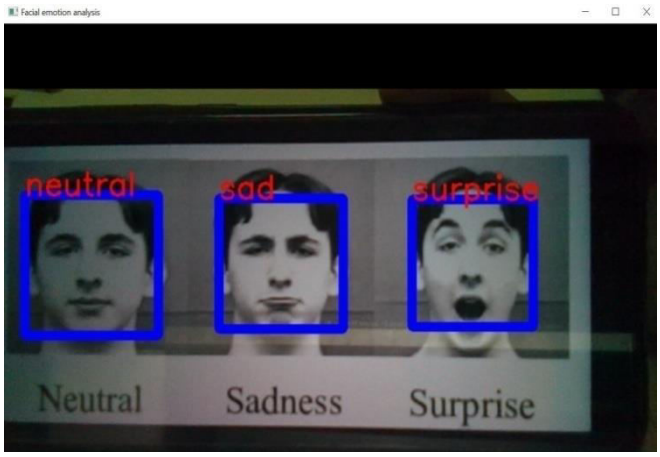

Fig. 5.8 Three emotions of a person

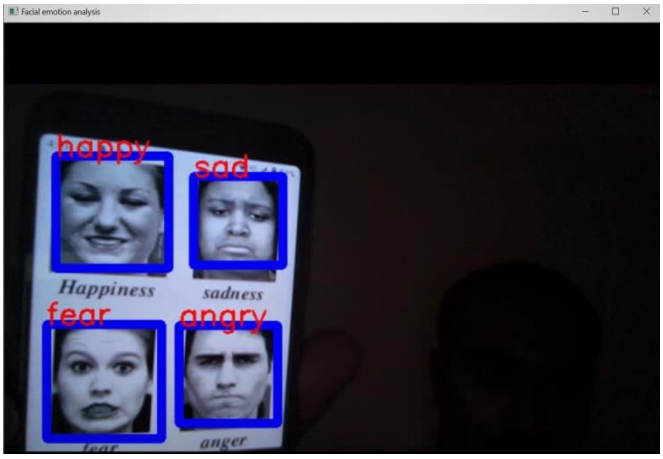

Fig. 5.9 Emotions of four persons

5.3 EMOTION ANALYSIS FOR INTERVIEWS

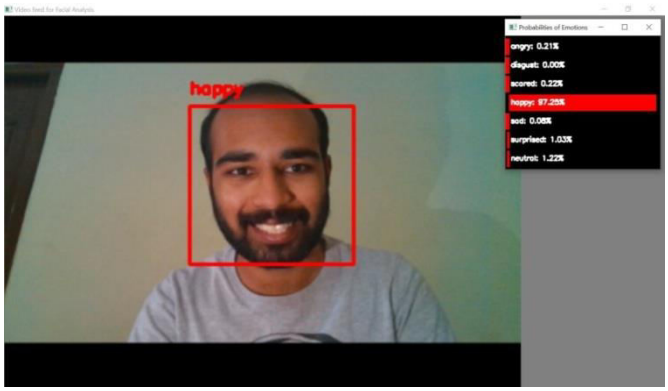

Fig 5.10 Emotion of interviewee (Happy) with probabilities of all 7emotions

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

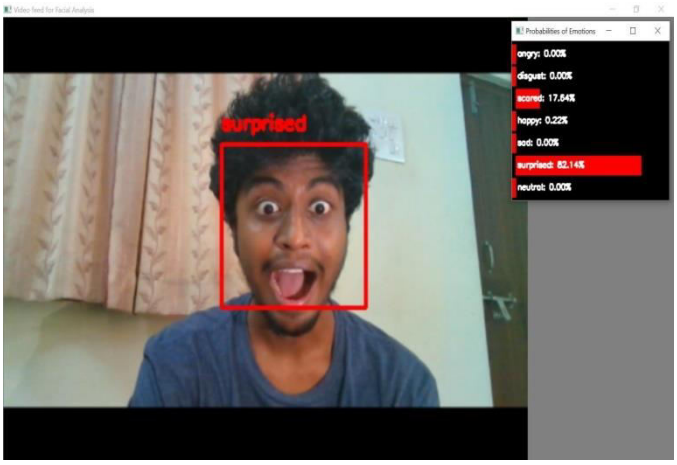

Fig 5.11 Emotion of interviewee (Surprised) with probabilities of all 7emotions

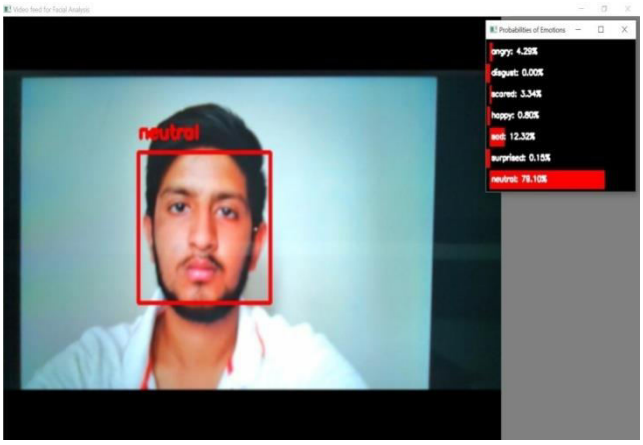

Figure5.12 Emotion of interviewee (Neutral) with probabilities ofall7emotions

5.4 RESULT DESCRIPTION

The results displayed above (Fig 5.4- 5.9) depict that the trained CNN model is capable of handling the following

- A single person with a single emotion
- A single person with multiple emotions
- Multiple persons with multiple emotions

During interviews, the interviewer can measure interviewee's facial expressions to comprehend their emotions and further assess their facial traits as shown in Figures 5.10-5.12.

VI. CONCLUSION

This paper proposes a method for recognizing the category of facial expressions. Face Detection and Extraction of expressions from facial images is useful in many applications, such as digital cameras, robotics vision, HCI, video surveillance and in numerous security aspects. This project's objective was to develop a facial emotion recognition system implementing computer vision and enhancing the advanced feature extraction and classification in facia l emotion recognition. Experiment results on the FER dataset displays that our proposed procedure can achieve an amazing overall performance. Given that Facial emotion recognition is a challenging problem, With this technology, a recruiter will be able to understand, say, the overall confidence level of an interviewee and make a selection approximately whether or not the candidate will be able to carry out well at a client-facing job. Similarly, it is possible

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 9, September 2021 ||

| DOI:10.15680/IJIRSET.2021.1009012 |

to find out whether the candidate is replying to all the questions honestly by measuring the change in emotions during his responses and correlating it with the vast amount of knowledge available in this field.

Future works would include focusing on enhancing the performance of the system and deriving more appropriate classifications which may be useful in many real-world applications.

Accuracy (Training)

print("Model accuracy= ",max(model_info.history['accuracy']))print("Validationaccuracy=",max(model_info.history['val_accuracy']))

```
Model accuracy  =  92.85935163497925
Validation accuracy  =  64.65944051742554
```

Accuracy (Testing)

Predicted_Test_Labels = np.argmax(model.predict(Test_Data, axis=1)
test_labels= np.argmax(Test_Labels,axis=1)
print ("The Accuracy Score is = ", accuracy_score(Test_Labels, Predicted_Test_Labels))

```
Accuracy score =  64.41905823349123
```

REFERENCES

[1] S L Happy, Anjith George, Aurobinda Routray. "A real time facial expression classification system using Local Binary Patterns", 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), 2012
S. Chundru, "Leveraging AI for Data Provenance: Enhancing Tracking and Verification of Data Lineage in FATE Assessment," International Journal of Inventions in Engineering & Science Technology, vol. 7, no.1, pp. 87-104, 2021.
[2] Sumit Das, Manas Kumar Synyal, Sourav Kumar Upadhyay, Supriyo Chatterjee. "An Intelligent Approach for Predicting Emotion Using Convolution Neural Network", Journal of Physics: Conference Series, 2021
[3]G. Vimal Raja, K. K. Sharma (2014). Analysis and Processing of Climatic data using data mining techniques. Envirogeochimica Acta 1 (8):460-467.
[4] R Raja Subramanian, Chunduri Sandya Niharika, Dondapati Usha Rani, Parvatha reddy Pavani, Ketepalli Poojita Lakshmi Syamala. "Design and Evaluation of a Deep Learning Algorithm for Emotion Recognition", 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021
[5] E. M. Bouhabba, A. A. Shafie, R. Akmeliawati. "Support vector machine for face emotion detection on real time basis", 2011 4th International Conference on Mechatronics (ICOM), 2011
[6] Nisha Thomas, Mercy Mathew. "Facial expression recognition system using neural network and MATLAB", 2012 International Conference on Computing, Communication and Applications, 2012
[7] Mahdi Ilbeygi, Hamed Shah-Hosseini. "A novel fuzzy facial expression recognition system based on facial feature extraction from color face images", Engineering Applications of Artificial Intelligence, 2012
[8] "Intelligent Information and Database Systems", Springer Science and Business Media LLC, 2018
[9]G. Vimal Raja, K. K. Sharma (2015). Applying Clustering technique on Climatic Data. Envirogeochimica Acta 2 (1):21-27.
[10] Tan Kiet Nguyen Thanh, Quoc Bao Truong, Quoc Dinh Truong, Hiep Huynh Xuan. "Chapter 53 Depth Learning with Convolutional Neural Network for Leaves Classifier Based on Shape of Leaf Vein", Springer Science and Business Media LLC, 2018
[11] Sumedha Singla, Mingming Gong, Craig Riley, Frank Sciurba, KayhanBatmanghelich. "Improving clinical disease subtyping and future events prediction through a chest CT-based deep learning approach", Medical Physics, 2021
[12]Vaka, P. R. (2021). Zero Trust Security Model. International Journal of Advanced Research in Engineering and Technology (IJARET), 12(6), 148–156.
[13] Darko Jocic, VelimirCirovic, Dragan Aleksendric. "Chapter 16 Identification and Recognition of Vehicle Environment Using Artificial Neural Networks", Springer Science and Business Media LLC, 2019

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details