A Project report on

FACIAL EMOTION RECOGNITION

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

By

T.VEENA	**184G1A05A9**
V.VENKATESH	**184G1A05B1**
K. SAI LAHARI	**184G1A0574**

Under the Guidance of

Mrs. M. Soumya, M.Tech.,(Ph.D)
Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)
(Accredited by NAAC with 'A' Grade &Accredited by NBA (EEE, ECE & CSE)
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu- 515701.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Project report entitled Facial Emotion Recognition is the bonafide work carried out by **T.Veena** bearing Roll Number: **184G1A05A9, V.Venkatesh** bearing Roll Number: **184G1A05B1, K.Sai Lahari** bearing Roll Number: **184G1A0574** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021 - 2022.

Signature of the Guide

Mrs.M.Soumya M.Tech, (Ph.D.)
Assistant Professor

Head of the Department

Mr. P. Veera Prakash M.Tech, (Ph.D.)
Assistant Professor & HOD

Date:

Place: Rotarypuram

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mrs.M.Soumya,** M.Tech., (Ph.D.) **Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project in time.

We express our deep felt gratitude to **Mr. K. Venkatesh,** M.Tech, **Assistant Professor, Project coordinator** for valuable guidance and unstinting encouragement that enabled us to accomplish our project successfully in time.

We are very much thankful to **Mr. P.Veera Prakash,** M.Tech., (Ph.D.), **Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna,** Ph.D., **Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non- teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all therequirements and facilities that we need.

184G1A05A9
184G1A05B1
184G1A0574

Declaration

We, Ms T.Veena with Reg no: 184G1A05A9, Mr. V.Venkatesh with Reg no: 184G1A05B1, Ms K.Sai Lahari with Reg no: 184G1A0574 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY , Rotarypuram , hereby declare that the dissertation entitled "FACIAL EMOTION RECOGNITION" embodies the report of our project work carried outby us during IV year Bachelor of Technology under the guidance of Mrs.M.Soumya M.Tech, (Ph.D.) Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

T.VEENA	Reg no: 184G1A05A9
V.VENKATESH	Reg no: 184G1A05B1
K. SAI LAHARI	Reg no: 184G1A0574

CONTENTS

LIST OF FIGURES

LIST OF SCREENS

LIST OF ABBREVIATIONS

CNN	Convolution Neural Network
FACS	Facial Action Coding System
SIANN	Space Invariant Artificial Neural Network
MLP	Multi-Layer Perceptron
LBP	Local Binary Pattern
VCSs	Version Control Systems
RAM	Random Access Memory
PIL	Python Imaging Library
CV	Computer Vision

ABSTRACT

The ability to analyze facial expressions plays a major role in non verbal communication. If a somebody only analyses what a person's mouth says and ignores what the person's face says, then we can only have a part of the story. Humans were the only ones who could distinguish between expressions but not anymore, with advancing technology our computers can learn how to detect emotions as well. This report is guide to facial expressions recognition software using OpenCV, keras, CNN by implementing a program in python. It has become possible to build an algorithm that perform detection, extraction and evaluation of these facial expressions for automatic recognition of human emotion in real time. The main features of the face are considered for the detection of facial expressions. To determine the different emotions, the variations in each of the main features are used. To detect and classify different classes of emotions, machine learning algorithms are used by training different sets of images. This report discuss a real time emotion classification of a facial expression into one among the seven universal human expressions: Anger, Disgust, Fear, Happy, Sad, Neutral, Surprise by the implementation of a real time vision system that can classify emotions.

CHAPTER 1

INTRODUCTION

Facial expressions are the vital identifiers for human feelings, because it corresponds to the emotions. Most of the times (roughly in 55% cases) , the facial expression is nonverbal way of emotional expression, and it can be considered as concrete evidence to uncover whether an individual is speaking the truth or not .The current approaches primarily focus on facial investigation keeping background intact and hence built up a lot of unnecessary and misleading features that confuse CNN training process. The current manuscript focuses on five essential facial expression classes reported, which are displeasure/anger, sad/unhappy, smiling/happy, feared, and surprised/astonished. The FERC algorithm presented in this manuscript aims for expressional examination and to characterize the given image into these five essential emotion classes. Reported techniques on facial expression detection can be described as two major approaches. The first one is distinguishing expressions that are identified with an explicit classifier, and the second one is making characterization dependent on the extracted facial highlights. In the facial action coding system (FACS) , action units are used as expression markers. These AUs were discriminable by facial muscle changes.

Humans interact with each other mainly through speech, but also through body gestures, to emphasize certain parts of their speech and to display emotions. One of the important ways humans display emotions is through facial expressions which are a very important part of communication. Though nothing is said very bally, there is much to be understood about the messages we send and receive through the use of nonverbal communication. Facial expressions convey nonverbal cues, and they play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioural science and in clinical practice. Although humans recognize facial expressions virtually without effort or delay, reliable expression recognition by machine is still a challenge. There have been several advances in the past few years in terms of face detection, feature extraction mechanisms and the techniques used for expression classification, but development of an automated

system that accomplishes this task is difficult. In this paper, we present an approach based on Convolutional Neural Networks (CNN) for facial expression recognition. The input into our system is an image; then, we use CNN to predict the facial expression label which should be one these labels: anger, happiness, fear, sadness, disgust and neutral.

1.1 BACKGROUND:

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, disgust, and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioural science and in clinical practice. It have been studied for a long period of time and obtaining the progress recent decades. Though much progress has been made, recognizing facial expression with a high accuracy remains to be difficult due to the complexity and varieties of facial expressions.

On a day to day basics humans commonly recognize emotions by characteristic features, displayed as a part of a facial expression. For instance happiness is undeniably associated with a smile or an upward movement of the corners of the lips. Similarly other emotions are characterized by other deformations typical to a particular expression. Research into automatic recognition of facial expressions addresses the problems surrounding the representation and categorization of static or dynamic characteristics of these deformations of face pigmentation. In machine learning, a convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be

approximated mathematically by a convolution operation. Convolutional networks were inspired by biological processes and are variations of multilayer perceptron designed to use minimal amounts of pre-processing. They have wide applications in image and video recognition, recommender systems and natural language processing.

The convolutional neural network is also known as shift invariant or space invariant artificial neural network (SIANN), which is named based on its shared weights architecture and translation invariance characteristics. LeNet is one of the very first convolutional neural networks which helped propel the field of Deep Learning. This pioneering work by Yann LeCun was named LeNet5 was used mainly for character recognition tasks such as reading zip codes, digits, etc. The basic architecture of LeNet can be shown as below:

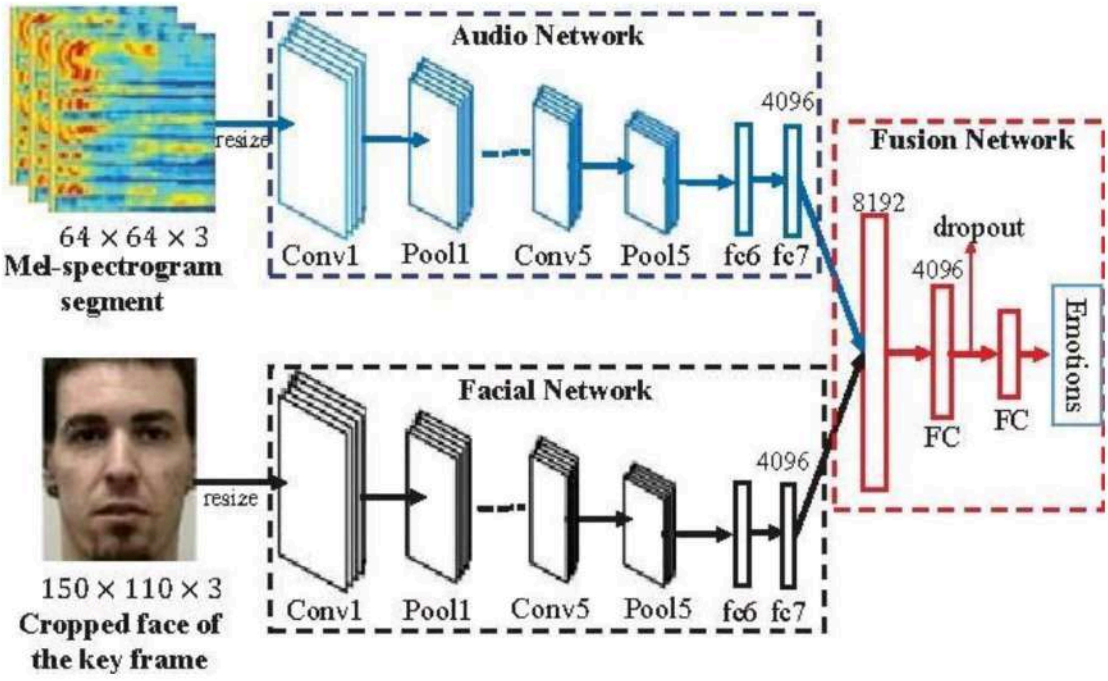


Fig.1.1 A Model of CNN

There are four main operations in the Convolution Neural Network shown in Figure 1.1 above:

Convolution:

The primary purpose of Convolution in case of a CNN is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. The convolution layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on a first layer of a CNN might have size 3x5x5 (i.e. images have depth 3 i.e. the colour channels, 5 pixels width and height).

During the forward pass, each filter is convolved across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As the filter convolve over the width and height of the input volume it produces a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an 3 edge of some orientation or a blotch of some colour on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network.

A filter convolves with the input image to produce a feature map. The convolution of another filter over the same image gives a different feature map. Convolution operation captures the local dependencies in the original image. A CNN learns the values of these filters on its own during the training process (although parameters such as number of filters, filter size, architecture of the network etc. still needed to specify before the training process). The more number of filters, the
more image features get extracted and the better network becomes at recognizing patterns in unseen images. The size of the Feature Map (Convolved Feature) is controlled by three parameters

- Depth: Depth corresponds to the number of filters we use for the convolution operation.
- Stride: Stride is the size of the filter, if the size of the filter is 5x5 then stride is 5.

- Zero-padding: Sometimes, it is convenient to pad the input matrix with zeros around the border, so that filter can be applied to bordering elements of input image matrix. Using zero padding size of the feature map can be controlled.

Rectified Linear Unit:

An additional operation called ReLU has been used after every Convolution operation. A Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation function to calculate its output given x:

$$\mathbf{R(x) = Max(0,x)}$$

Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. When initializing the weights uniformly, half of the weights are negative. This helps creating a sparse feature representation. Another positive aspect is the relatively cheap computation. No exponential function has to be calculated. This function also prevents the vanishing gradient error, since the gradients are linear functions or zero but in no case non-linear functions.

Pooling (sub-sampling):

Spatial Pooling (also called subsampling or down sampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc. In case of Max Pooling, a spatial neighbourhood (for example, a 2×2 window) is defined and the largest element is taken from the rectified feature map within that window. In case of average pooling the average or sum of all elements in that window is taken. In practice, Max Pooling has been shown to work better. Max Pooling reduces the input by applying the maximum function over the input xi. Let m be the size of the filter, then the output calculates as follows:

$$M(x_i) = max\ \{xi+k,+l\ |k| \leq m/2\ ,\ |l| \leq m/2\ k,\ l\epsilon\ N\}$$

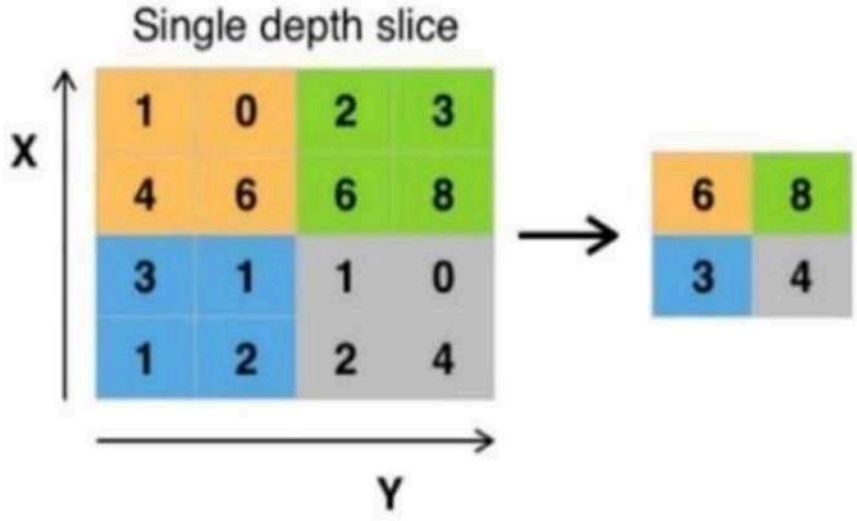


Figure 1.2: Max Pooling

The function of Pooling is to progressively reduce the spatial size of the input representation. In particular, pooling

- Makes the input representations (feature dimension) smaller and more manageable
- Reduces the number of parameters and computations in the network, therefore, controlling over-fitting
- Makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in input will not change the output of Pooling.
- Helps us arrive at an almost scale invariant representation. This is very powerful since objects can be detected in an image no matter where they are located.

Classification (Multilayer Perceptron):

The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a SoftMax activation function in the output layer. The term "Fully Connected" implies that every neuron in the previous layer is connected to every neuron on the next layer. The output from the convolutional and pooling layers represent high-level features of the input image.

The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset. SoftMax is used for activation function. It treats the outputs as scores for each class. In the SoftMax, the function mapping stayed unchanged and these scores are interpreted as the unnormalized log probabilities for each class. SoftMax is calculated as:

$$f(z)_j = \frac{exp(zj)}{\sum_{k=1}^{K} exp(zk)}$$

where j is index for image and K is number of total facial expression class.

Apart from classification, adding a fully-connected layer is also a (usually) cheap way of learning non-linear combinations of these features. Most of the features from convolutional and pooling layers may be good for the classification task, but combinations of those features might be even better.

The sum of output probabilities from the Fully Connected Layer is 1. This is ensured by using the as the activation function in the output layer of the Fully Connected Layer. The SoftMax function takes a vector of arbitrary real-valued scores and squashes it to a vector of values between zero and one that sum to one.

CHAPTER - 2

LITERATURE SURVEY

[1] "An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment" by D. Yang , Abeer Alsadoon, P.W.C.Prasad , A. K. Singh, A. Elchouemi

Description:

Sablik, Velten, and Kummert use object cascade classifier Haar feature detection. It is a machine learning method, which is based on a large number of positive and negative image training cascade functions. Then it is used to detect objects in other images. For the proposal solution, firstly we train input images, get the face features and store these in face.xml; secondly, we detect eyes and mouth based on facial features.

In this paper provided a proposed model to solve the problems of emotion recognition based on facial recognition in virtual learning environments, and the efficiency and accuracy are considered at the same time. Using HAAR Cascades to detect eyes and mouth and identify all kinds of emotion through the neural network method, the combination of efficiency and accuracy is achieved. It can be applied to real distance education. The application of emotion recognition in virtual learning environments is a much-researched topic. In addition to the change of uncertainty factors makes teachers and students face pattern is more complex, so the emotion recognition in the online learning network application mode is a very challenging topic. Accordingly, it is proposed that the subject requires further research from the following aspects. Due to the fact that this research does not involve the illumination and pose of the image, it is uncertain how much these factors influence facial expressions and thus the final emotion recognition. All of these issues need to be explored in future research and validated by experiments, To make the theory and technology of emotion recognition fully meet the practical requirement, there are suggesting that the comprehensive application of image processing, pattern recognition, computer vision and neural networks, psychology, cognitive science, and integrates with other biometric authentication methods and methods of human-computer interaction perception based on the in-depth and meticulous research work.

[2] "Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure". By JI-HAE KIM, BYUNG-GYU KIM, (Senior Member, IEEE), PARTHA PRATIM ROY, (Member, IEEE), AND DA-MI JEONG

Description:

They have proposed an efficient facial expression recognition algorithm combining appearance feature and geometric feature based on deep neural networks for more accurate and efficient facial expression recognition. The appearance feature-based network extracts the holistic feature of the LBP feature containing the AUs information. The geometric feature-based network extracts the dynamic feature, which is the face landmark change centered on the coordinate movement between the neutral face and the peak emotion. As a result, we constructed more robust feature by combining static appearance feature from the appearance network and dynamic feature from the geometric feature-based network. In the experiments, we have shown that the Top-2 error frequently occurred with average about 82% using only appearance feature-based network. As a result of improving this error with the proposed algorithm, we achieved about 96.5% accuracy with 1.3% improvement when comparing to the other algorithms in the CK+ dataset. In addition, the proposed algorithm yielded 91.3% of the accuracy which was improved by 1.5% when compared with other existing methods in the JAFFE dataset. From the experiments for all datasets, the accuracy of six emotions was at least maintained or enhanced significantly.

[3] "A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing" by Hongli Zhang, Alireza Jolfaei , and Mamoun Alazab

Description:

The original pictures of facial expressions have complex background, different sizes, different shades and other factors, a series of image pre-processing processes have to be completed before facial expressions are input into the network for training. Firstly, we locate the face in the image and cut out the face image. Then, we normalize the face image to a specific size. Next, we equalize the histogram of the image to reduce the influence of illumination and other factors. Finally, we extract the edge of each layer of the image in the convolution process. The extracted edge information is superimposed on each feature image to preserve the edge structure information of texture image.

Here they uses a Haar classifier for human detection. The Haar classifier is trained by Haar-like small features and an integral graph method combined with the AdaBoost algorithm. The Haar-like is a commonly used texture descriptor, and its main features are linear, edge, centre and diagonal. Adaboost is an improvement of Boosting algorithm and its core idea is to form a strong classifier by iterating not only weak classifiers but also weak classifiers. The ViolaJones detector is a milestone in the history of face detection. It has been widely used because of its high efficiency and fast detection. This method uses the Haar-like to extract facial features, and uses an integral graph to realize fast calculation of Haar-like features, and screens out important features from a large number of Haar-like features. Then, we use the Adaboost algorithm to train and integrate the weak classifier into a strong classifier. Finally, several strong classifiers are cascaded in series to improve the accuracy of the face detection.

In the actual image acquisition process, it is easy to be affected by illumination, shadows and other factors, which makes the collected image show a state of uneven distribution of light and shade, which will increase the difficulty of feature extraction. Therefore, it is necessary to average the Gray level of the image to enhance the contrast of the image. In this paper, the Histogram Equalization (HE) method is used to process images.

[4] " Facial emotion recognition using convolutional neural networks (FERC)" by Ninad Mehendale".

Description:

Convolutional neural network (CNN) is the most popular way of analysing images. CNN is different from a multi-layer perceptron (MLP) as they have hidden layers, called convolutional layers. The proposed method is based on a two-level CNN framework. The first level recommended is background removal, used to extract emotions from an image. Here, the conventional CNN network module is used to extract primary expressional vector (EV). The expressional vector (EV) is generated by tracking down relevant facial points of importance. EV is directlyrelated to changes in expression. The EV is obtained using a basic perceptron unit applied on a background-removed face image. In the proposed FERC model, we also have a non-convolutional perceptron layer as the last stage. Each of the convolutional layers receives the input data (or image), transforms it, and then outputs it to the next

level. FERC works with an image as well as video input. In case, when the input to the FERC is video, then the difference between respective frames is computed. The maximally stable frames occur whenever the intra-frame difference is zero. Then for all of these stable frames, a Canny edge detector was applied, and then the aggregated sum of white pixels was calculated. After comparing the aggregated sums for all stable frames, the frame with the maximum aggregated sum is selected because this frame has maximum details as per edges (more edges more details). This frame is then selected as an input to FERC. The logic behind choosing this image is that blurry images have minimum edges or no edges.

Once the input image is obtained, skin tone detection algorithm is applied to extract human body parts from the image. This skin tone-detected output image is a binary image and used as the feature, for the first layer of background removal CNN (also referred to as the first-part CNN in this manuscript). This skin tone detection depends on the type of input image. If the image is the coloured image, then YCbCr colour threshold can be used. For skin tome, the Y-value should be greater than 80, Cb should range between 85 and 140, Cr value should be between 135 and 200. The set of values mentioned in the above line was chosen by trial-and-error method and worked for almost all of the skin tones available. We found that if the input image is grayscale, then skin tone detection algorithm has very low accuracy. To improve accuracy during background removal, CNN also uses the circles-in-circle filter. This filter operation uses Hough transform values for each circle detection. To maintain uniformity irrespective of the type of input image, Hough transform was always used as the second input feature to background removal CNN.

[5] "Facial emotion recognition using convolutional neural networks" by Ketan Sarvakar, R. Senkamalavalli , S. Raghavendra , J. Santosh Kumar , R. Manjunath, Sushma Jaiswal"

Description:

Six two-dimensional convolution layers, two maximum pooling levels, and two fully connected layers comprise the network. The preceding layer's Max Neurons use the maximum value for each cluster. This lowers the number of dimensions. Array of output values Array of output values Array of output values Array of output values. The network input is pre-processed 48 by 48 pixel face. The model was created by studying

the performance of earlier iterations. It was chosen to go via a larger network. The benefit of having additional layers is that memorising is avoided. A large yet shallow network is readily stored, but it is not adequately generalised. Multi-layer networks provide abstraction functions that can be generalised effectively. The number of layers used to ensure high accuracy while being quick enough for real-time applications. The suggested CNN varies from a conventional CNN in that it employs four different coevolutionary and coevolutionary filter sizes. It also used maximum pooling and drop-out to reduce overfitting. The network has two convolutional layers, each with a filter size of 64. Then comes a max layer of pooling. To minimise overfitting, a drop of 0.25 is utilised. A series of four convolutionary layers follows.

The first two have 128 filters, whereas the second two contain 256 filters. A single layer of pooling After these four levels, there is a 0.25 decrease. The preceding output layers were smoothed to turn the result into a single vector dimension. After that, a fully connected 0.001 penalty controller L2 layer is utilised. Along reduces at a 0.5 percent faster pace. Finally, a complete layer with a SoftMax activation function is used. The output layer the kernel size is defined as the 2D width and height. Every convolutionary window is made up of three levels. Each maximum pooling layer has two dimensions and a pool size of two by two. This reduces output by 50% after each pooling layer. Every layer in the output layer utilised a ReLU. Activation of a Function Activation of a Function The ReLU activation function is now available. Because of advantages such as sparsity and a lower probability of gradient vanishing. The activation function SoftMax has been used. The chance that the final production level will be reached. Every feeling. Every emotion. Every emotion. Every feeling. For the test, this model offered a 0.55 foundation. The hyperparameters, particularly the lot, were then set. Size, optimizer, and epoch count Each model was programmed to run for a total of 100 epochs. However, network time and computing power were allowed to be saved. Stop training if the precision of the epochs does not change. There have been no changes in accuracy across four consecutive epochs if the network stops training. Particularly when accuracy in previous periods saved time and processing resources. Earlier times he choice was made since none of the models were more than 20 years old.

CHAPTER 3

METHODOLOGY

3.1 Facial Emotion Recognition

Convolutional neural network (CNN) is the most popular way of analysing images. CNN is different from a multi-layer perceptron (MLP) as they have hidden layers, called convolutional layers.

The block diagram of the system is shown in following figures:

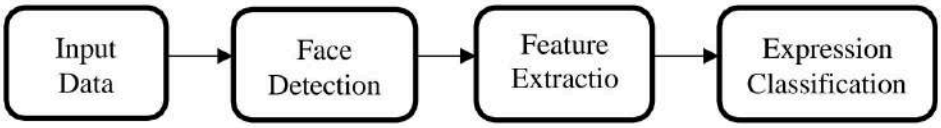


Fig.3.1. Block diagram of the facial emotion recognition

3.1.1 Input Data

The dataset from a Kaggle Facial Expression Recognition Challenge (FER2013) is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labelled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has training set of 35887 facial images with facial expression labels. The dataset has class imbalance issue, since some classes have large number of examples while some has few. The dataset is balanced using oversampling, by increasing numbers in minority classes. The balanced dataset contains 40263 images, from which 29263 images are used for training, 6000 images are used for testing, and 5000 images are used for validation.

Fig 3.1.1 Images for different emotions

3.1.2 Face Detection

Facial detection is the first part of our pipeline. We provide input as image or video or live camera. We have used the python library Face Recognition that we found easy to install and very accurate in detecting faces. This library scans the input image and returns the bounding box coordinates of all detected faces as shown below:

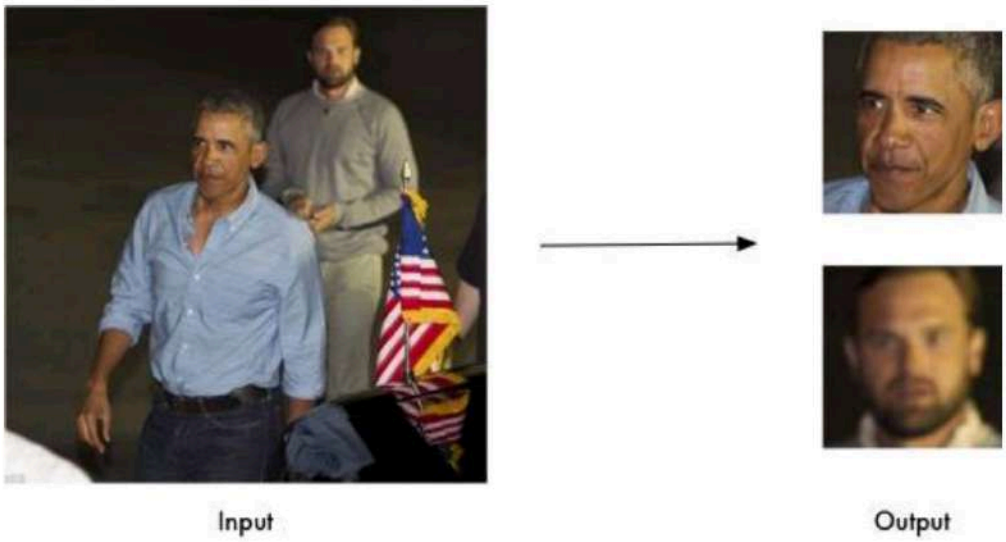


Fig 3.1.2 Example for Face Detection

The below snippet shows how to use the face_recognition library for detecting faces.

```
face_locations = face_recognition.face_locations(image)

top, right, bottom, left = face_locations[0]

face_image = image[top:bottom, left:right]
```

It take single frame as input and mark the all faces are located on that frame and crop the face from the frame and provide the output as different human faces which are available on the input frame.

3.1.3 Feature Extraction:

Selection of the feature vector is the most important part in a pattern classification problem. The image of face after pre-processing is then used for extracting the important features. The inherent problems related to image classification include the scale, pose, translation and variations in illumination level . The important features are extracted using LBP algorithm which is described below:

3.1.3.1 Local Binary Pattern:

LBP is the feature extraction technique. The original LBP operator points the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel. Each pixel is compared with its eight neighbours in a3 X 3 neighbourhood by subtracting the center pixel value. In the result, negative values are encoded with 0 and the others with 1. For each given pixel, a binary number is obtained by merging all these binary values in a clockwise direction, which starts from the one of its top-left neighbour. The corresponding decimal value of the generated binary number is then used for labelling the given pixel. The derived binary numbers are referred to be the LBPs or LBP codes.

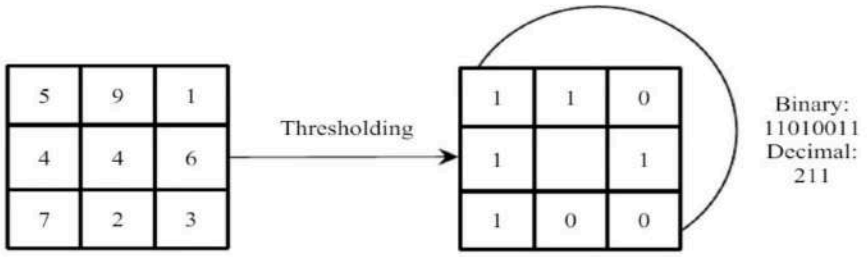
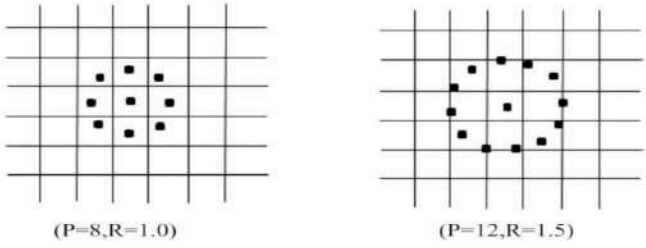


Fig 3.1.3 The Basic LBP Operator



The limitation of the basic LBP operator is that its small 3×3 neighborhood cannotcapture the dominant features with large scale structures. As a result, to deal with thetexture at different scales, the operator was later extended to use neighborhoods of different sizes .Using circular neighborhoods and bilinearly interpolating the pixelvalues allow any radius and number of pixel in the neighborhood. Further extension of LBP is to user uniform patterns. A LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is

considered circular. E.g.00000000, 001110000 and 11100001 are uniform patterns.

3.1.4 Expression Classification:

Expression Classification is used to classify the expressions, there are six expressions that are classified as 'neutral', 'happy', 'sad', 'angry', 'surprised', and 'disgusted'. With the help of CNN, we can classify the expression, particularly we are using Max Pooling. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

Pooling Layer:

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction.

On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

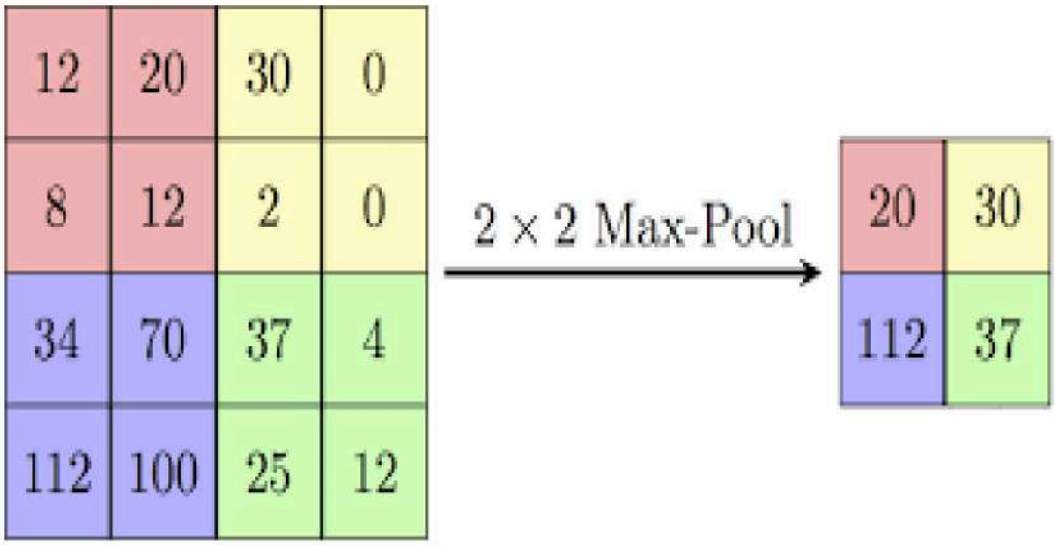


Fig 3.1.4 Max pooling

3.2 Proposed Work:

The proposed work is to recognise the multiple image emotions from the given input data frame.

Fig 3.2 Flow diagram for the proposed work

3.2.1 Input Data:

We can input data as image or video or live camera. Every time it take input as a frame. The frame must contain multiple faces and objects.

3.2.2 Pre-processing:

The facial expression recognition system is trained using supervised learning approach in which it takes images of different facial expressions. The system includes the training and testing phase followed by image acquisition, face detection, image pre-processing, feature extraction and classification. Face detection and feature extraction are carried out from face images and then classified into six classes belonging to six basic expressions which are outlined below:

3.2.3 Image Acquisition:

Images used for facial expression recognition are static images or image sequences. Images of face can be captured using camera.

3.2.4 Face detection:

Face Detection is useful in detection of facial image. Face Detection is carried out in training dataset using Haar classifier called Voila-Jones face detector and implemented through OpenCV. Haar like features encodes the difference in average intensity indifferent parts of the image and consists of black and white connected rectangles in which the value of the feature is the difference of sum of pixel values in

black and white regions.

3.2.5 Image Pre-processing:

Image pre-processing includes the removal of noise and normalization against the variation of pixel position or brightness.

a) Colour Normalization

b) Histogram Normalization

3.2.6 Feature Extraction:

Selection of the feature vector is the most important part in a pattern classification problem. The image of face after pre-processing is then used for extracting the important features. The inherent problems related to image classification include the scale, pose, translation and variations in illumination level . The important features are extracted using LBP algorithm which is described below:

3.2.6.1 Local Binary Pattern:

LBP is the feature extraction technique. The original LBP operator points the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel. Each pixel is compared with its eight neighbours in a3 X 3 neighbourhood by subtracting the center pixel value. In the result, negative values are encoded with 0 and the others with 1. For each given pixel, a binary number is obtained by merging all these binary values in a clockwise direction, which starts from the one of its top-left neighbour. The corresponding decimal value of the generated binary number is then used for labelling the given pixel. The derived binary numbers are referred to be the LBPs or LBP codes.

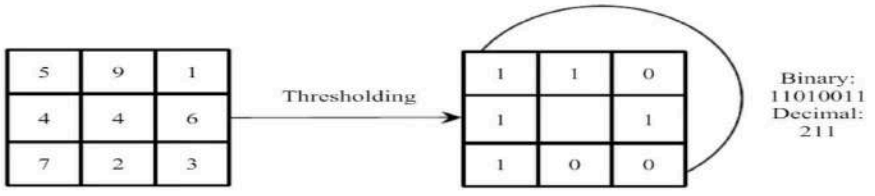


Fig 3.2.6.1 The Basic LBP operator

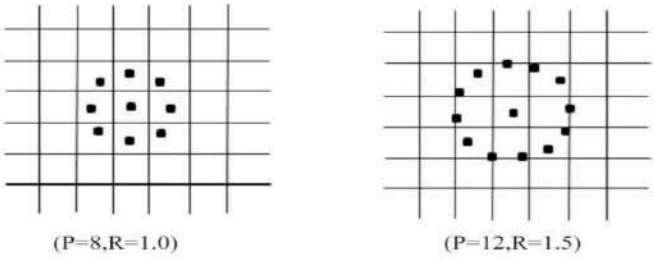


Fig 3.2.6.1.1 Two examples for extended LBP

The limitation of the basic LBP operator is that its small 3×3 neighborhood cannotcapture the dominant features with large scale structures. As a result, to deal with thetexture at different scales, the operator was later extended to use neighborhoods of different sizes . Using circular neighborhoods and bilinearly interpolating the pixelvalues allow any radius and number of pixel in the neighborhood.Further extension of LBP is to user uniform patterns. A LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. E.g.00000000, 001110000 and 11100001 are uniform patterns.

3.2.7 Classification:

The extracted features are passed through a classifier to find its corresponding labels. The classifier used here is a multiclass Support Vector Machine . A Support Vector Machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. The multi-class SVM is implemented for a set of data with M classes, where M binary classifiers can be trained that can distinguish each class against all other classes and then select the class that classifies the test sample with the greatest margin.

3.2.8 Output:

The output must be emotions like angry, disgust, fear, happy, neutral, sad, surprise.

3.2.9 System Design:

Designing the CNN model for emotion detection using functional API. We are creating blocks using Conv2D layer, Batch-Normalization, Max-Pooling2D, Dropout, Flatten, and then stacking them together and at the end-use Dense Layer for output

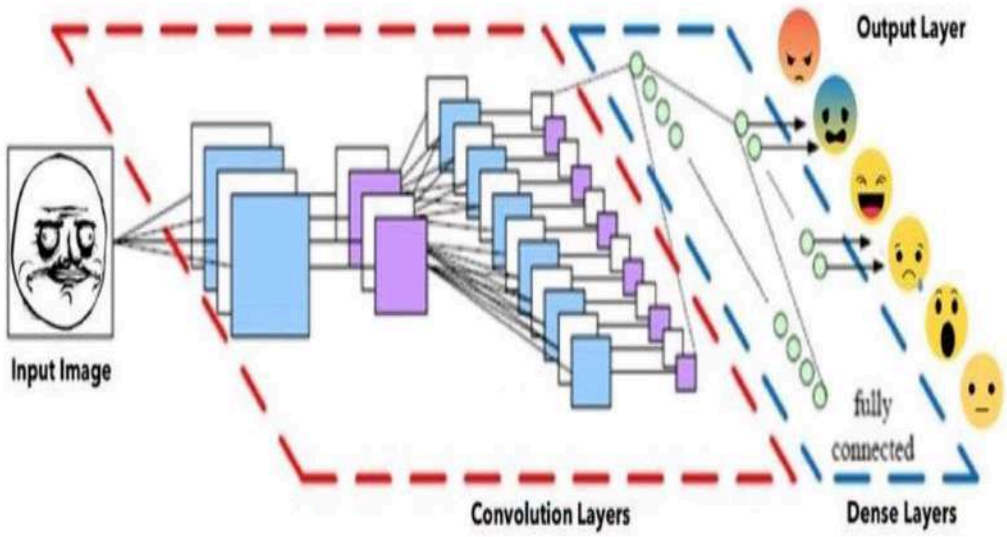


Fig 3.2.9 Layers of Convolutional Neural Network

CHAPTER 4
CONVOLUTIONAL NETWORKS

Convolutional Neural Networks:

It is a class of deep neural networks that extracts features from images, given as input, to perform specific tasks such as image classification, face recognition and semantic image system. A CNN has one or more convolution layers for simple feature extraction, which execute convolution operation (i.e. multiplication of a set of weights with input) while retaining the critical features (spatial and temporal information) without human supervision. CNN's were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. This was a major drawback forCNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

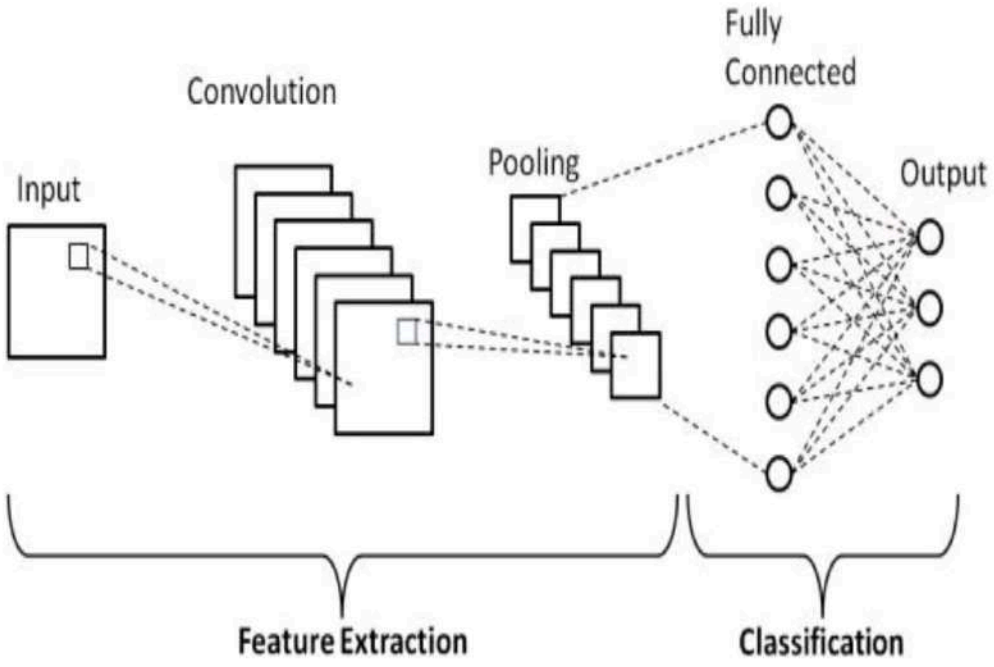


Fig.4.1 Architecture of CNN

4.1 Working of CNN:

In deep learning a convolutional neural network (**CNN/ConvNet**) is a class deep neural networks, most commonly applied to analyze visual imagery. It uses a special technique called Convolution. Now in mathematics **c**onvolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other. CNN is composed of three layers where input image is divided into matrix of pixels. Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. Features of a fully connected layer. Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification**.** The three layers of CNN are:

- Convolutional layer
- Pooling layer
- Fully connected layer

4.2 Input Layer:

The input layer has pre-determined, fixed dimensions, so the image must be pre-processed before it can be fed into the layer. Normalized Gray scale images of size 48 X 48 pixels from Kaggle dataset are used for training, validation and testing. For testing propose laptop webcam images are also used, in which face is detected and cropped using OpenCV Haar Cascade Classifier and normalized. The input layer of a neural network is composed of artificial input neurons, and brings the initial data into the system for further processing by subsequent layers of artificial neurons. The input layer is the very beginning of the workflow for the artificial neural network.

4.3 Sequential Layer:

Sequential is the easiest way to build a model in Keras. It allows you to build a model layer by layer. We use the 'add()' function to add layers to our model. Our first 2 layers are Conv2D layers. These are convolution layers that will deal with our input images, which are seen as 2-dimensional matrices.64 in the first layer and 32 in the second layer are the number of nodes in each layer. This number can be adjusted to be higher or lower, depending on the size of the dataset. In our case, 64 and 32 work well,

so we will stick with this for now. Kernel size is the size of the filter matrix for our convolution. So a kernel size of 3 means we will have a 3x3 filter matrix. Refer back to the introduction and the first image for a refresher on this. Activation is the activation function for the layer. The activation function we will be using for our first 2 layers is the ReLU, or Rectified Linear Activation. This activation function has been proven to work well in neural networks. Our first layer also takes in an input shape. This is the shape of each input image, 28,28,1 as seen earlier on, with the 1 signifying that the images are greyscale. In between the Conv2D layers and the dense layer, there is a 'Flatten' layer. Flatten serves as a connection between the convolution and dense layers. 'Dense' is the layer type we will use in for our output layer. Dense is a standard layer type that is used in many cases for neural networks. We will have 10 nodes in our output layer, one for each possible outcome (0–9). The activation is 'SoftMax'. SoftMax makes the output sum up to 1 so the output can be interpreted as probabilities. The model will then make its prediction based on which option has the highest probability.

4.4 Convolutional Layer:

This is the first layer of the convolutional network that performs feature extraction by sliding the filter over the input image. The output or the convolved feature is the element-wise product of filters in the image and their sum for every sliding action. The output layer, also known as the feature map, corresponds to original images like curves, sharp edges, textures, etc. In the case of networks with more convolutional layers, the initial layers are meant for extracting the generic features while the complex parts are removed as the network gets deeper. The image below shows the convolution operation.

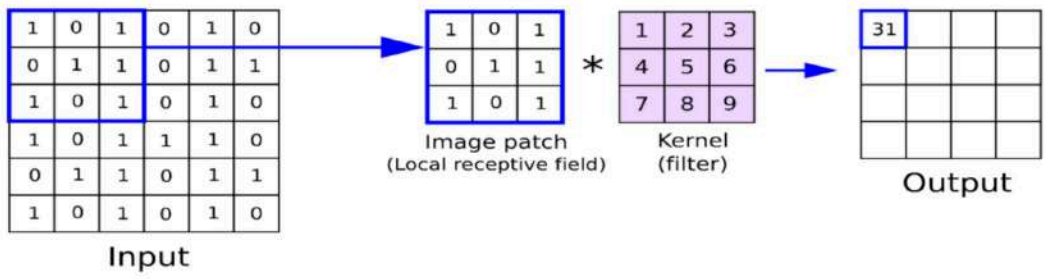


Fig.4.4 Convolutional Layer

4.5 Pooling Layer:

The primary purpose of this layer is to reduce the number of trainable parameters by decreasing the spatial size of the image, thereby reducing the computational cost. The image depth remains unchanged since pooling is done independently on each depth dimension. Max Pooling is the most common pooling method, where the most significant element is taken as input from the feature map. Max Pooling is then performed to give the output image with dimensions reduced to a great extent while retaining the essential information.

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

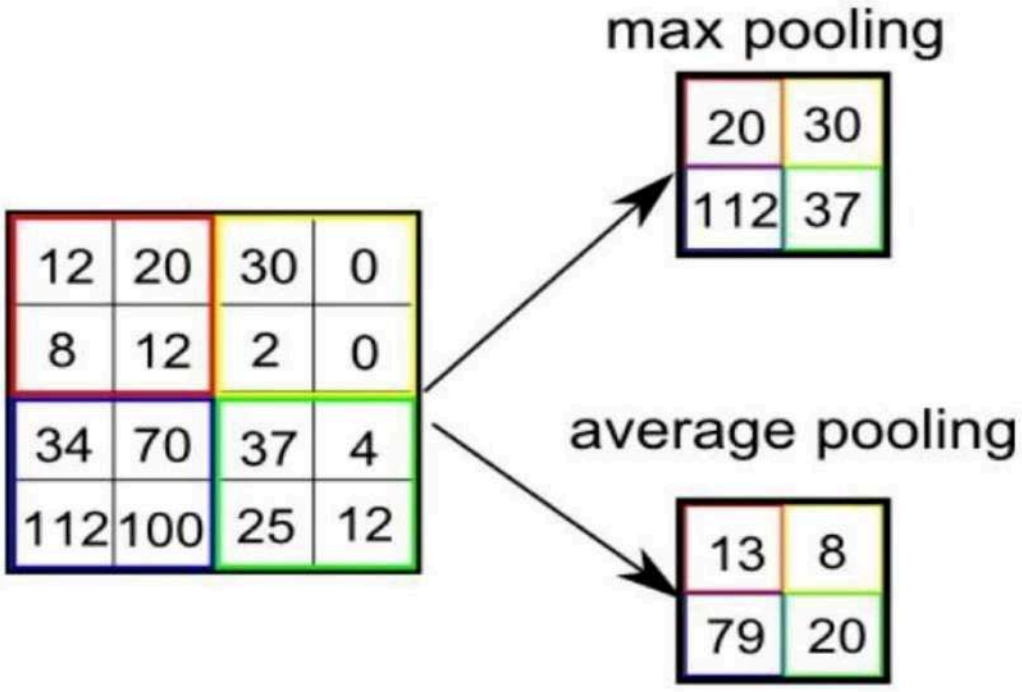


Fig.4.5 Pooling Layer

4.6 Fully Connected Layer:

The last few layers which determine the output are the fully connected layers. The output from the pooling layer is Flattened into a one-dimensional vector and then given as input to the fully connected layer. The output layer has the same number of neurons as the number of categories we had in our problem for classification, thus associating features to a particular label. After this process is known as forwarding propagation, the output so generated is compared to the actual production for error generation. The error is then backpropagated to update the filters(weights) and bias values. Thus, one training is completed after this forwarding and backward propagation cycle.

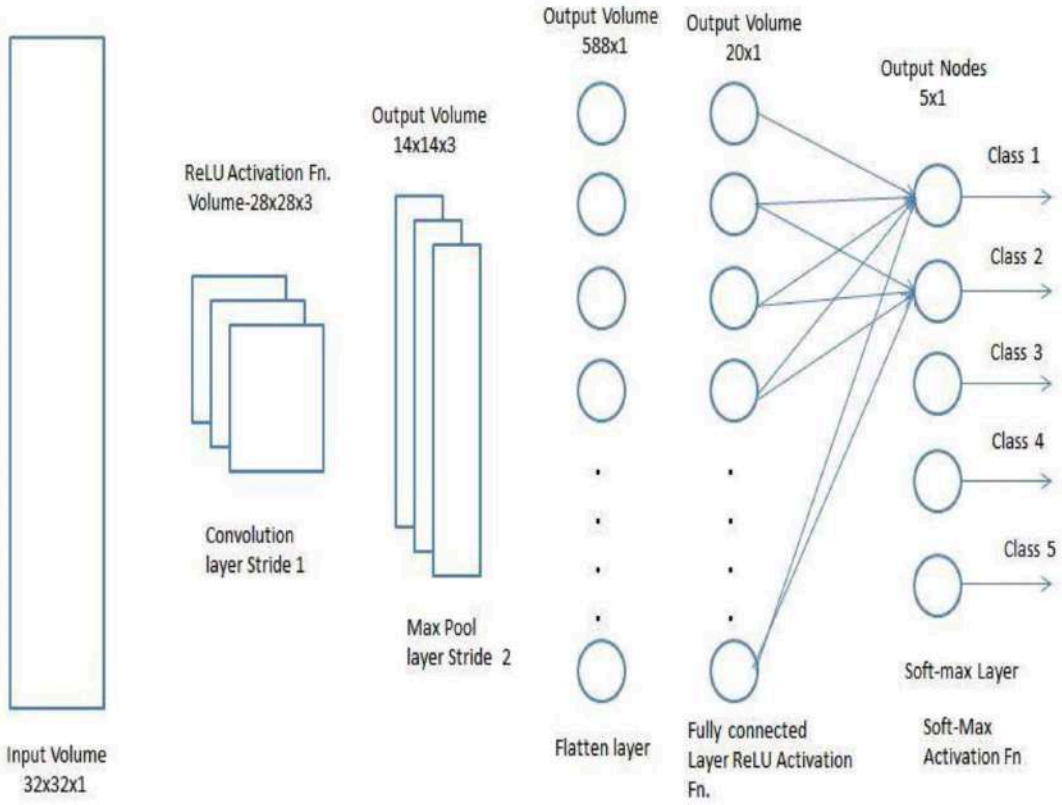


Fig.4.6 Fully Connected Layer

4.6.1 Feedforward and Feed backward:

A feedforward network is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element computes based upon the weighted sum of its inputs. The new values become the new input values that feed the next layer (feed-forward). This continues through all the layers and determines the output. Feedforward networks are often used in, for example, data mining.

A feedback network (for example, a recurrent neural network) has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed. Since loops are present in this type of network, it becomes a non-linear dynamic system which changes continuously until it reaches a state of equilibrium. Feedback networks are often used in optimization problems where the network looks for the best arrangement of interconnected factors.

4.7 Flatten Layer:

After finishing the previous steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below.

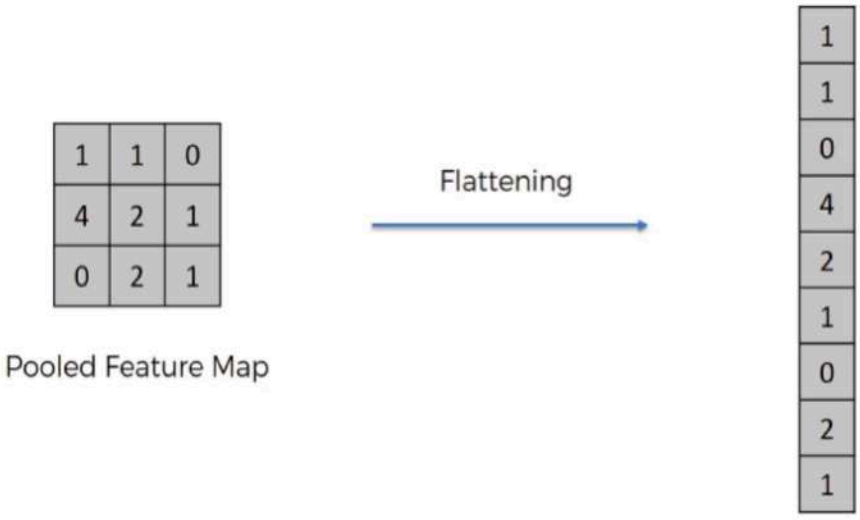


Fig.4.7 Flatten Layer

4.8 Dense Layer:

In any neural network, a dense layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer. This layer is the most commonly used layer in artificial neural network networks. The dense layer's neuron in a model receives output from every neuron of its preceding layer, where neurons of the dense layer perform matrix-vector multiplication. Matrix vector multiplication is a procedure where the row vector of the output from the preceding layers is equal to the column vector of the dense layer. The general rule of matrix-vector multiplication is that the row vector must have as many columns like the column vector.

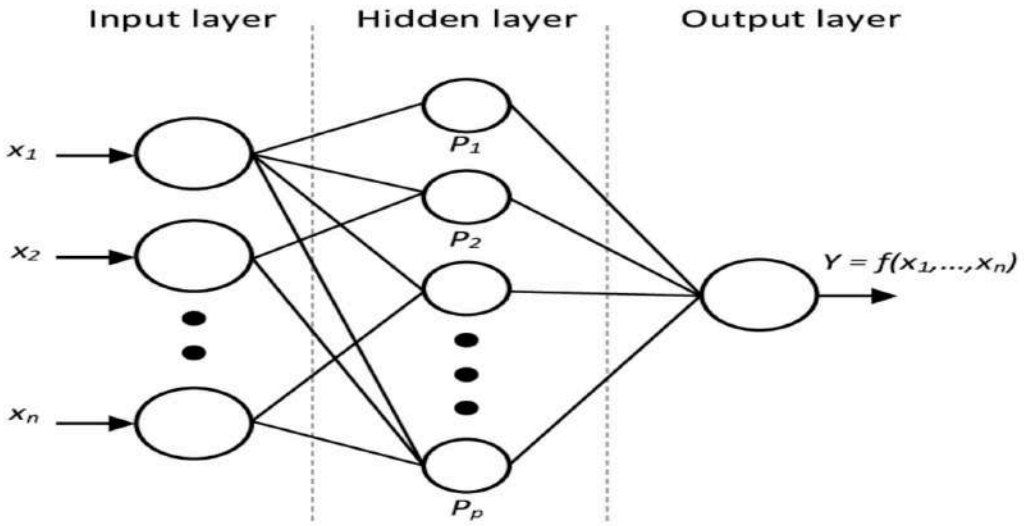


Fig.4.8 Dense Layer

4.9 Output Layer:

Output from the second hidden layer is connected to output layer having seven distinct classes. Using SoftMax activation function, output is obtained using the probabilities for each of the seven class. The class with the highest probability is the predicted class. The output layer is responsible for producing the final result. There must always be one output layer in a neural network. The output layer takes in the inputs which are passed in from the layers before it, performs the calculations via its neurons and then the output is computed. The output layer is the layer in a neural network model that directly outputs a prediction. All feed-forward neural network models have an output layer.

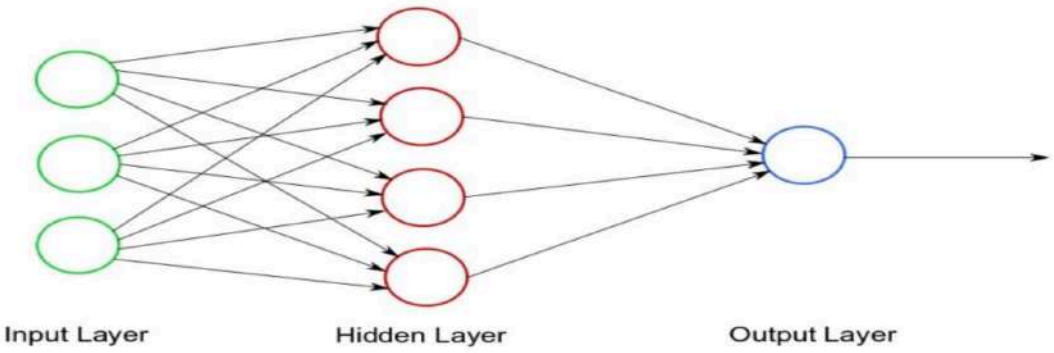


Fig.4.9 Output Layer

4.10 Activation Functions:

The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1. It's an abstraction that represents the rate of action potential firing in the cell. It's a number that represents the likelihood that the cell will fire. The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range.

4.10.1 Rectified Linear Unit:

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

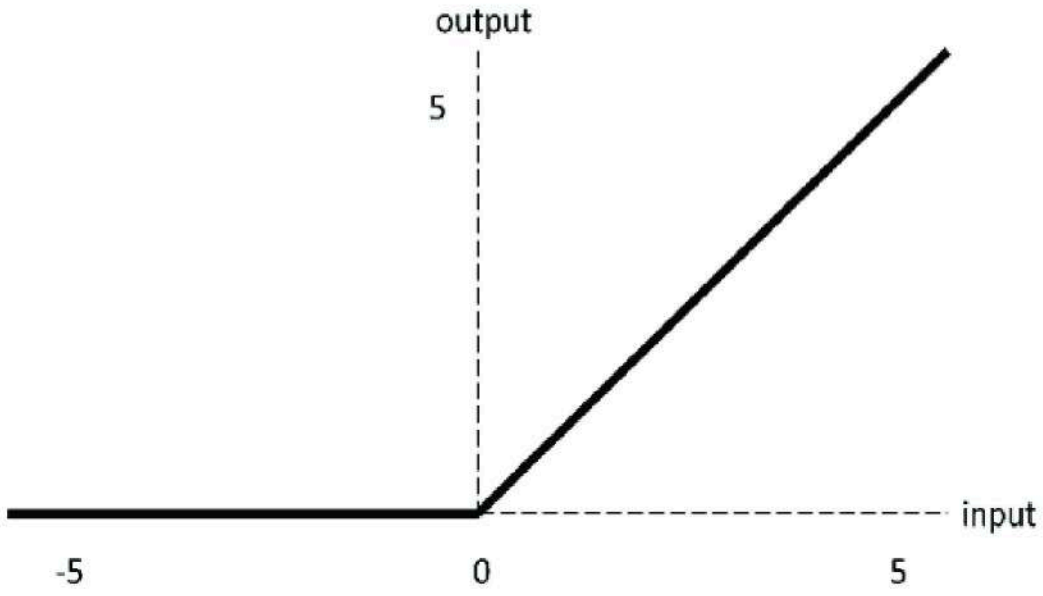


Fig.4.10.1 Relu Function

4.10.2 SoftMax Function:

SoftMax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the

relative scale of each value in the vector. The most common use of the SoftMax function in applied machine learning is in its use as an activation function in a neural network model. Specifically, the network is configured to output N values, one for each class in the classification task, and the SoftMax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the SoftMax function is interpreted as the probability of membership for each class.

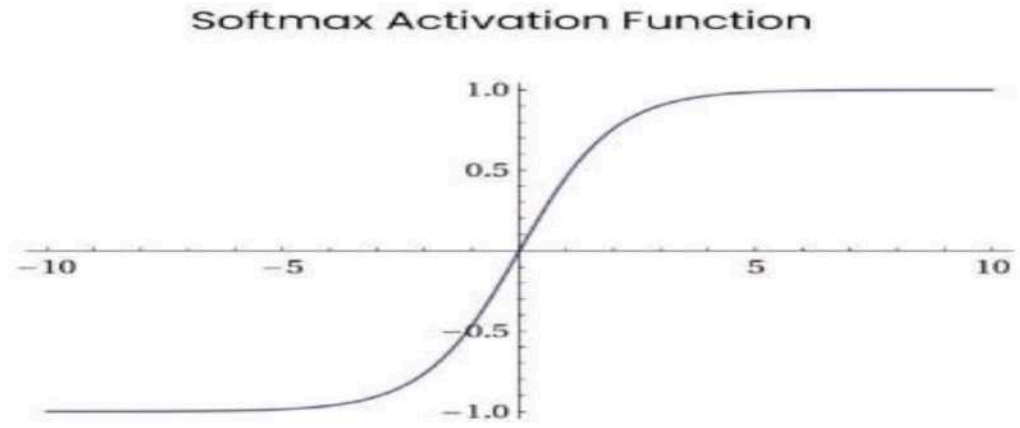


Fig.4.10.2 SoftMax Function

CHAPTER 5

REQUIREMENT ANALYSIS

REQUIREMENTS:

Operating System	:	Windows 10
RAM	:	4GB
Language	:	Python
Platform	:	PyCharm

5.1 About PyCharm :

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

Available as a cross-platform application, PyCharm is compatible with Linux, macOS, and Windows platforms. Sitting gracefully among the best Python IDEs, PyCharm provides support for both Python 2 (2.7) and Python 3 (3.5 and above) versions. PyCharm comes with a plethora of modules, packages, and tools to hasten Python development while cutting-down the effort required to do the same to a great extent, simultaneously. Further, PyCharm can be customized as per the development requirements, and personal preferences call for. It was released to the public for the very first time back in February of 2010. In addition to offering code analysis, PyCharm features:

- A graphical debugger
- An integrated unit tester
- Integration support for version control systems (VCSs)
- Support for data science with Anaconda

The main reason PyCharm for the creation of this IDE was for Python programming, and to operate across multiple platforms like Windows, Linux, and macOS. The IDE comprises code analysis tools, debugger, testing tools, and also version control options. It also assists developers in building Python plugins with the

help of various APIs available. The IDE allows us to work with several databases directly without getting it integrated with other tools. Although it is specially designed for Python, HTML, CSS, and JavaScript files can also be created with this IDE. It also comes with a beautiful user interface that can be customized according to the needs using plugins.

Installing and Setting Up PyCharm

Minimum System Requirements

Memory - 4GB

Storage Space - 2.5GB (main) + 1GB (caches)

Resolution - 1024x768
OS - 64-bit version of macOS 10.11/Microsoft Windows 7 SP1/any Linux distribution supporting Gnome, KDE, or Unity DE

Standalone Installation

Here is how to install the popular Python IDE in the traditional way:

Step 01 - Go to https://www.jetbrains.com/pycharm/download.

Step 02 - Choose a platform among Windows, Mac, and Linux.
For Windows

Step 03 - Choose the PyCharm edition; Professional (paid), or Community (free).

Step 04 - Hit the Download button to start downloading the .exe installer.

Step 05 - Once downloaded, successfully run the installer and follow the setup wizard steps.

5.2 Libraries Used:

- Open CV
- Keras

- NumPy
- Tensor Flow
- Pillow

Open CV:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e., whatever operations one can do in Numpy can be combined with OpenCV.

Keras :

Keras is a powerful deep learning library that runs on top of other open-source machine learning libraries such as TensorFlow and is also open-source itself. To develop deep learning models, Keras adopts a minimal structure in Python that makes it easier to learn and quick to write. Keras is a neural networks library written in Python that is high-level in nature – which makes it extremely simple and intuitive to use. It works as a wrapper to low-level libraries like TensorFlow or Theano high-level neural networks library, written in Python that works as a wrapper to TensorFlow or Theano. Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

NumPy :

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Applications of NumPy:

- NumPy maintains minimal memory
- An alternative for lists and arrays in Python
- Mathematical operations with NumPy
- Using NumPy for multi-dimensional arrays

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. The data API enables you to build complex input pipelines from simple, reusable pieces. TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

Pillow:

Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new images, etc. The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation. Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future

usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

CHAPTER 6

DESIGN

6.1UML Introduction:

The unified modelling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from a distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioural modeling, implementation modeling and environmental model views.

6.2 Usage of UML in Project:

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

6.3 UML Diagrams:

UML stands for Unified Modelling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying,

Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

6.3.1 Use Case Diagram:

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

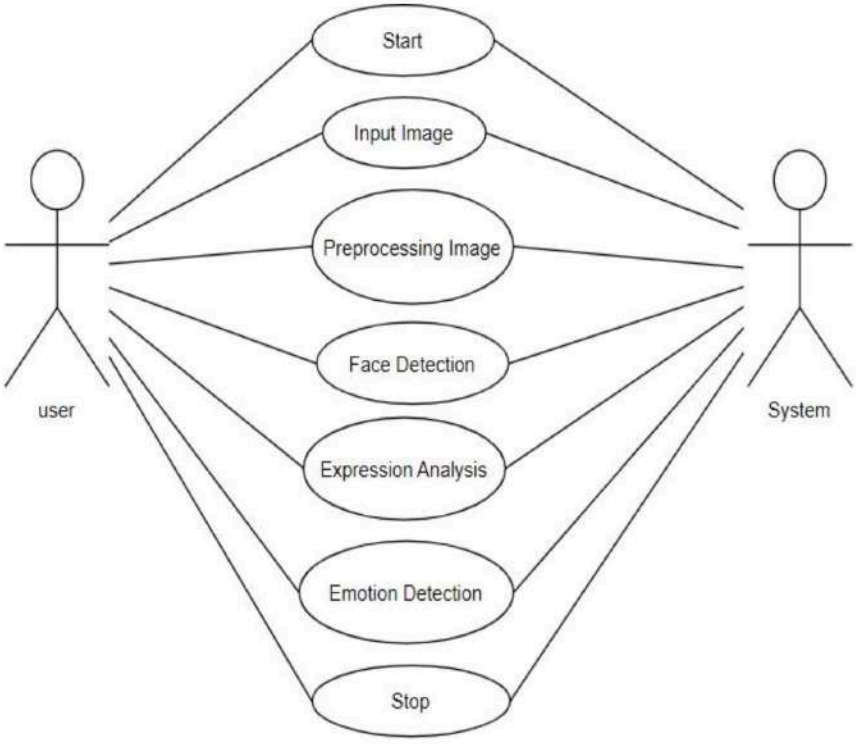


Fig 6.3.1 Use case diagram

6.3.2 Sequence diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

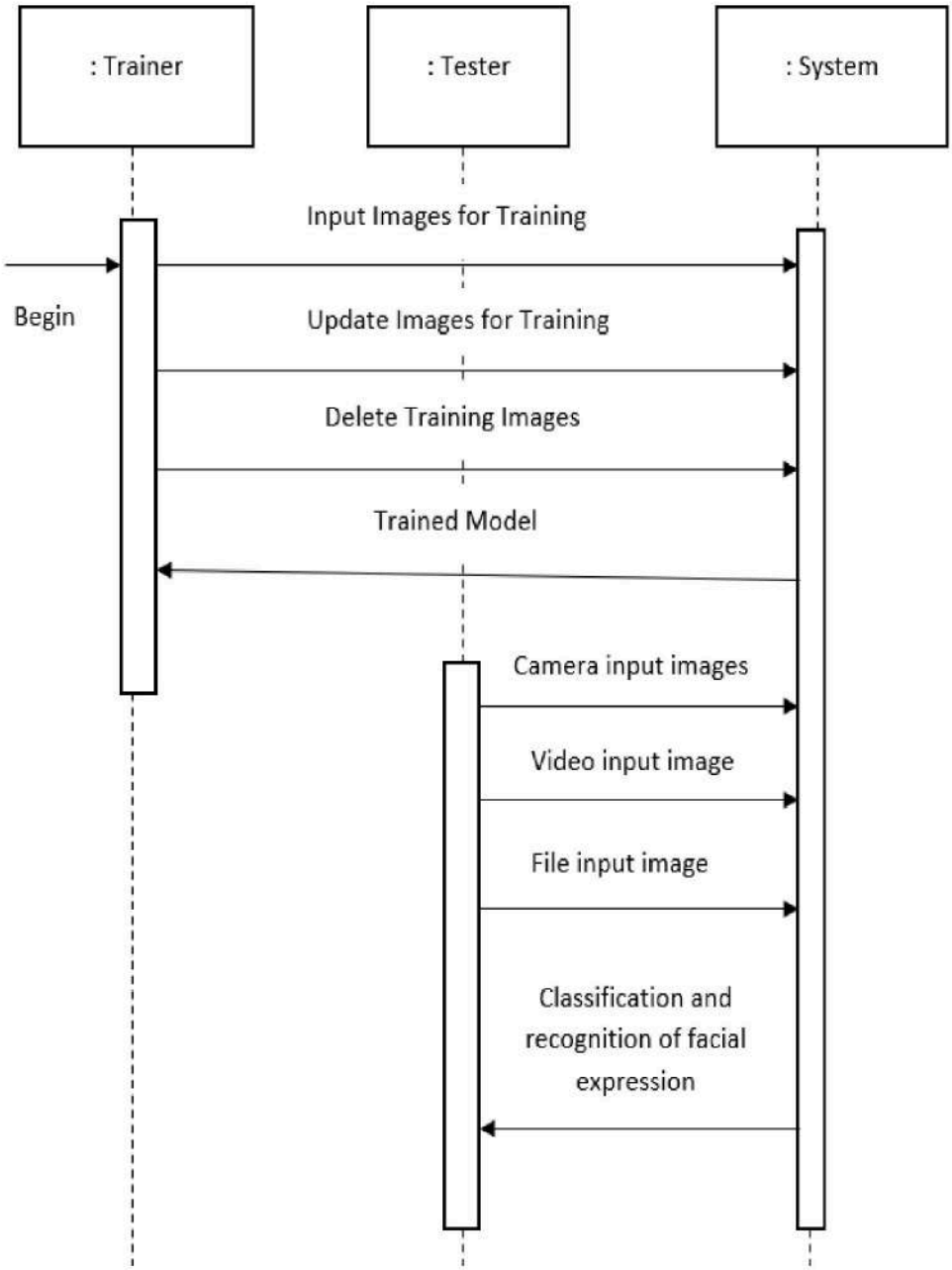


Fig 6.3.2 Sequence diagram

6.3.3 Activity diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

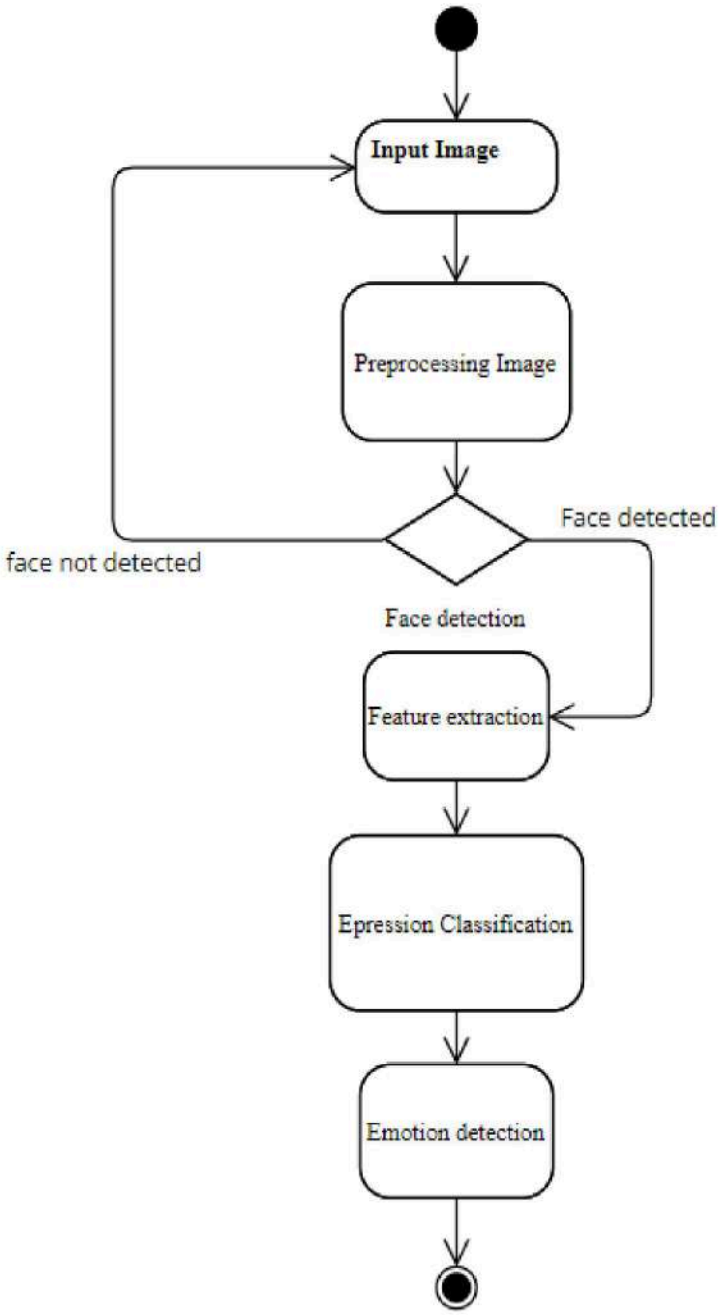


Fig 6.3.3 Activity diagram

6.3.4 Deployment Diagram:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

Fig 6.3.4 Deployment diagram

6.3.5 Data flow diagram:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

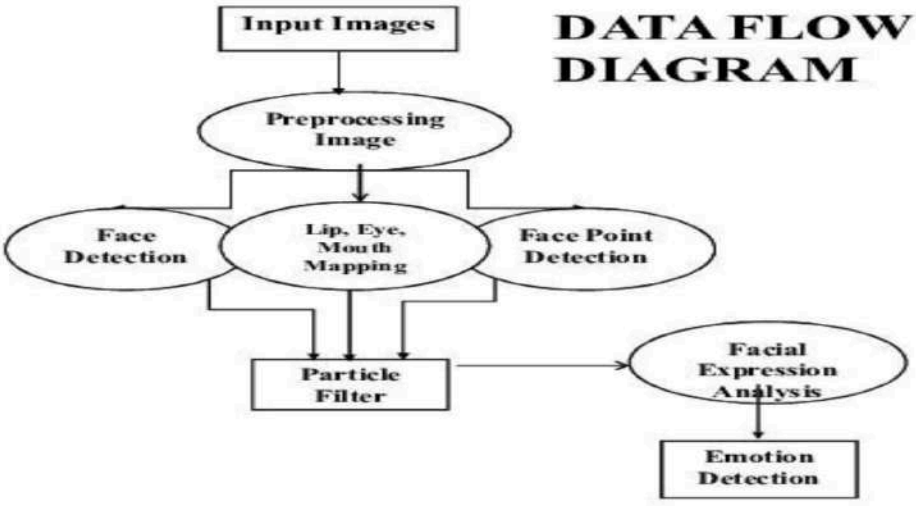


Fig 6.3.5 Data flow diagram

CHAPTER 7

IMPLEMENTATION

7.1Background and Motivation:

Computer vision (CV) is the field of study that helps computers to study using different techniques and methods so that it can capture what exists in an image or a video. There are a large number of applications of computer vision that are present today like facial recognition, driverless cars, medical diagnostics, etc. We will discuss one of the interesting applications of CV that is Emotion Detection through facial expressions. CV can recognize and tell you what your emotion is by just looking at your facial expressions. It can detect whether you are angry, happy, sad, etc. Face Detection is already implemented and now we want to detect the emotion of persons at a time . Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process. A reliable face-detection approach based on the genetic algorithm and the eigen-face technique: Firstly, the possible human eye regions are detected by testing all the valley regions in the Gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

Each possible face candidate is normalized to reduce both the lighting effect, which is caused by uneven illumination; and the shirring effect, which is due to head movement. The fitness value of each candidate is measured based on its projection on the eigen-faces. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate.

First we have to download the dataset which is suitable for the development of the model. Here the dataset we are using is the FER dataset which can be downloaded from the Kaggle. So the dataset contains the train images and test images

which are already trained and after downloading the dataset we have to train the images and also test the images while developing the model. While training the dataset we can use different epoch levels based on the system configuration. It contains 48 X 48-pixel grayscale images of the face. There are seven categories (1=Angry, 2=Disgust, 3=Fear, 4=Happy, 5=Sad, 6=Surprise, 7=Neutral) present in the data.

We are implementing the code using Python and particularly using PyCharm platform for the implementation of the code. To this we have to import all the necessary libraries that are used for the developing of the model. Here we are using or importing libraries particularly like Open CV, keras, Pillow, Numpy, Tensor flow and from those libraries we also import another modules which are necessary for that. For importing libraries we have to using particular commands like

For installing the open cv library, the command should be-pip install OpenCV

For installing the NumPy library, the command should be-pip install NumPy

For installing the keras library, the command should be-pip install keras

For installing the TensorFlow library, the command should be-pip3 install TensorFlow

For installing the pillow library, the command should be-pip install pillow

OpenCV :

Open Source Computer Vision Library is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Numpy:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. It stands for 'Numerical Python'.

Keras:

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is:

- **Simple** -- of the problem that really matter.
- **Flexible** -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.
- **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Tensor Flow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

Pillow:

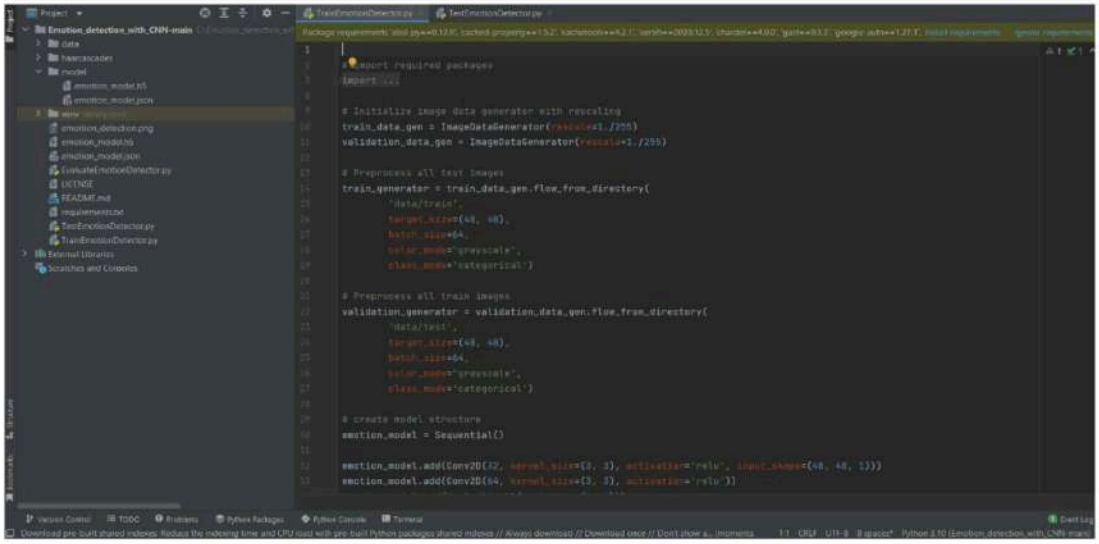
The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation. Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

In implementation of the code we have training and testing emotions for the dataset which is downloaded from the Kaggle. So first we have to train the dataset with the particular code and after successful training of the code only we have to for the testing process.

Screen 7.1 Python Terminal

7.2 Training the dataset:

We implement the code in Python Programming Language. We have done our project execution in PyCharm IDE which provides wide range of essential tools for python developers. Here, this is the environment that is set up and when imported all the files, packages , the PyCharm looks exactly like this which means we are good to go for the execution.



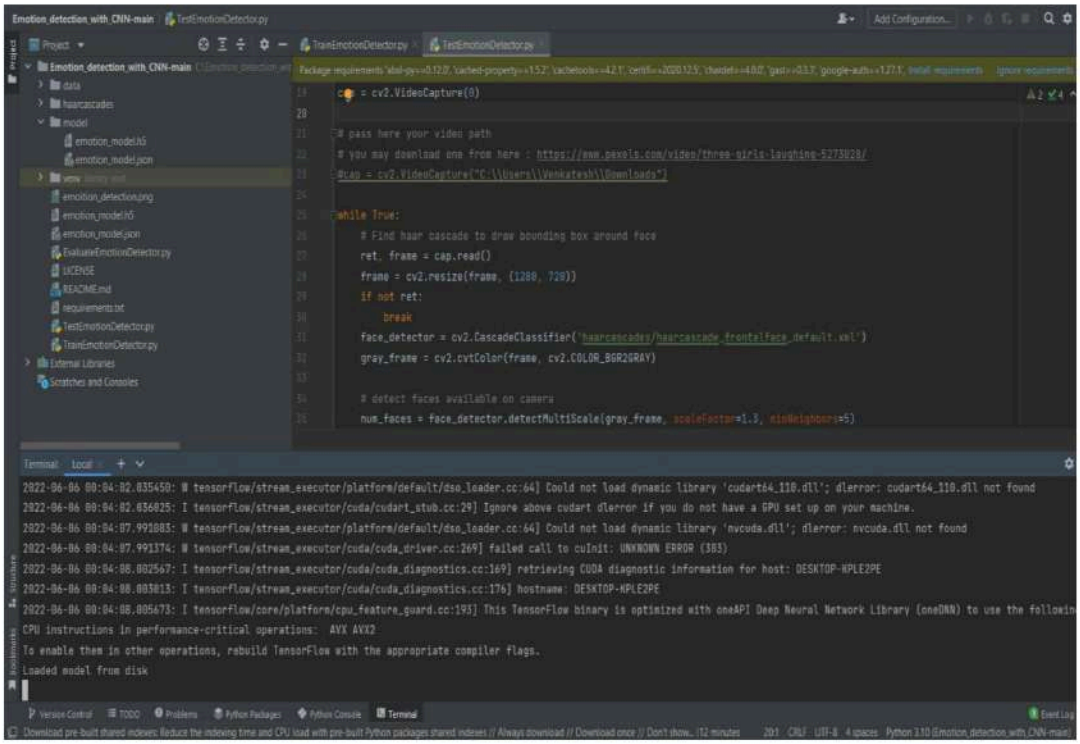
Screen 7.2 Training Model

After the successful training of the dataset we can test the dataset. After that we

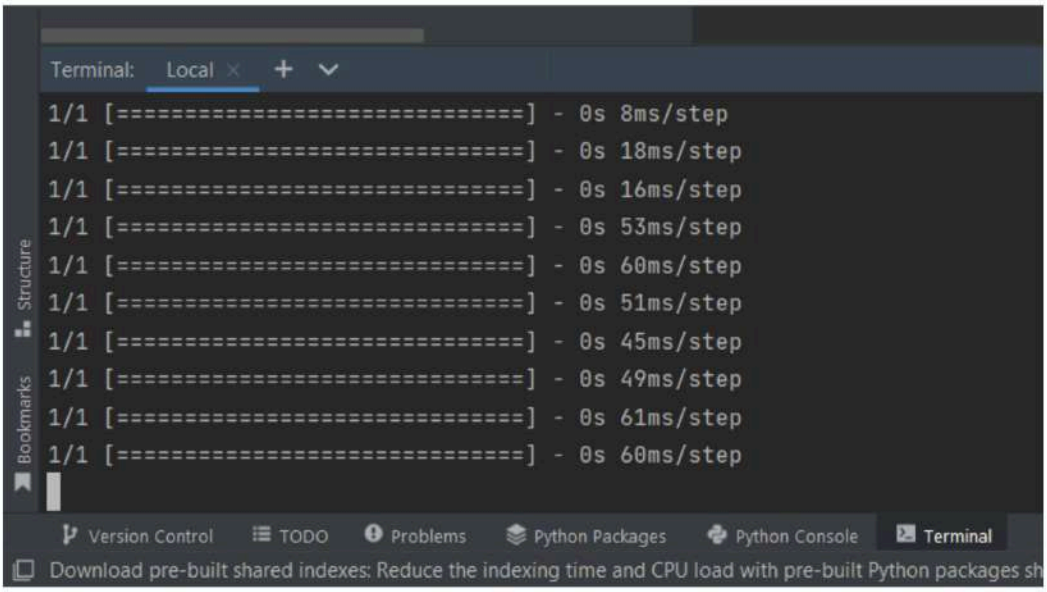
will get the suitable results which can be obtained from the testing. The command that is used for the training the dataset is the python TrainEmotionDetector.py

7.3 Testing the dataset:

After the successful completion of the training only testing will be takes place. In testing the execution results will be obtained. The command used for testing is python TestEmotionDetector.py. While running the process the terminal will be shown differently,While testing here the live webcam will be opened and the detection of the face and the recognition of the emotion will be done, when the person is in the in front of the camera. It will capture the expression or emotion of the person and will display the emotion in a particular bounding box representing the corresponding emotion (Angry, Sad, Happy, Disgust, Fear, Surprise, Neutral).

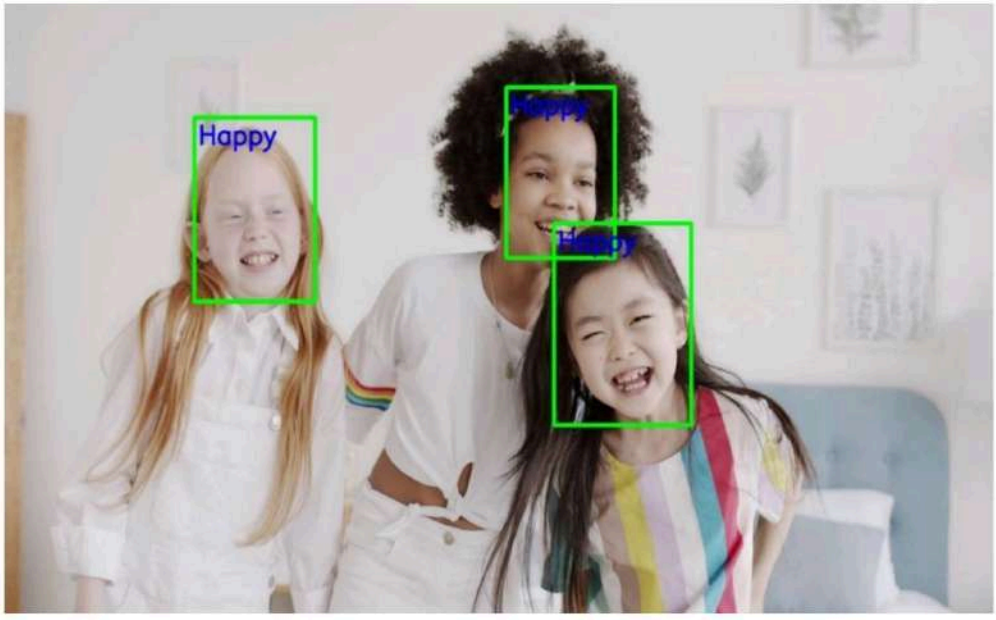


While running the test code the screen or terminal will be representing as the below.



Screen 7.3 Testing Model

After the successful execution of the testing code the result will be



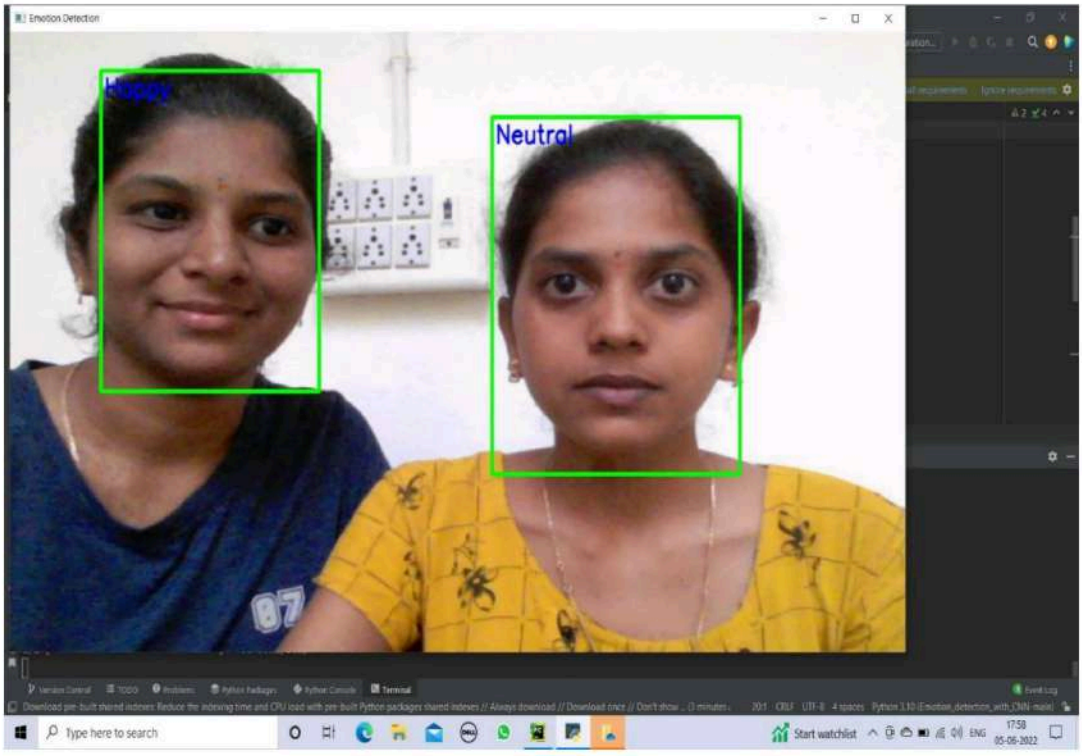
Screen 7.4 The Result

CHAPTER 8

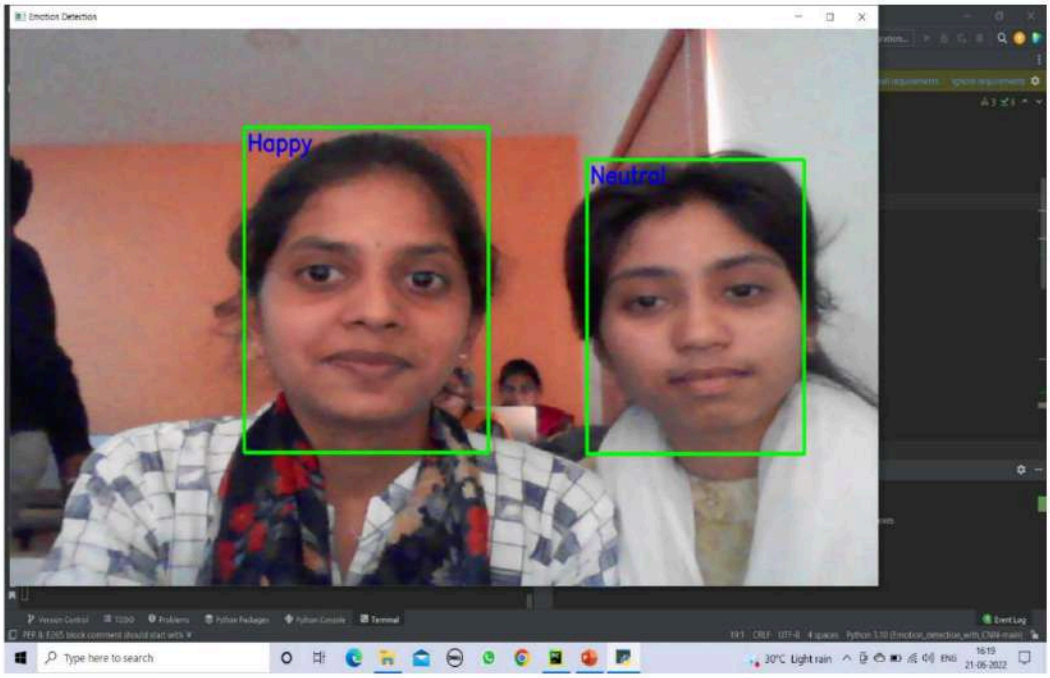
RESULT

The implementation of the proposed emotion recognition system based on facial expressions is successfully executed. This proposed system uses a python platform with different libraries that are used for the implementation of the code. Mainly, here we are using OpenCV library with many functions. It is easy to deal with the image transactions. The application imports the Open CV library to solve the computer vision problems. OpenCV library method is mainly used to detect the objects and tracking of the objects.

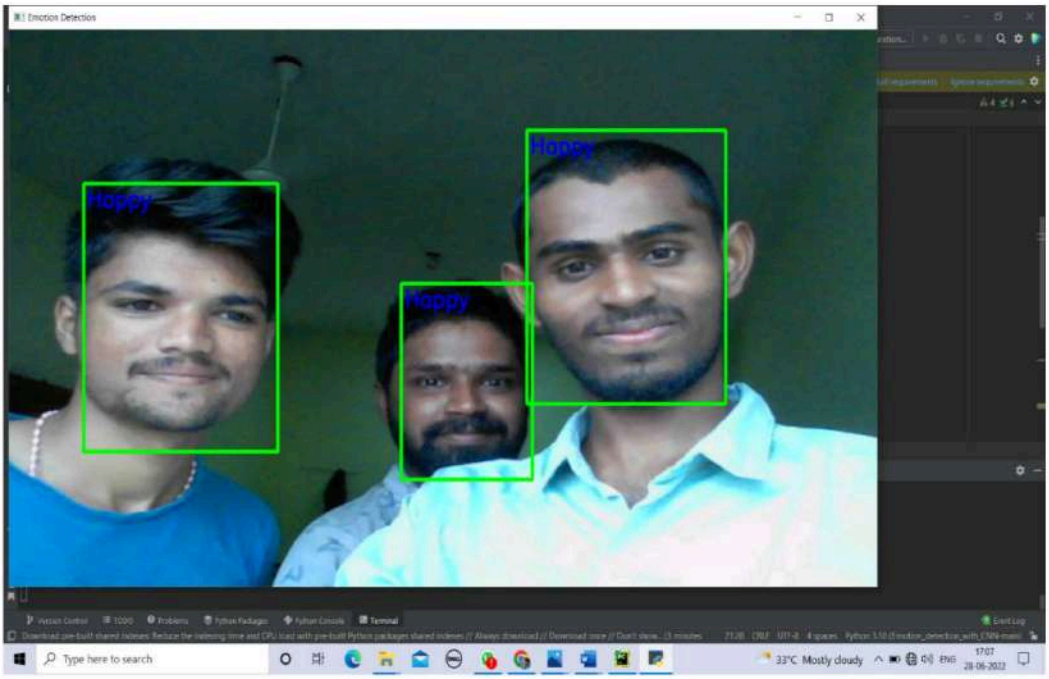
For testing the images, we used the FER dataset .This dataset contains approximately 37,000 images, with each image defined in the original expression. For the detection of the face we use algorithm like Haar-Cascade, which is widely used for face detection. Here we are trying to capture the emotions of the multiple persons in a single image at a time. The captured images of the persons or people can be obtained as below.



Screen 8.1 Output 1



Screen 8.2 Output 2



Screen 8.3 Output 3

CONCLUSION

In this we propose a facial emotion recognition method using a CNN model which extracts facial features effectively. Compared to traditional methods, the proposed method can automatically learn features. The FER2013 dataset was used and 7,074 images for emotions selected. The emotions considered were happy, sad, angry, surprise, fear, disgust and neutral. These images were converted into NumPy arrays.

A CNN model was developed with four phases where the first three phases had convolution, pooling, batch normalization and dropout layers. The final phase consists of flatten, dense and output layers. So in this as we are using CNN model for capturing the facial expressions of the persons i.e, multiple facial expressions at a time .In this facial emotion recognition project we can go through the live capture of the image or we can also give input from the system which is already present in the system.

REFERENCES

[1] "An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment".

[2] "Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure".

[3] "A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing" by Hongli Zhang, Alireza Jolfaei , and Mamoun Alazab

[4] " Facial emotion recognition using convolutional neural networks (FERC)" by Ninad Mehendale".

[5] "Facial emotion recognition using convolutional neural networks" by Ketan Sarvakar, R. Senkamalavalli , S. Raghavendra , J. Santosh Kumar , R. Manjunath, Sushma Jaiswal"