Social Emotion Detector

A Project Report

submitted in partial fulfillment of the requirements

by

Mishra Akanksha Rajeshkumar(21EC449)

Mishra Deepanshi Rajeshkumar(21EC448)

Under the Esteemed Guidance of

Jay Rathod

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the AI Saksham for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. Bhargav goradiya, Heads of the Department of Electronics and communication Engineering and Dr. Ronak Vashi, our ECE Counselor for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Mr Jay Rathod AI Saksham mentor, for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and communication engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Facial emotion recognition technology has emerged as a promising tool with applications spanning mental health, depression diagnosis, and societal well-being. This technology leverages computer vision and pattern recognition techniques to accurately identify and interpret facial expressions, of social emotion detecting insights into individuals' emotional states. In the context of mental health, facial emotion recognition holds immense potential for early detection and intervention in conditions such as depression and anxiety.

By analyzing subtle changes in facial expressions, this technology can provide objective indicators of emotional distress, complementing traditional diagnostic methods and facilitating more personalized treatment approaches. Furthermore, facial emotion recognition systems can be integrated into telemedicine platforms, enabling remote monitoring and support for individuals experiencing mental health challenges. Additionally, in societal settings, such technology can contribute to creating more empathetic and supportive environments by fostering better understanding and communication of emotions. Overall, the advancement of facial emotion recognition technology holds significant promise for improving mental health outcomes and enhancing societal well-being.

The work of this thesis aims at designing a robust Social emotion detector (SOCIAL EMOTION DETECTOR) system by combining various techniques from computer vision and pattern recognition. Expression recognition is closely related to face recognition.

TABLE OF CONTENTS

Abstract ....

List of Figures ....

Chapter 1. Introduction ....

1.1 Problem Statement ....

1.2 Problem Defination ....

1.3 Expected Output ....

Chapter 2. Literature Survey ....

2.1 Face Recognition : A Litrature Review...

2.1.1 Brief Introduction of Paper ....

2.1.1 Technique used in paper ....

Chapter 3. Proposed Methodology ....

3.1 Existing System...

3.2 System Functionality ....

3.3 Library and Packages...

3.4 Requirement Specification...

Chapter 4. Software Development Methodology ....

4.1. Description of Diagram ....

4.2. System Design ....

4.3. System Flowchart ....

4.4. Social Emotion Detection using CNN..

4.4. Architecture Design...

4.4. CNN Methodology...

Chapter 5. Implementation and Result ....

5.1. Using Open CV haarcascade xml ....

5.2. Testing the Model...

5.3. Application ....

5.4 Out[put Result...

Chapter 6. Conclusion ....

6.1. Advantage ....

6.2. Scope……………………………………………………………………………………………………

Video Link…….

Reference…. ………………………………………………………………………………………………………

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	Expected output	8
3.1	Classification of seven basic emotion	13
4.1	Block Diagram	20
4.2(a)	System architechture	21
4.2(b)	Process Flow	22
4.5 4.6	Haar Features (A) Architecture Diagram (B) Architecture Diagram of CNN	27
4.6.2 & 4.6.3	Important Libraries and package	31
4.6.4	Training and Validation generator (A) Train (B) Validation	33
5.1	Detected Facial Emotion	37

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. Problem Statement:

Face recognition is important for the interpretation of facial expressions in applications such as intelligent, man-machine interface and communication, intelligent visual surveillance, teleconsocial emotion detectorence and real-time animation from live motion images.

The facial expressions are useful for efficient interaction. Most research and system in social emotion detector are limited to six basic expressions (joy, sad, anger, disgust, fear, surprise).

It is found that it is insufficient to describe all facial expressions and these expressions are 3 categorized based on facial actions.

Detecting face and recognizing the facial expression is a very complicated task when it is a vital to pay attention to primary components like: face configuration, orientation, location where the face is set.

1.2. Problem Definition:

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.

Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Health care providers can provide better service by using additional information about patient‟s emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content. Humans are well - trained in reading the

emotions of others, in fact, at just 14 months old, babies can already tell the difsocial emotion detectorence between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, We designed a deep learning neural.

1.3. Expected Outcomes:

In this deep learning project, we have built a convolution neural network to recognize facial emotions. We have trained our model on the dataset.

Then we are mapping those emotions with the corresponding text .

We are using OpenCV's haar cascade xml we are getting the bounding box of the faces in the webcam. Then we feed these boxes to the trained model for classification.

Angry surprise sad fear happy neutral

FIG 1.1: Expected outcomes

The emotions are correctly imposed on their corresponding faces. The input will be raw image of the expression, and output will be shown as above fig 1.1

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1. Face Recognition: A Literature Review

2.1.1. Brief Introduction of Paper:

1 Face recognition technology has become an essential aspect of modern security and authentication systems, with applications ranging from surveillance to personal device access.
2 The development of face recognition methods has seen significant advancements since its inception in the 1960s.
3 This paper provides a comprehensive literature review of face recognition, tracing its evolution from early traditional methods such as Eigen faces and Fisher faces to contemporary deep learning techniques like Convolutional Neural Networks (CNNs).
4 The review highlights key milestones, compares various methodologies, discusses the challenges faced by current systems, and explores potential future directions in the field.
5 Through this examination, the paper aims to provide a clear understanding of the progress and current state of face recognition technology.

2.1.2. Techniques used in Paper:

Traditional Methods

1. **Eigen faces (Principal Component Analysis - PCA)**
   a. Introduced by Turk and Pentland in 1991.
   b. Uses PCA to reduce the dimensionality of face images.
   c. Transforms face images into a set of orthogonal basis vectors (eigen faces).
   d. Recognition is performed by projecting new images into the subspace and comparing distances to known faces.

2. **Fisher faces (Linear Discriminant Analysis - LDA)**
   a. Proposed by Belhumeur et al. in 1997.
   b. Utilizes LDA to improve discriminative power.

c. Maximizes the ratio of between-class variance to within-class variance.
d. Of social emotion detectors better performance under varying lighting conditions and facial expressions.

Modern Approaches

1. **Convolutional Neural Networks (CNNs)**
    a. Automate feature extraction and learn complex representations from large datasets.
    b. Key architectures include VGG-Face, Face Net, and Deep Face.
    c. **VGG-Face**
        i. Developed by the Visual Geometry Group at Oxford.
        ii. Employs a deep CNN architecture for high accuracy.
        iii. Trained on large datasets and fine-tuned for face identification and verification.
    d. **Face Net**
        i. Proposed by Google.
        ii. Uses a triplet loss function to learn a compact and discriminative embedding.
        iii. Maps faces to a Euclidean space where distances correspond to face similarity.
    e. **Deep Face**
        i. Developed by Facebook.
        ii. Leverages a deep neural network for robust face representations.
        iii. Combines 3D alignment, a nine-layer neural network, and a large-scale training dataset to achieve near-human accuracy.

2. These technologies represent the primary methods discussed in the paper, illustrating the evolution from traditional statistical approaches to advanced deep learning techniques in the field of face recognition.

CHAPTER 3

PROPOSED METHODOLOGY

CHAPTER 3

PROPOSED METHODOLOGY

3.1 EXISTING SYSTEM

The compactness of emotions reduces the effort of input to express not only emotions, but also serves to adjust message tone, increase message engagement, manage conversations and maintain social relationships.

Moreover, emotions do not have language barriers, making it possible for users across countries and cultural backgrounds to communicate.

. As validation of the usefulness of mapping emotions to emotions, preliminary investigations reported by Jaeger et al. suggest that emotions may have potential as a method for direct measurement of emotional associations to foods and beverages.

FIG 3.1 CLASSIFICATION OF SEVEN BASIC EMOTION

Facial Emotion Recognition has become an increasingly researched topic in recent years, mainly because it has a lot of applications in the fields of Computer Vision, robotics, and Human Computer Interaction.

In a study on the facial recognition technology (SOCIAL EMOTION DETECTORET) dataset, Paul Ekam has presented 7 universal expressions (anger, fearful, happy, sad, neutral, disgust, and surprise) with the positioning of faces, and the muscular movements required to create these expressions in his study.

3.2 SYSTEM FUNCTIONALITIES

Deep learning is currently a hot topic in Machine Learning and is also an essential tool in the field of Artificial Intelligence. Deep Learning is good at doing image recognition, object detection, natural language translation, trend prediction. Emotions is a visual symbol widely used in wireless communication. It includes faces, hand gestures, animals, human figures, and signs. Instead of typographic, emotions are actual pictures.

As there are increasing uses of emotions, the needs for an instant generator of emotions have become urgent. With the promising results of image recognition tasks by some deep learning models, we propose a real-time emotions generator, which enables the user to generate an emotions given a corresponding human facial expression. In this work, we applied deep learning models to perform feature extraction and pre-processed our data with data augmentation strategies and morphology operations.

Our model uses the model of the convolutional neural network. It consists of convolutional layers, ReLu layers, and max-pooling layers. For our model, we have two sets of convolutional layers, ReLu layers, and max-pooling layers. After that, we put a fully-connected layer, a ReLu layer, and a softmax layer to predict the label.

Social emotion detector is a process performed by humans or computers, which consists of:

1 Locating faces in the scene (e.g., in an image; this step is also re-social emotion detectorred to as face detection),
2 Extracting facial features from the detected face region (e.g., detecting the shape of facial components or describing the texture of the skin in a facial area; this step is re-social emotion detected to as facial feature extraction),
3 Analyzing the motion of facial features and/or the changes in the appearance of facial features and classifying this information into some facial-expression interpretation categories such as facial muscle activations

like smile or frown, emotion (affect) categories like happiness or anger, attitude categories like (dis)liking or ambivalence, etc.(this step is also resocial emotion detectorred to as facial expression interpretation).

4 Converting the detected facial expression into emotions by mapping the predicted results. Several Projects have already been done in this field and our goal will not only be to develop an Automatic Facial Expression Emotions Generation System but also improving the accuracy of this system compared to the other available system

3.3 LIBRARY AND PACKAGES

1 **OpenCV**: 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 commercial products.

Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people in the user community and an estimated number of downloads exceeding 14 million.

The library is used extensively in companies, research groups and by governmental bodies. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available.

A full-featured CUDA and OpenCL interface is being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

OpenCV's application areas include : - 2D and 3D feature toolkits - Facial recognition system - Gesture recognition - Human–computer interaction (HCI) - Object identification - Stereopsis stereo vision:

Depth perception from 2 cameras - Motion tracking - Augmented reality To support some of the above areas, OpenCV includes a statistical machine learning library that contains : - Decision tree learning - k-nearest neighbor algorithm - Naive Bayes classifier - Artificial neural networks - Random forest Random forest - Support vector machine (SVM) - Deep neural networks (DNN)

2. ● **Numpy** : NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines.

   Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays.

   Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays. It is the fundamental package for scientific computing with Python.

   It contains various features including these important ones: - A powerful N-dimensional array object - Sophisticated (broadcasting) functions - Tools for integrating C/C++ and Fortran code - Useful linear algebra, Fourier Transform, and random number capabilities

3. **Keras:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

   Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier.

   The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU).

4. SciPy : SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others. NumPy is based on two earlier Python modules dealing with arrays.

   One of these is Numeric. Numeric is like NumPy, a Python module for high-performance, numeric computing, but it is obsolete nowadays.

   Another predecessor of NumPy is Numarray, which is a complete rewrite of Numeric but is deprecated as well. NumPy is a merger of those two, i.e. it is built on the code of Numeric and the features of Numarray 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 ofTensorFlow.

5. **Haar Cascade Classifier in OpenCv** : Haar feature-based cascade classifiers is an effectual machine learning based approach, in which a cascade function is trained using a sample that contains a lot of positive and negative images.

The outcome of AdaBoost classifiers is that the strong classifiers are divided into stages to form cascade classifiers.

The term “cascade” means that the classifier thus produced consists of a set of simpler classifiers which are applied to the region of interest until the selected object is discarded or passed.

The cascade classifier splits the classification work into two stages: training and detection. The training stage does the work of gathering the samples which can be classified as positive and negative.

3.3 REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. The requirements are split into three categories, namely:

Software Requirements

The basic software requirements to run the program are:

1. Microsoft Windows XP, Windows 7, Windows 10
2. Python
3. IDE e.g. Google Colab, Anaconda Navigator, Jupyter.
4. Browser e.g. Mozilla Firefox, Safari, chrome, etc. v. TensorFlow, openCV.

Data set Requirements

The dataset required for training the Model is:

1 SOCIAL EMOTION DETECTOR from kaggle

Hardware Requirements

The basic hardware required to run the program are:

1 Hard disk of 500 GB or higher.
2 System memory (RAM) of 4GB or higher.
3 I3 processor-based computer or higher.
4 Web Camera.

Software Interfaces

1. Microsoft Word 2007
2. Dataset Storage : Microsoft Excel

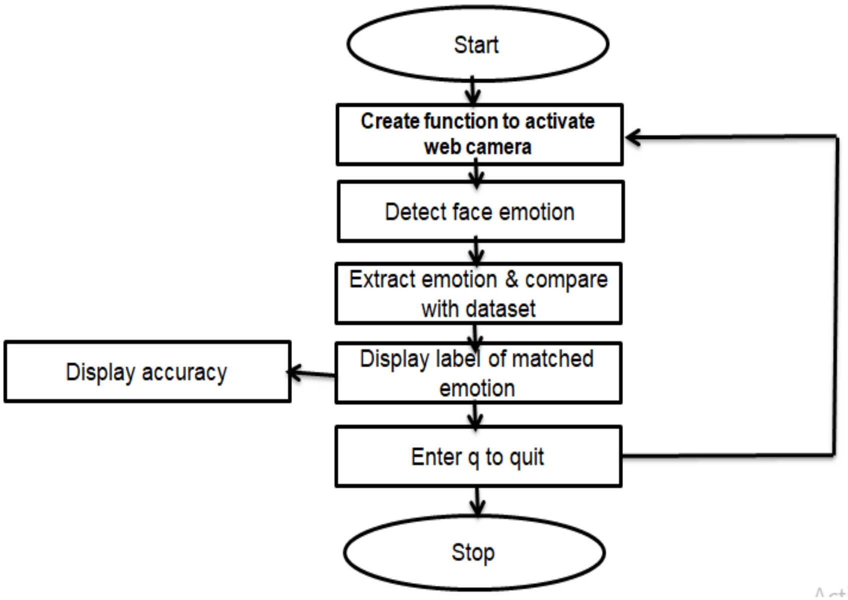
CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

4.1 DESCRIPTION OF DIAGRAM



In the Fig 4.1, Facial expressions can be described as the arrangement of facial muscles to convey a certain emotional state to the observer in simple words. Emotions can be divided into six broad categories—Anger, Disgust, Fear, Happy, Sad, Surprise, and Neutral. In this, train a model to difsocial emotion detectiate between these, train a convolutional neural network using the SOCIAL EMOTION DETECTOR dataset and will use various hyper-parameters to fine-tune the model.

The design starts with the initializing CNN model by taking an input image (static or dynamic) by adding a convolution layer, pooling layer, flatten layers, and dense layers. Convolution layers will be added for better accuracy for large datasets. The dataset is collected from a CSV file (in pixel format) and it's converted into images and then classify emotions with respective expressions.

4.2 SYSTEM DESIGN

System design shows the overall design of the system. In this section we discuss in detail the design aspects of the system

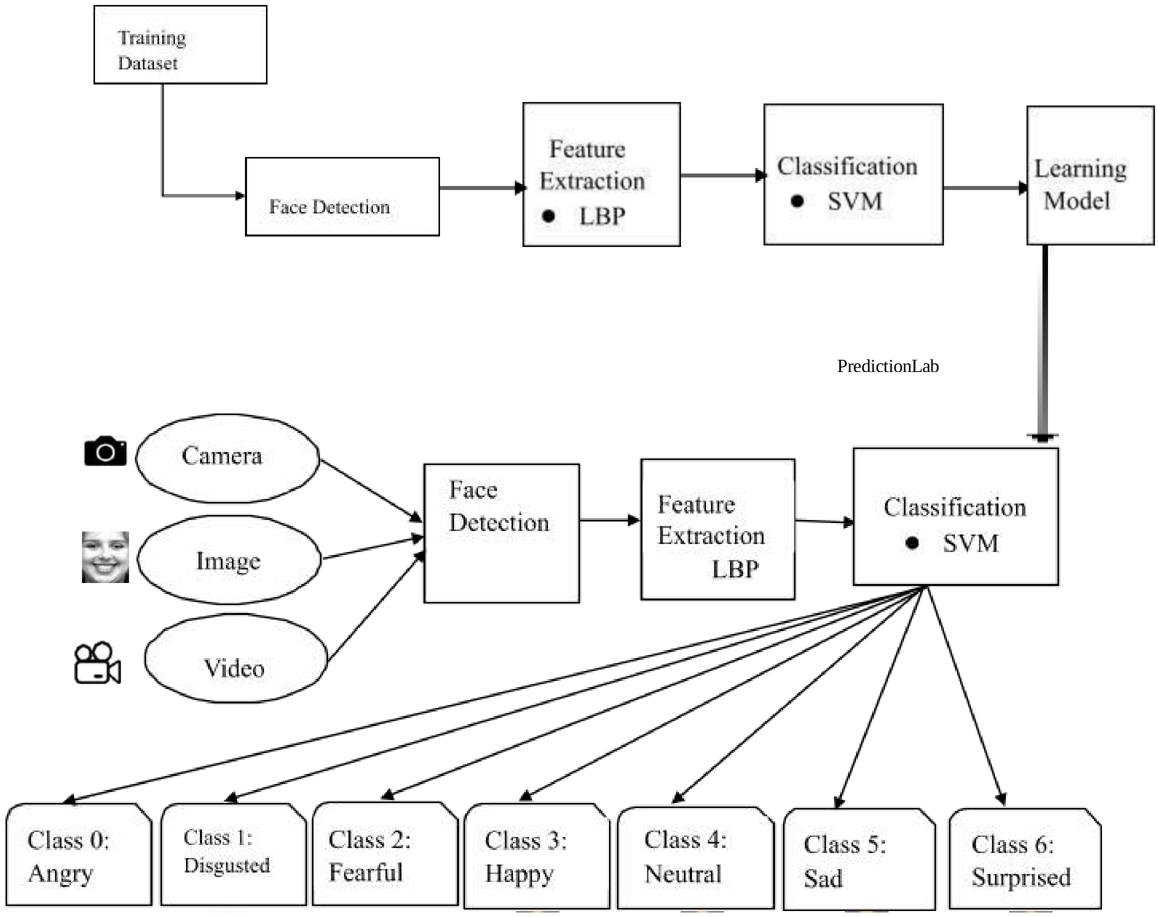


Fig 4.2(a)

Here emotions are classified as happy, sad, angry, surprise, neutral, disgust, and fear with 34,488 images for the training dataset and 1,250 for testing. Each emotion is expressed with difsocial emotion detectorent facial features like eyebrows, opening the mouth, Raised cheeks, wrinkles around the nose, wide-open eyelids and many others.Trained the large dataset for better accuracy and result that is the object class for an input image. Based on those features it performs convolution layers and max pooling. These are the seven difsocial emotion detectorent universal emotions with the following expressions above Fig 4.2

4.3 SYSTEM FLOWCHART

The social emotion detector system is implemented using convolutional neural networks. The block diagram of the system is shown in following figures:

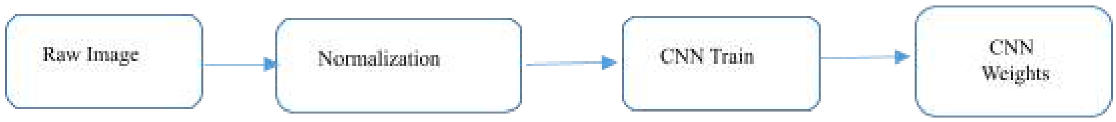


Fig 4.3(a) Flowchart of Training Phase

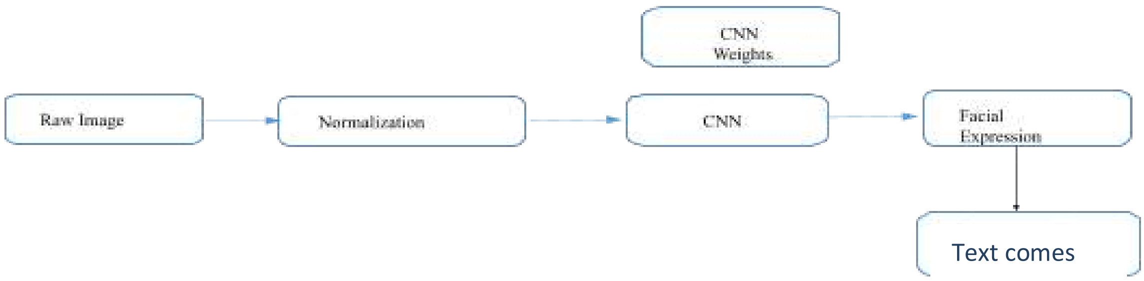
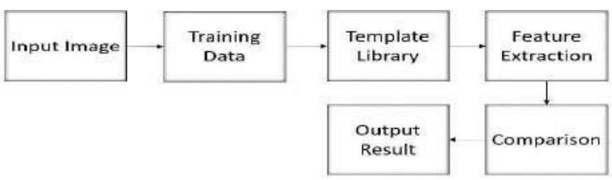


Fig 4.2(b) Flowchart of Testing Phase

During training, the system receives training data comprising grayscale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, validation dataset is used to choose the final best set of weights out of a set of training exercises performed with samples presented in difsocial emotion detectorent orders. The output of the training step is a set of weights that achieve the best result with the training data. During the test, the system received a grayscale image of a face from the test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a single number that represents one of the



seven basic expressions.

The original structure contains six steps which are input image, training data, template library, feature extraction, comparison and output result, as shown in Figure 4.3(c).

However, a simplified structure that is used in this paper only has four steps after we combine the step of template library, feature extraction and comparison to social emotion detector, as shown in Figure above. It will greatly increase the efficiency and reduce the running time

4.4 FACIAL EMOTION RECOGNITION USING CNN METHODOLOGY

4.4.1 Dataset The dataset from a Kaggle Social emotion detector Challenge (SOCIAL EMOTION DETECTOR) is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has a training set of 35887 facial images with facial expression labels..

The dataset has class imbalance issues, since some classes have a large number of examples while some have few. The dataset is balanced using oversampling, by increasing numbers in minority classes.

The images in the SOCIAL EMOTION DETECTOR dataset have size 48x48 and are black and white images. The SOCIAL EMOTION DETECTOR dataset contains images that vary in viewpoint, lighting, and scale. Fig.4.4(b) shows some sample images from the SOCIAL EMOTION DETECTOR dataset, and Table 4.4(c) illustrates the description of the dataset

Fig. 4.3 Sample images from the SOCIAL EMOTION DETECTOR dataset

4.4.2 Process of Social emotion detector

The process of SOCIAL EMOTION DETECTOR has three stages. The preprocessing stage consists of preparing the dataset into a form which will work on a generalized algorithm and generate efficient results. In the face detection stage, the face is detected from the images that are captured real time. The emotion classification step consists of implementing the CNN algorithm to classify input images into one of seven classes. These stages are described using in a flowchart in Fig. 2:

Fig.4.4 Process flow

4.4.3 Pre-processing

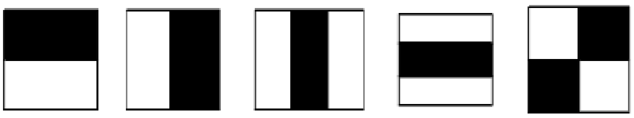
The input image to the SOCIAL EMOTION DETECTOR may contain noise and have variation in illumination, size, and color. To get accurate and faster results on the algorithm, some preprocessing operations were done on the image. The preprocessing strategies used are conversion of image to grayscale, normalization, and resizing of image.

1. Normalization - Normalization of an image is done to remove illumination variations and obtain improved face image
2. Grayscaling - Grayscaling is the process of converting a colored image input into an image whose pixel value depends on the intensity of light on the image. Grayscaling is done as colored images are difficult to process by an algorithm.

3. Resizing - The image is resized to remove the unnecessary parts of the image. This reduces the memory required and increases computation speed.

4.4.4 Face Detection

Face detection is the primary step for any SOCIAL EMOTION DETECTOR system. For face detection, Haar cascades were used (Viola & Jones, 2001). The Haar cascades, also known as the Viola Jones detectors, are classifiers which detect an object in an image or video for which they have been trained. They are trained over a set of positive and negative facial images. Haar cascades have proved to be an efficient means of object detection in images and provide high accuracy. Haar features detect three dark regions on the face, for example the eyebrows. The computer is trained to detect two dark regions on the face, and their location is decided using fast pixel calculation. Haar cascades successfully remove the unrequired background data from the image and detect the facial region from the image. The face detection process using the Haar cascade classifiers was implemented in OpenCV. This method was originally proposed by Papageorgiou et al, using rectangular features which are shown in figure 3



4.4.5 Emotion Classification

In this step, the system classifies the image into one of the seven universal expressions - Happiness, Sadness, Anger, Surprise, Disgust, Fear, and Neutral as labelled in the SOCIAL EMOTION DETECTOR dataset.

The training was done using CNN, which is a category of neural networks proved to be productive in image processing. The dataset was first split into training and test datasets, and then it was trained on the training set. Feature extraction process was not done on the data before feeding it into CNN.

The approach followed was to experiment with difsocial emotion detectorent architectures on the CNN, to achieve better accuracy with the validation set, with minimum overfitting. The emotion classification step consists of the following phases:

1. Splitting of Data: The dataset was split into 3 categories according to the "Usage" label in the SOCIAL EMOTION DETECTOR dataset: Training, PublicTest, and PrivateTest. The Training and PublicTest set were used for generation of a model, and the PrivateTest set was used for evaluating the model.
2. Training and Generation of model: The neural network architecture consists of the following layers:
   a Convolution Layer: In the convolution layer, a randomly instantiated learnable filter is slid, or convolved over the input. The operation performs the dot product between the filter and each local region of the input. The output is a 3D volume of multiple filters, also called the feature map.
   b Max Pooling: The pooling layer is used to reduce the spatial size of the input layer to lower the size of input and the computation cost.
   c Fully connected layer: In the fully connected layer, each neuron from the previous layer is connected to the output neurons. The size of the final output layer is equal to the number of classes in which the input image is to be classified.
   d Activation function: Activation functions are used to reduce the overfitting. In the CNN architecture, the ReLu activation function has been used. The advantage of the ReLu activation function is that its gradient is always equal to 1, which means that most of the error is passed back during back-propagation .
   e $f(x) = \max(0, x)$
   f Equation 1: Equation of ReLu Activation Function
   g Softmax: The softmax function takes a vector of N real numbers and normalizes that vector into a range of values between (0, 1).
   h Batch Normalization: The batch normalizer speeds up the training process and applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

3. Evaluation of model: The model generated during the training phase was then evaluated on the validation set, which consisted of 3589 images.
4. Using model to classify real time images: The concept of transsocial emotion detector learning can be used to detect emotion in images captured in real time. The model generated during the training process consists of pretrained weights and values, which can be used for implementation of a new facial expression detection problem. As the model generated already contains weights, SOCIAL EMOTION DETECTOR becomes faster for real time images. 4

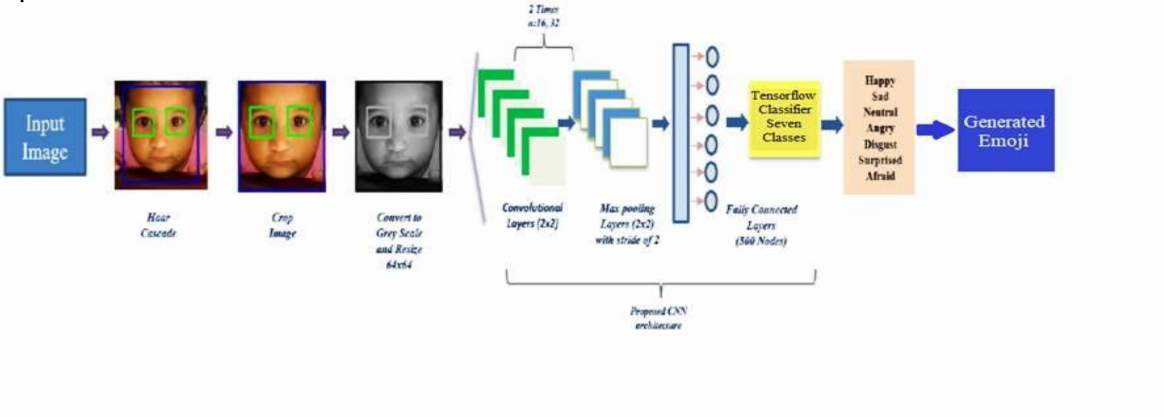


Fig 4.5 Architecture Diagram of Emotions Generation from SOCIAL EMOTION

4.5 ARCHITECTURAL DESIGN

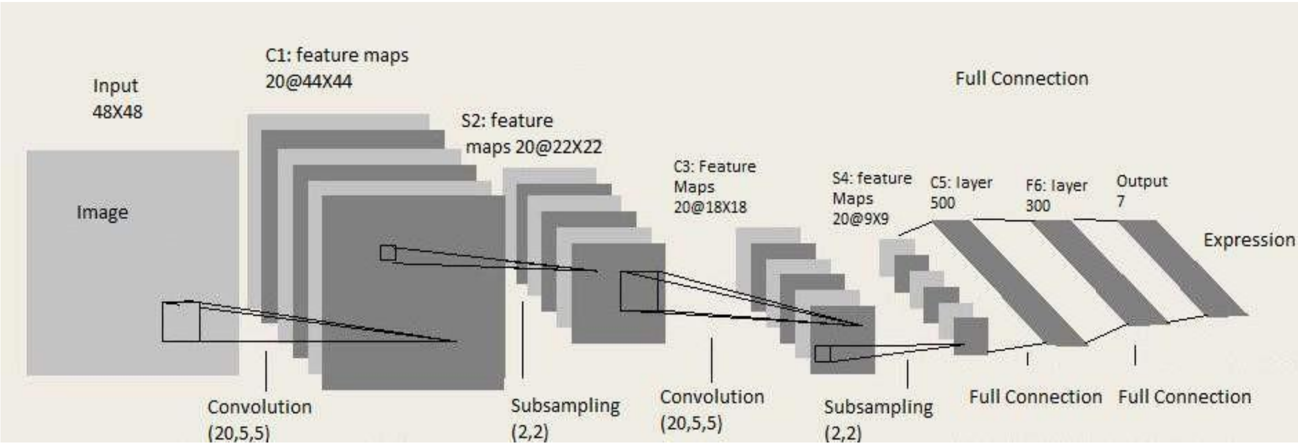


Fig 4.6: Architecture of CNN

A typical architecture of a convolutional neural network contains an input layer, some convolutional layers, some fully-connected layers, and an output layer. CNN is designed with some modification on LeNet Architecture.

It has 6 layers without considering input and output. The architecture of the Convolution Neural Network used in the project is shown in the above figure 4.5(b).

1. **Input Layer**: The input layer has predetermined, 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 proposed laptop webcam images are also used, in which face is detected and cropped using OpenCV Haar Cascade Classifier and normalized.

2. **Convolution and Pooling (ConvPool) Layers:** Convolution and pooling is done based on batch processing. Each batch has N images and CNN filter weights are updated on those batches.

Each convolution layer takes image batch input of four dimension N x Color-Channel x width x height. Feature maps or filters for convolution are also four dimensional (Number of feature maps in, number of feature maps out, filter width, filter height). In each convolution layer, four dimensional convolution is calculated between image batch and feature maps. After convolution only the parameter that changes is image width and height.
New image width = old image width – filter width +1 New image height = old image height – filter height + 1

After each convolution layer down-sampling / sub-sampling is done for dimensionality reduction. This process is called Pooling. Max pooling and Average Pooling are two famous pooling methods. In this project max pooling is done after convolution. Pool size of (2 x 2) is taken, which splits the image into a grid of blocks each of size (2 x 2) and takes a maximum of 4 pixels. After pooling only height and width are affected.

Two convolution layers and a pooling layer are used in the architecture. The first convolution layer size of the input image batch is (N x 1 x 48 x 48). Here, the size of the image batch is N, number of color channels is 1 and both image height and width are 48 pixels. Convolution with a feature map of (1 x 20 x 5 x 5) results in an image batch of size (N x 20 x 44 x 44). After convolution pooling is done with a pool size of (2 x 2), which results in an image batch of size (N x 20 x 22 x 22). This is followed by a second convolution layer with a feature map of 20x20x5x5, which results in an image batch of size (N x 20 x 18 x 18). This is followed by a pooling layer with pool size (2 x 2), which results in an image batch of size (N x 20 x 9 x 9).

3. **Fully Connected Layer**: This layer is inspired by the way neurons transmit signals through the brain. It takes a large number of input features and transforms features through layers connected with trainable weights. Two hidden layers of size 500 and 300 units are used in fully-connected layers. The weights of these layers are trained by forward propagation of training data then backward propagation of its errors.

Back propagation starts from evaluating the difsocial emotion detectorence between prediction and true value, and back calculates the weight adjustment needed to every layer before. We can control the training speed and the complexity of the architecture by tuning the hyper-parameters, such as learning rate and network density. Hyper-parameters for this layer include learning rate, momentum, regularization parameter, and decay. The output from the second pooling layer is of size Nx20x9x9 and input of the first hidden layer of fully-connected layer is of size Nx500. So, the output of the pooling layer is flattened to Nx1620 size and fed to the first hidden layer.

Output from the first hidden layer is fed to the second hidden layer. Second hidden layer is of size Nx300 and its output is fed to the output layer of size equal to the number of facial expression classes.

4. **Output Layer**: Output from the second hidden layer is connected to the output layer having seven distinct classes. Using Softmax activation function, output is obtained using the probabilities for each of the seven classes. The class with the highest probability is the predicted class.

4.6 CNN METHODOLOGY

CNN architecture for social emotion detector as mentioned above was implemented in Python. Along with Python programming language, Numpy, Keras, CV2 libraries are used. In the below steps will build a convolution neural network architecture and train the model on SOCIAL EMOTION DETECTOR dataset for Emotion recognition from images.

4.6.1 Data Preprocessing

The dataset consists of 48*48 pixel gray scale images of faces. The task was to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). Data set contains two columns, “emotion” and “pixels”.

The “emotion” column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The “pixels” column contains a string surrounded in quotes for each image. The contents of this string are space-separated pixel values in row major order. test.csv contains only the “pixels” column and your task is to predict the emotion.

4.6.2 Importing necessary libraries and packages

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
```

4.6.3 Initialize training and validation generator

```
epochs = 10

from keras.callbacks import ModelCheckpoint

checkpoint = ModelCheckpoint("model_weights.h5", monitor='val_acc', verbose=1, save_best_
callbacks_list = [checkpoint]

history = model.fit_generator(generator=train_generator,
                                steps_per_epoch=train_generator.n//train_generator.batch_
                                epochs=epochs,
                                validation_data = validation_generator,
                                validation_steps = validation_generator.n//validation_ger
                                callbacks=callbacks_list
                                )
```

```
225/225 [==============================] - 1232s 5s/step - loss: 1.4092 - acc: 0.4609 - val_loss:
1.3140 - val_acc: 0.4996

Epoch 00009: val_acc improved from 0.47766 to 0.49957, saving model to model_weights.h5
Epoch 10/10
225/225 [==============================] - 1233s 5s/step - loss: 1.3671 - acc: 0.4768 - val_loss:
1.3597 - val_acc: 0.4935

Epoch 00010: val_acc did not improve from 0.49957
```

The pixels have been scaled between -1 and 1. The dataset has been split into train and validation sets.

4.6.4 Build the model using Convolutional Neural Networks(CNN)

The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data. Central to the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation called a convolution.

The layers to be added are:

- Convolution layer
- Pooling layer
- Batch normalization
- Activation Layer
- Dropout Layer
- Flatten Layer
- Dense layer

Convolution is a linear operation that involves the multiplication of a set of weights with the input, much like a traditional neural network. Given that the technique was designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel.

Batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers. 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(ReLu) function 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.

A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. In Data Augmentation we take each batch and apply some series of random transformations (random rotation, resizing, shearing) to increase generalizability of the model. 4.6.5 Compile and train the model Now compile the model with any optimizer and any loss.I have used Adam optimizer and categorical_crossentropy loss. Now fit the model with some batch size and number of epochs and some parameters that make the model more effic

4.6.5 Compile and train the model

Now compile the model with any optimizer and any loss.I have used Adam optimizer and categorical_cross entropy loss. Now fit the model with some batch size and number of epochs and some parameters that make the model more efficient. The feature will be extracted through the max-

pooling method by creating the model with .h5 extension and then compile the model with loss and optimizer. Here we import haar cascade for face recognition which is in XML format.

CHAPTER 5

Implementation and Result

CHAPTER 5

IMPLEMENTATION AND RESULT

5.1 Using openCV haarcascade xml detect the bounding boxes of face in the webcam and predict the emotions

Takes pictures or webcam video as input. It detects all faces in each frame, and then classifies which emotion each face is expressing. Recognized emotions: Neutral, Angry, Sad, Happy, Disgust, Fear,Surprise.Training accuracy was 95% with the following requirements:

1. Facial expression must be strong / exaggerated.
2. Adequate Lighting (no shadows on face)
3. Camera is at eye level or slightly above eye level.

5.2 Testing the Model

The project started off by defining a loading mechanism and loading the images. Then a training set and a testing set are created. After this a fine model and a few callback functions are defined.

The basic components of a convolutional neural network are considered and then training is done to the network. Create a folder named emotionss and save the emotionss corresponding to each of the seven emotions in the dataset.

5.3 APPLICATIONS

At Airports to observe the pilot‟s psychological condition before take-off. At Hospitals can be performed on a psychological disorder patient by a psychiatric doctor.

By the Crime Department used as a lie detector. Social websites feedback depicted through the face in the absence of written feedback or rating. Social Welfare gathering information would be profitable in the case of deaf and dumb people(autism patients).

Driver Monitoring monitoring driver facial expressions while driving. OTT Platforms and Video Trailers monitor the viewers emotions while watching movies or video trailers and generate the feedback Showrooms and Shops after purchasing any product, the buyer can give their feedback regarding the service provided to them. Text messages communication through emotionss is way too easier than typing long messages.

Robust spontaneous expression recognizers can be developed and deployed in real-time systems and used in building emotion sensitive HCI interfaces. The project can have an impact on our day to day life by enhancing the way we interact with computers or in general, our surrounding living and work spaces.

High correct recognition rate , significant performance improvements in our system. Promising results are obtained under face registration errors, fast processing time. System is fully automatic and has the capability to work with video feeds as well as images. It is able to recognize spontaneous expressions.

Our system can be used in Digital Cameras where the image is captured only when a person smiles, or if the person doesn’t blink his eyes. In security systems which can identify a person, in any form of expression he presents himself. Rooms in homes can set the lights, television to a person's taste when they enter the room. This can be used by the doctors in understanding the intensity of pain or illness of a deaf patient

5.4 Output Result

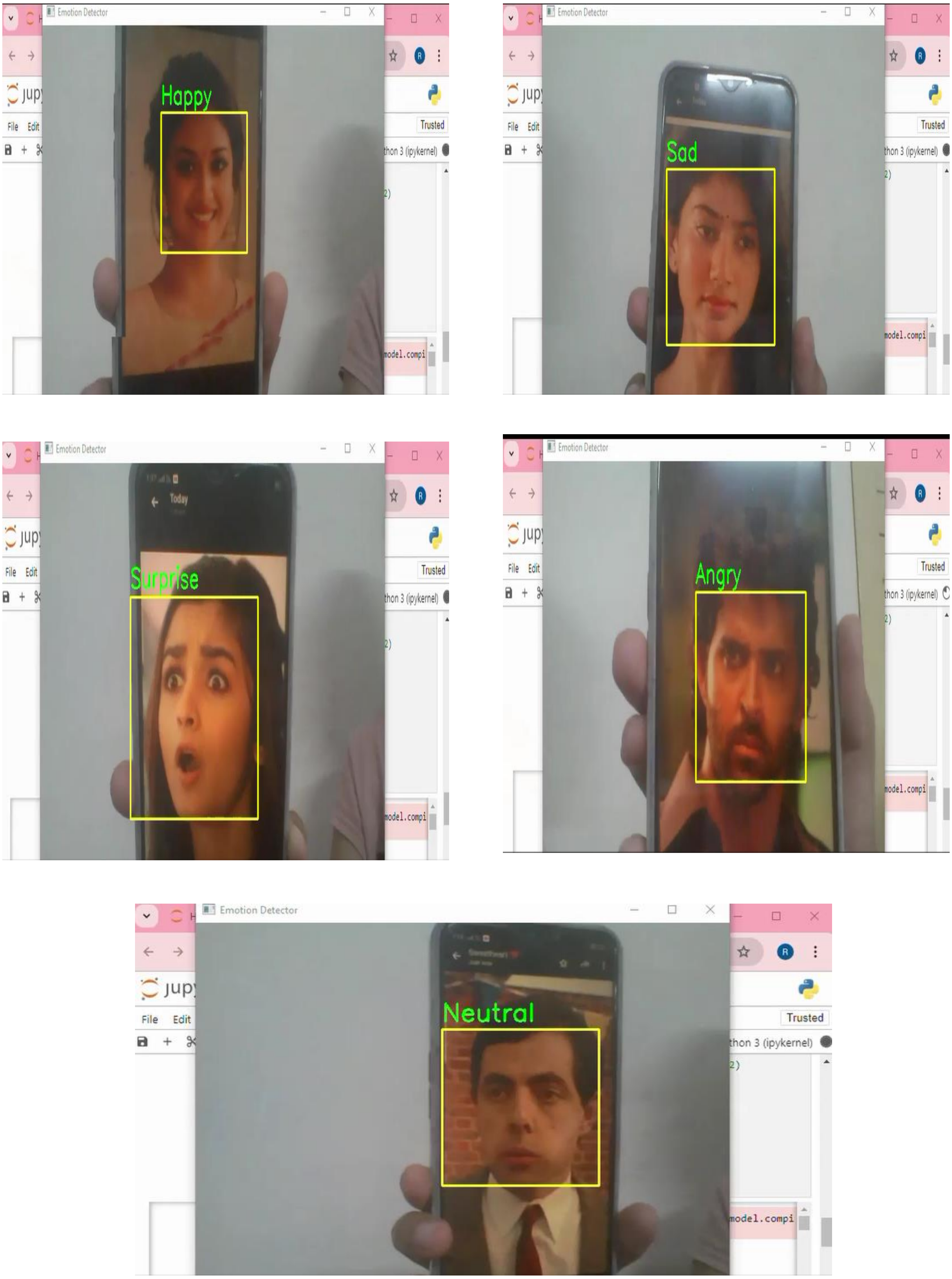


Fig 5.1 Detected facial emotion

CHAPTER 6

CONCLUSION

CHAPTER 6

CONCLUSION

6.1 ADVANTAGES:

1. **Enhanced Human-Computer Interaction (HCI)**
    - Improves user experience.
    - Creates empathetic and intuitive interfaces.
2. **Healthcare and Therapy**
    - Assists in diagnosing and monitoring mental health conditions.
    - Enables remote patient monitoring.
    - Supports therapeutic applications with real-time feedback.
3. **Marketing and Customer Service**
    - Analyzes customer emotions for better satisfaction.
    - Provides deeper insights into consumer reactions.
    - Personalizes customer service interactions.
4. **Security and Surveillance**
    - Identifies suspicious or abnormal behavior.
    - Assists law enforcement in assessing situations.
5. **Education**
    - Adapts educational content to students' emotional states.
    - Enhances personalized learning experiences.

6.2 SCOPE:

Face expression recognition systems have improved a lot over the past decade. The focus has definitely shifted from posed expression recognition to spontaneous expression recognition.

Promising results can be obtained under face registration errors, fast processing time and significant performance improvements can be obtained in our system.

System is fully automatic and has the capability to work with images feed. It is able to recognize spontaneous expressions.

The system can be used in Digital Cameras wherein the image can be captured only when the person smiles. In security systems which can identify a person, in any form of expression he presents himself. Doctors can use the system to understand the intensity of pain or illness of a deaf patient. Our system can be used to detect and track a user’s state of mind, and in mini-marts, shopping centers to view the feedback of the customers to enhance the business etc

REFERENCES

https://www.kaggle.com/code/jonathanoheix/face-expression-recognition-with-deep-learning/input

Video Link

https://drive.google.com/file/d/1bVRWhYgbovsyHiASXYVCXD6by5UzgyWX/view?usp=drive_link