EMOJIFY

UNIVERSITY NAVIGATION SYSTEM

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

IN

ARTIFICIAL INTELLIGENCE

AND MACHINE LEARNING

Submitted by:

PARTH

(20BCS6258)

DUSHANT SINGH

(20BCS6264)

Aditya Raj

(20BCS6275)

Vansh

(20BCS6263)

Under the Supervision of:

Ms. Arun Mittal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

2021-2022

DECLARATION

I, **'PARTH & DUSHANT SINGH**, student of **'Bachelor of Engineering in CSE(AI/ML)'**, **session: 2021-2022** , Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**EMOJIFY'** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

(PARTH)
UID: 20-BCS-6258
(DUSHANT SINGH)
UID: 20-BCS-6264

Date: May-2022

Place: Chandigarh

ABSTRACT

The aim of this project is to design a Detecting Human Face Emotion Using Emoj's. The system will let us know about the current sentiments of the person through picture video or live camera video.

Emoji's or avatars are ways to show facial expression or nonverbal cues. These expression have become more and more essential part of chatting/texting, showing emotion and many more. With advancement in computer vision and machine learning, we can detect human facial emotions from images or videos. In this project, we will detect human facial expression to filter and map corresponding emoji's or avatars.

The schools, colleges, institutions and parents can use the system for not only knowing emotions of their students but also make them understand more about themselves. Finding more about one's self can lead one to healthy mental state and it will be helpful for their mentors.This System will Save their energy and thus will keep them motivated throughout their learning journey.

The system can be used right from the kindergarten till student enter their teenage or for the person who is suffering from Alexithymia i.e when a person has difficulty identifying and expressing emotions.

The requirements are simple , the students only need a phone with good internet connection along with camera and they are ready to know. Also , the benefit of this system is that the parents or mentors will not have to guess there emotions as accuracy of our model is near about 70%.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Chandigarh University** for their guidance and constant supervision as well as for providing necessary information regarding the project and for their support in completing the project.

In the completion of my project **" EMOJIFY "**, I would like to convey my special gratitude to my mentor **Ms. Arun Mittal**, of AIT Department (Chandigarh University). I would like to express my special gratitude and thanks for giving me such attention and time. Your valuable guidance and suggestions helped me in various phases of the completion of this project. I will always be thankful to you in this regard.

Table of Contents

-

LIST OF FIGURES

1. INTRODUCTION

Deep learning (DL), a branch of machine learning (ML) and artificial intelligence (AI) is nowadays considered as a core technology. Due to its learning capabilities from data, DL technology originated from artificial neural network (ANN), has become a hot topic in the context of computing, and is widely applied in various application areas like healthcare, visual recognition, text analytics, cybersecurity, and many more. DL technology uses multiple layers to represent the abstractions of data to build computational models. While deep learning takes a long time to train a model due to a large number of parameters, it takes a short amount of time to run during testing as compared to other machine learning algorithms.

In traditional teaching activities, face-to-face communication between teachers and students enables learners to maintain a positive interest in learning at any time. In contrast, it is difficult for teachers and students to feel each other's emotional state in time due to the constraints of time and space in the intelligent learning environment. Therefore, this project of ours is an effective way to combine knowledge transmission with emotional communication in the current intelligent learning environment.

1.1 PROBLEM DEFINITION

Human emotions are complex and simple. As being a smart species on the earth, humans can express emotions through various methods, such as voice, text, and facial expressions. In this project we are trying to capture the emotion through the facial expressions.

The process of identifying the emotion from the face includes segmentation, isolation, and validation of facial features from the unstable environment and likely real faces. The initial effort to identify face was done by calculating unique facial characteristics such as nose size, brows width, and forehead area are new to campus and the major problem that they face is tiredness, waste of time, low energy and several other problems and in summers, this problem just gets worst.

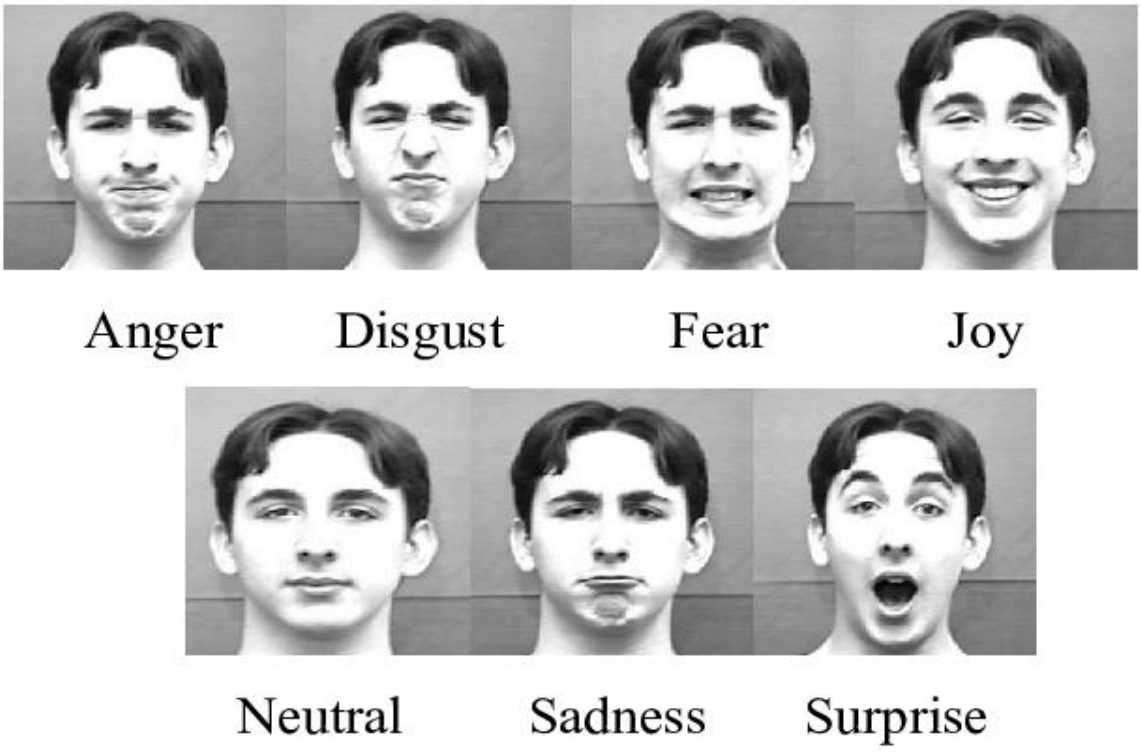

Fig. 1 Facial Emotions

The above image shows a variety of different number of Emotions and sentiments of human. And this system will help us analyze all these with just on click. Can be used to study persons reaction to numerous different things and all that data can help us analyze personality of that one person.

The process of face recognition comprises two major steps, the extraction of the feature and the classification. Raw face pictures can take a lot of time to identify as it results from an enormous amount of pixels. The number of pixels must be reduced. This is called reducing dimensional space (can be done using PCA). The extraction of features corresponds to the conversion of face space into a space of feature . Classification is the mechanism by which the class of variables is predicted.

The project is made using the frameworks like TensorFlow, Keras , Tkinter and libraries like numpy, pandas, PIL, etc.

FIG. 2 Development

1.2 PROJECT OVERVIEW/ SPECIFICATIONS

The project is titled as EMOJIFY. A project that can ease Teachers especially kindergarten efforts in their starting phase of the schooling. EMOJIFY is an advanced project combined with different powerful python libraries.

The main technology that will be used in our project is motion capture. Motion Capture is one of the concepts of In this deep learning project, we will classify human facial expressions to filter and map the corresponding emojis or avatars. By far, MediaPipe and CV2 is one of the most famous cross-platform libraries widely used for motion capture. The communication channel consists of a pipeline with optimized face, pose and hand components, each running in real time, with minimal memory, metamorphosing between their inference backend. MediaPipe's main quality is that it's a cross-platform library, which means it's platform-independent and available in multiple languages. In this project we have used CV2(haarcascade_frontalface_default.xml)

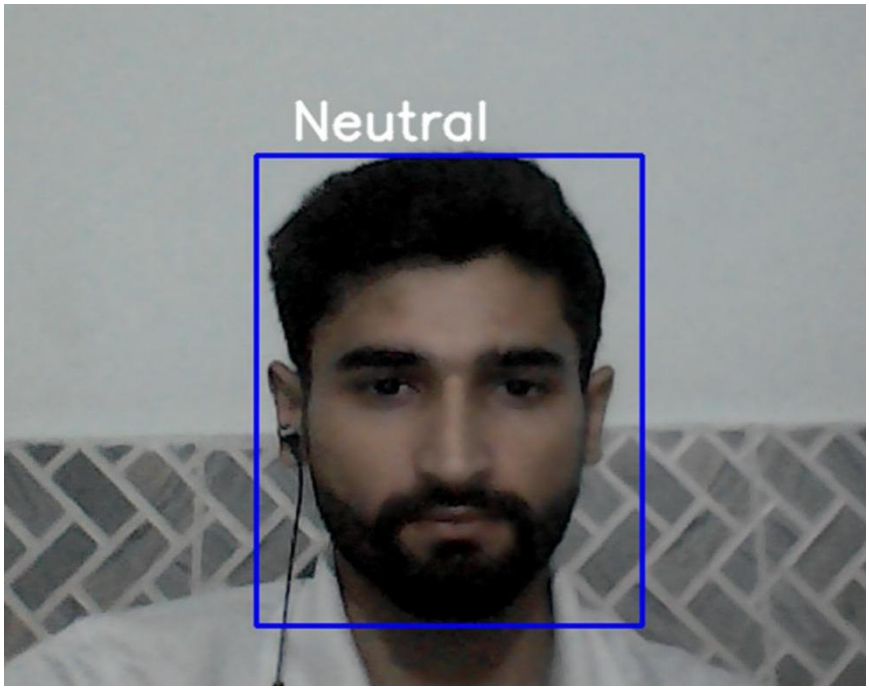

FIG. 3 A glimpse of used Sentiment Analysis

As we can see in the above fig 3, a picture of person is used and system is able to predict sentiment using facial expressions, the system will ask camera access to predict and will reflect the output on display screen.

1.3 Hardware Requirements:

- **PC/LAPTOP:** A laptop, desktop computer, or notebook computer is personal computer (PC) with a screen and alphanumeric keyboard. Laptops combine all the input/output components and capabilities of a desktop computer, including the display screen, small speakers, a keyboard, data storage device, sometimes an optical disc drive, pointing devices (such as a touch pad or pointing stick), with an operating system, a processor and memory into a single unit. Most modern laptops feature integrated webcams and built-in microphones, while many also have touchscreens. Laptops can be powered either from an internal battery or by an external power supply from an AC adapter. Hardware specifications, such as the processor speed and memory capacity, significantly vary between different types, models and price points.

  Portable computers, which later developed into modern laptops, were originally considered to be a small niche market, mostly for specialized field applications, such as in the military, for accountants, or traveling sales representatives. As portable computers evolved into modern laptops, they became widely used for a variety of purposes.

- **I5 Processor or AMD:** An Intel Corei5 provides better performance against heavier and demanding applications, games and rich audio-visual data using the embedded Intel Turbo Boost Technology. The Intel Corei5 comes in variations of two to four cores, all supporting four different threads simultaneously. Its processor clock speed ranges from 1.50 GHz to up to 3.10 GHz, with cache memory from 3 to 6 MB.

  The thermal design power (TDP) range goes from 84 TDP to as low as 15 TDP. Similar to Corei3, some of the latest generations of the Intel Corei5 support error correction code (ECC) memory and Intel Platform Protection Security and Intel OS Guards. These features provide embedded security abilities for protecting BIOS, enabling secure boot and prevention against attacks.

  Ryzen (pronounced RYE zen) is an AMD CPU aimed at the server, desktop, workstation, media center PC and all-in-one markets. AMD's Ryzen base models feature eight cores and 16-thread processing at 3.4Ghz with 20MB cache, neural net-based prediction hardware and smart prefetch. Codenamed Zen in development, Ryzen is the first major architectural change for AMD since Bulldozer.

  The AMD processor runs on new 1331 pin AM4 socket motherboards shared by FX models in high-end systems as well as APU models. AM4 motherboards support DDR4 RAM, PCIe Gen3. Ryzen can function as a system on a chip for low-cost, bare-bones setups. AM4 offers other chipset-enabled features, such as additional PCIe slots,

enhanced sound, SATA express, and USB 3.1 gen2 10Gbs connectivity. The shared socket also makes it easier to upgrade from base APU systems to discrete graphics cards along with FX CPUs for upgrades as well as those looking to at start entry level. Also of interest are claims of machine learning to predict instructions, making the CPU capable of learning how to run programs faster.

In launch demonstrations, un-boosted at 3.4Ghz, AMD Ryzen did slightly better than Intel's 6900K 3.2-3.7Ghz with turbo enabled. Those stats equal performance with a $1100 Intel part. One historical disadvantage to AMD's competitiveness has been Intel's manufacturing process lead. Using a smaller process has often given Intel's processors the ability to run at higher speed.

1.4 Software Requirements:

1.4.1 Language Used:

- **Python:** Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and

packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

1.4.2 Libraries Used:

- **Numpy:** NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

  NumPy provides an N-dimensional array type, the ndarray, which describes a collection of “items” of the same type. The items can be indexed using for example N integers. All ndarrays are homogeneous i.e. every item takes up the same size block of memory, and all blocks are interpreted in exactly the same way.

  An ndarray is a (usually fixed-size) multidimensional container of items of the same type and size. The number of dimensions and items in an array is defined by its shape, which is a tuple of N non-negative integers that specify the sizes of each dimension. The type of items in the array is specified by a separate data-type object (dtype), one of which is associated with each ndarray.

- **Open-CV(CV2):** OpenCV (Open Source Computer Vision Library) is an open-source library that includes several hundreds of computer vision algorithms. OpenCV contains various tools to solve computer vision problems. It contains low level image processing functions and high level algorithms for face detection, feature matching and tracking. Some of the main image processing techniques are given below:

a) Image Filtering:

It is a technique for modifying or enhancing an image. Image filtering is of two types. The one is linear image filtering, in which, the value of an output pixel is a linear combination of the values of the pixels of the input pixel's neighborhood. The second one is the non-linear image filtering, in which, the value of output is not a linear function of its input.

b) Image Transformation:

Image transformation generates "new" image from two or more sources which highlight particular features or properties of interest, better than the original input images. Basic image transformations apply simple arithmetic operations to the image data. Image subtraction is often used to identify changes that have occurred between images collected on different dates.

c) Object Tracking:

Object tracking is the process of locating a object (or multiple objects) over a sequence of images. It is one of the most important components in a wide range of applications in computer vision, such as surveillance, human computer interaction, and medical imaging.

d) Feature Detection:

Feature detection is a process of finding specific features of a visual stimulus, such as lines, edges or angle. It will be helpful for making local decisions about the local information contents (image structure) in the image. The modules of opens for image processing applications are given below:

CORE module contains basic data structures and basic functions used by other modules.

IMGPROC module contains image processing related functions such as linear, non-linear image filtering and geometrical image transformations etc.

VIDEO module contains motion estimation and object tracking algorithms.

ML module contains machine-learning interfaces.

HighGUI module contains the basic I/O interfaces and multi-platform windowing capabilities.

- **TensorFlow:** TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

TensorFlow is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.

- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

- **Keras:** Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

  Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

  MaxPooling2D Layer: Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool_size) for each channel of the input. The window is shifted by strides along each dimension.

  Tensorflow.keras.optimizer(ADAM): Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Adam

as combining the advantages of two other extensions of stochastic gradient descent. Specifically:

- **Adaptive Gradient Algorithm** (AdaGrad) that maintains a per-parameter learning rate that improves performance on problems with sparse gradients (e.g. natural language and computer vision problems).

- **Root Mean Square Propagation** (RMSProp) that also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight (e.g. how quickly it is changing). This means the algorithm does well on online and non-stationary problems (e.g. noisy).

Adam realizes the benefits of both AdaGrad and RMSProp.

Instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam also makes use of the average of the second moments of the gradients (the uncentered variance).

Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages.

The initial value of the moving averages and beta1 and beta2 values close to 1.0 (recommended) result in a bias of moment estimates

towards zero. This bias is overcome by first calculating the biased estimates before then calculating bias-corrected estimates.

ImageDataGenerator: Keras **ImageDataGenerator** class provides a quick and easy way to augment your images. It provides a host of different augmentation techniques like standardization, rotation, shifts, flips, brightness change, and many more.
However, the main benefit of using the Keras ImageDataGenerator class is that it is designed to provide real-time data augmentation. Meaning it is generating augmented images on the fly while your model is still in the training stage.

ImageDataGenerator class ensures that the model receives new variations of the images at each epoch. But it only returns the transformed images and does not add it to the original corpus of images. If it was, in fact, the case, then the model would be seeing the original images multiple times which would definitely overfit our model.

Another advantage of ImageDataGenerator is that it requires lower memory usage. This is so because without using this class, we load all the images at once. But on using it, we are loading the images in batches which saves a lot of memory.

- **Tkinter:** A Graphical User Interface allows the user to interact with the application created on different platforms. GUI interfaces use different indicators like audio indicators, graphical icons, different widgets which makes it highly interactive and user friendly rather than Command-Line applications which are not visually appealing and are text-based interactions.

  Tkinter provides a GUI look to the standard python interface. It comes pre-installed with the standard versions of Python on Windows, Linux, and macOS. Tkinter is a Python binding to the Tk GUI toolkit which is why it is named Tkinter. It is the most commonly used python GUI toolkit due to a large variety of widgets it supports and its ease of use.

  Tkinter provides powerful GUI based widgets and functions which create a visually appealing and highly creative application in just a few lines of codes. Tkinter is famous for creating a GUI application because it opens up in a new window where the user can interact with the application.

- **PIL:** The PIL or Python Imaging Library is often confused with Pillow. Pillow is a fork of the PIL library in Python, that's why to install PIL we write "pip install Pillow", instead of "pip install PIL". Some of the important features that this library offers you for image processing are:

    1. extensive file format support
    2. efficient internal representation
    3. creating thumbnails
    4. converting image files format
    5. applying filters to images
    6. also provides some powerful image processing capabilities

2. DESIGN FLOW AND DEPENDENCIES

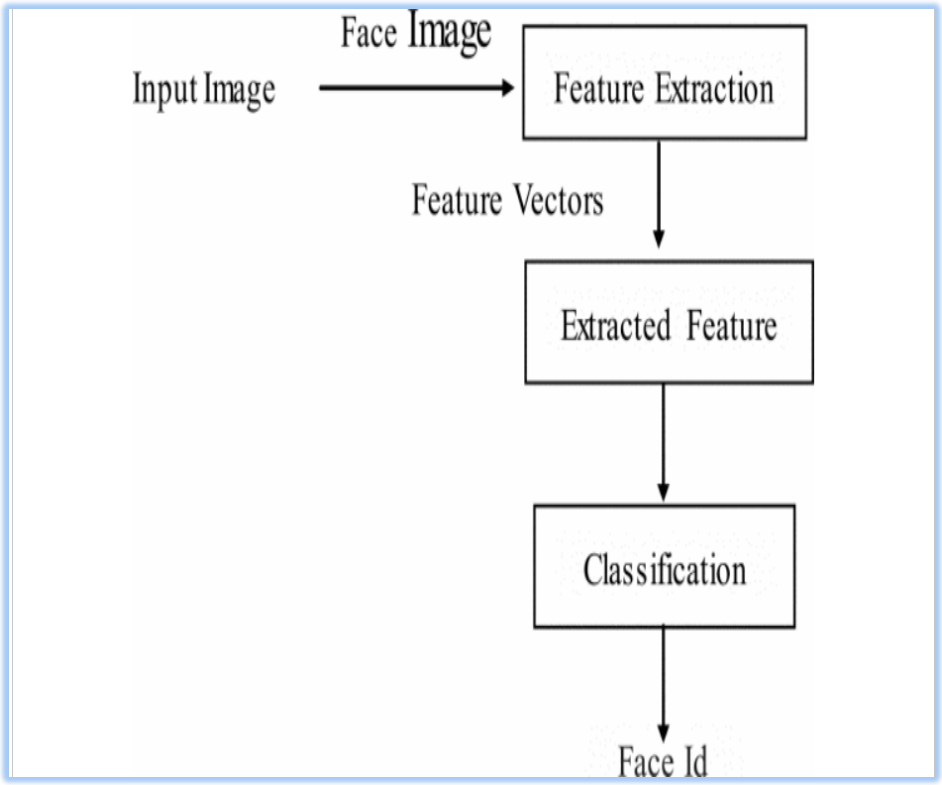

FIG 4. Design Flow

➢ **Dependency: Finish to Start**

- Data-set should be provided before hand.
- Run EDA steps(Exploratory Data Analysis).
- Create a ***CLASSIFICATION*** model using keras.sequential() model.

FIG 5. Model Flow

- **Data-Set:** The FER2013 dataset (facial expression recognition) consists of 48*48 pixel grayscale face images. The images are centred and occupy an equal amount of space. This dataset consist of facial emotions of following categories:

  - 0:angry
  - 1:disgust
  - 2:feat
  - 3:happy
  - 4:sad
  - 5:surprise
  - 6:natural

  This dataset is divided into training and testing set. Training dataset contain approx. 28,900 images capturing different emotions and testing data contain approx. 8000 images.

- **Model:** We used Sequential Model of keras for this project, but this can also be done by other machine-learning algorithms like: SVM, Naïve Bayes.

i. Input Image

Face picture is utilized as input to the extraction process of the feature.

ii. Feature Extraction

Transforming the actual image towards a more lightweight, and therefore more fundamentally different illustration. Here, principal component analysis (PCA) is used. PCA considers a different set of parameters such that all the parameters are orthogonal and measured by the deviation of the data within them. It implies that, first, there is a more essential principle axis.

iii. Extracted Features

This is the output of the Feature Extracted method. Extracted features will give as input for the classifier.

iv. Classifier

Evaluating the facial expression is achieved by the classifier for a specified feature matrix. The machine-learning algorithm is used as a classifier. It has been experimented using linear discriminant analysis, multilayer perceptron, Naive Bayes, and support vector machine. In this project we have used Sequential() model of tensorflow.

1.2.1 Types of Computer Vision

Computer vision, a type of artificial intelligence, enables computers to interpret and analyze the visual world, simulating the way humans see and understand their environment. It applies machine learning models to identify and classify objects in digital images and videos, then lets computers react to what they see.

- **Face mapping:-** Very first thing in face detection is mapping the face. There are tons of facial features that all in total makes facial expression such as :-
  - Mouth
  - Right eyebrow
  - Left eyebrow
  - Right eye
  - Left eye
  - Nose
  - Jaw

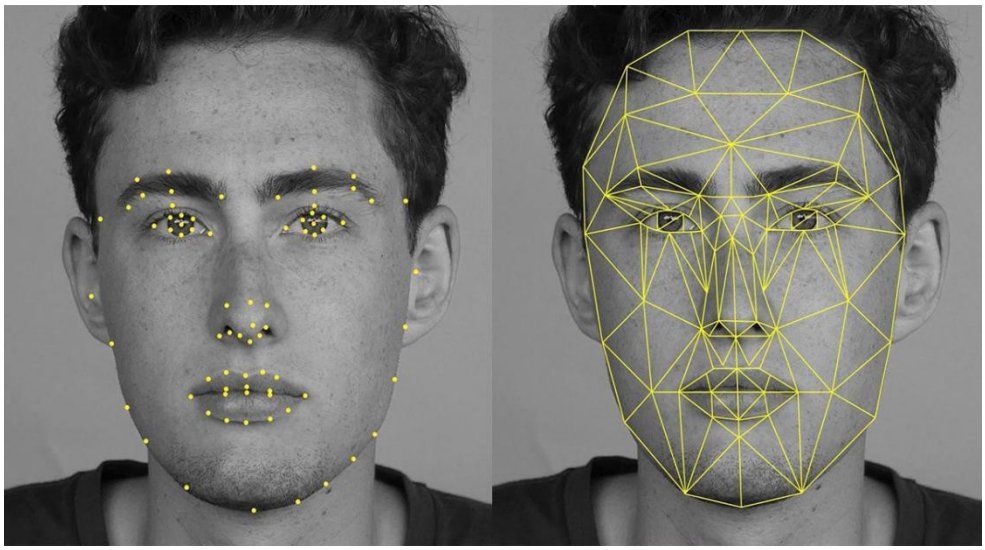

FIG 6. Face Mapping

- **Body mapping:-**Second thing in face detection is mapping the face. There are tons of facial features that all in total makes facial expression such as :-
  - Shoulders
  - Ankle
  - Knee
  - Wrist
  - Hands(Fingers)
  - Head

Fig 7: Body Mapping

1.5 History Of Sentiment Analysis

Sentiment_analysis is a term that refers to the use of natural language processing, text analysis, and computational linguistics in order to ascertain the attitude of a speaker or writer toward a specific topic. Basically, it helps to determine whether a text is expressing sentiments that are positive, negative, or neutral. Sentiment analysis is an excellent way to discover how people, particularly consumers, feel about a particular topic, product, or idea.

Fig 9: Ancient predictive analysis

As the technology advanced and technology reached India, in other words if we say when the mobile phones reached India, it started to see some change in people behaviour. It had both negative and positive impact on people. Technology made the start of development and soon after years and years and years, as India advanced in technology, around 2000's several companies started working on machine learning. We all know about the predictive analysis of AI Based future.

FIG 10. AI Based Future

Digital India Campaign:

In 2020, the Indian government increased the outlay for Digital India to $477 million to boost AI, IoT, big data, cybersecurity, machine learning and robotics. India's flagship digital initiative aims to make the internet more accessible, promoting e-governance, e-banking, e-education and e-health. In the 2019 Union Budget, Finance Minister Nirmala Sitharaman said the government would offer industry-relevant skill training for 10 million youth in India in technologies like AI, Big Data and robotics.

According to a report by AIMResearch titled "How The Indian Government Is Championing The AI Revolution", the use cases of AI in the Indian government include facial recognition and hotspot analysis, biometric identification, criminal investigation, traffic and

crowd management, wearables to empower women safety, optimising revenues in the forest, cleaning river, tiger protection, digital agriculture, student progress monitoring and more.

Fig 11 : Digital India Campaign

Additionally, policy-level initiatives by the Ministry of Electronics and Information Technology (MeitY) and programmes around AI by NASSCOM and Defence Research & Development Organization (DRDO) have laid the groundwork for future disruption and created a roadmap for AI in India.

One such initiative was establishing the Centre for Artificial Intelligence and Robotics (CAIR), a laboratory of the DRDO, in 2014 for research and development in AI, robotics, command and control,

networking, information and communication security. CAIR shoots for the development of mission-critical products for battlefield communication and management systems.

Overview of the Artificial Intelligence Market in India

The Artificial Intelligence technology function is no longer an emerging technology segment – AI as a function has pervaded almost all industries and functions – from eCommerce to BFSI and from Manufacturing to Agriculture – Data Science and Deep Learning are increasingly utilised to solve complex business challenges. AI is increasingly utilised across several B2B, B2C, and, even, C2C (Consumer-to-Consumer) channels.

- AI is increasingly adopted across Contact Center Customer Services (RPA-driven Chat Bots), Media Delivery (ML and AI-driven Social Media, Streaming content, and eCommerce recommendations), and Intelligent Networks / Telecom Services, to name a few.

- AI adoption is no longer restricted to financial enterprises or large technology implementations or organizations. AI adoption in increasingly becoming democratized with personnel from non-technological backgrounds adopting AI processes in their functional roles.

- Machine Learning and AI technologies are central to the functioning of Smartphones, Smart TVs, Household Appliances, and Automobiles.

As revealed in the AI and Analytics start-up investment study for 2020 – 2021, the Data Science space in India, specifically the AI function, is evolving into an innovative enterprise segment. MNC Technology, Domestic, Advanced Engineering, Healthcare, and Semiconductor firms, to name a few, are now developing advanced Artificial Intelligence capabilities in India and providing top-notch AI services to firms across various industries and geographies – they are being rewarded with substantial investments by foreign and domestic funds.

Artificial Intelligence Market Size by Company Type:

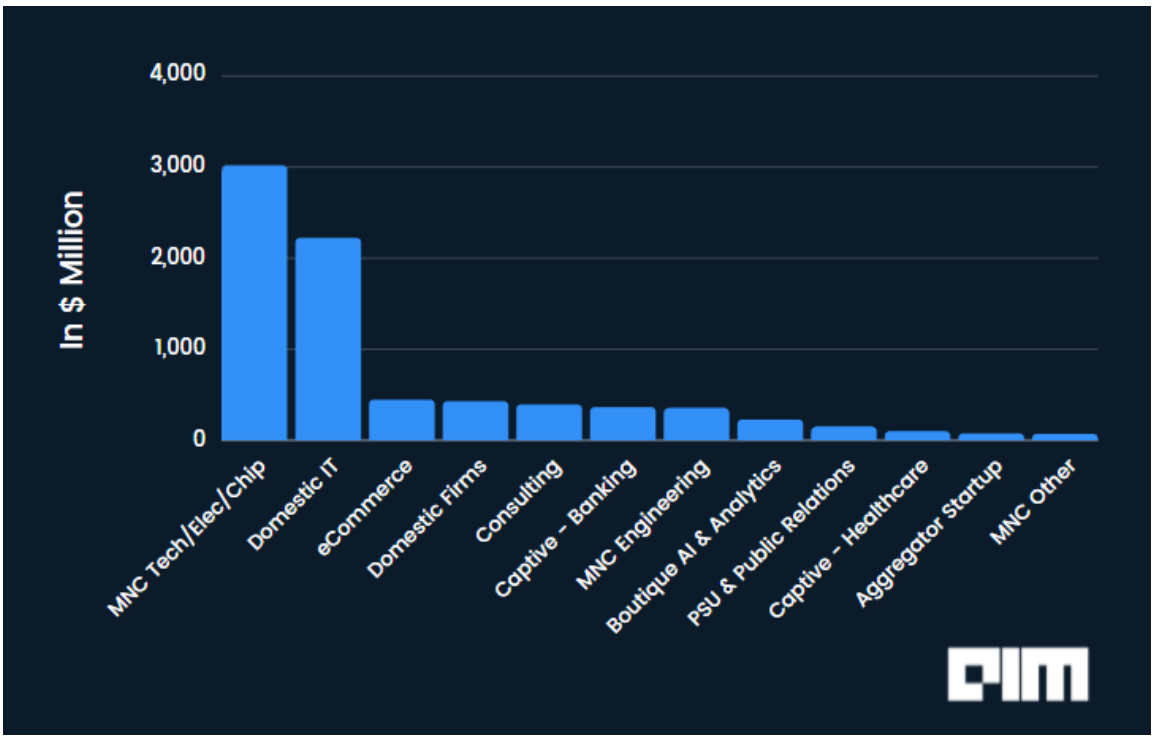

FIG 12. Market Size and Comapny type

The Artificial Intelligence market size in India covers the companies providing Artificial Intelligence services from the India geography, regardless of the geographical market and type of industry the services are provided to. The Artificial Intelligence market in India is valued at $7.8 Bn.

MNC IT | Technology | Semi-conductor | Electronics

In terms of Company Type, the broad-based MNC IT, Technology, and Electronics category has the highest share of the AI market at 38.8% in percentage share and $3012.4 Mn in terms of Market Value. The large share of this category is obvious given the size and scale of the IT and

Technology firms that fall under this category, all of which have significant AI operations in India. Accenture, as it had last year, has the largest AI operations base in India across the broad MNC IT and Tech category.

The enterprises that fall under this category include:

- MNC IT Services firms, such as Accenture, IBM, and Capgemini,among others, which provide AI services, as part of the larger Digital and Data Science service offerings, to their international and domestic clients.
- Software Technology firms, such as Microsoft, Google, SAP, Oracle and AWS, among others – Microsoft as it had last year, has the largest AI operations in India across this sub-category.
- MNC Hardware Technology, Networking Equipment, and MNC Telecom firms, including Dell, Cisco, Juniper, Nokia, Verizon, and AT&T, among others.
- Electronics and High-end Semi-conductor firms, such as Samsung, Intel, **Nvidia, Qualcomm**, and **AMD** among others, which develop AI technologies for Smart Phones, Processing Units, Chips, Memory Devices, Servers, Data Centres, and other sensor-driven technologies.

Artificial Intelligence Market Size by Sector/Industry:

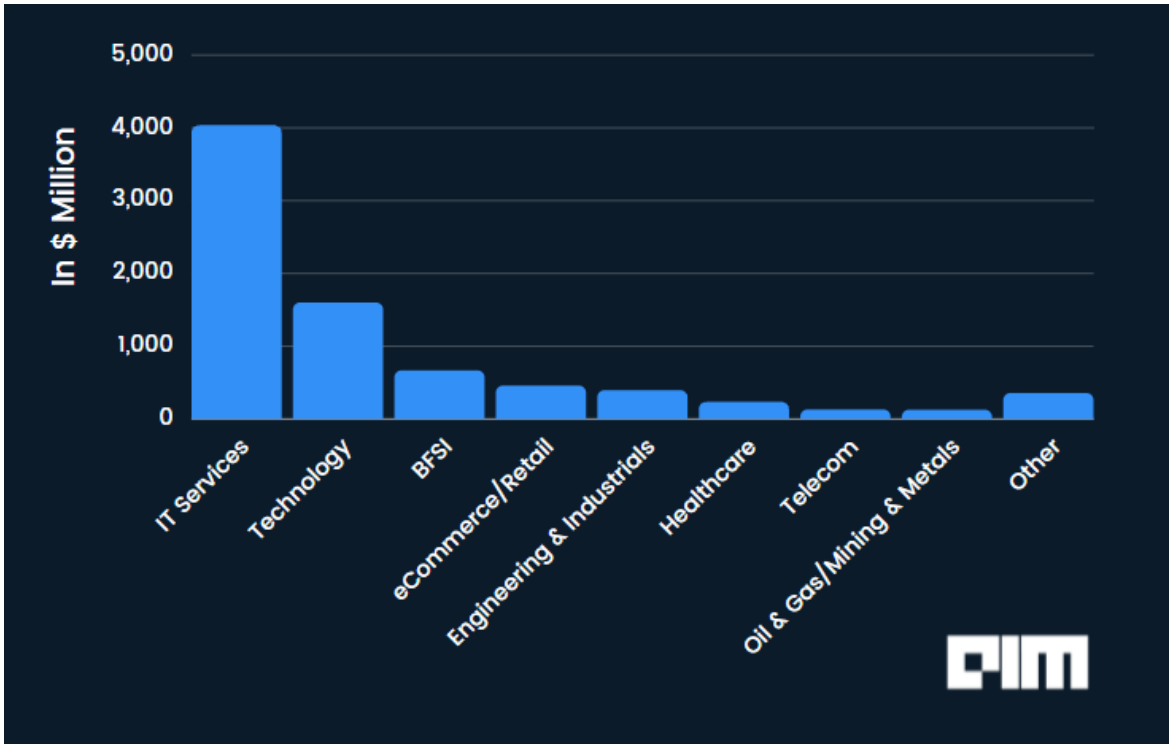

FIG 13. Market Size and Sector type

Analytics India Magazine has analysed the AI market for 2021 by sector or industry.

The sector-wise distribution of AI services covers:

- Firms that provide services within a sector or industry and utilize AI to deliver those services – such as Amazon within eCommerce and JP Morgan within BFSI.

- Firms within another segment, such as Boutique AI and Analytics, but provide exclusive services to a particular function within an industry – such as Mad Street Den providing Computer Vision and

Artificial Intelligence services to the Retail sector or Spyne providing services to the broad-based Digital and Media industry.

3. Literature Survey:

3.1 Existing System:

In an existing system user can make emojis in iOS, android or windows but it does not show the person's body language or facial emotion. Also it is for fun loving peoples and not particularly developed for learning and teaching.

3.2 Proposed System:

In our proposed system user can make their emojis or bit-emojis on any device but also it will detect their body language and facial expressions or emotions. In future, after some modifications we will be making web based application implementing this model so that this product can be used for learning and teaching purpose in school, NGOs and other government felicitated or private organizations.

4. METHODOLOGY

- **DFD**: We used the Sequential Model of Keras for this project, but this can also be done by other machine-learning algorithms like SVM, and Naïve Bayes. We have used FER2013 dataset. This dataset is divided into training and testing set. Training dataset contain approx. 28,900 images capturing different emotions and testing data contain approx. 8000 images

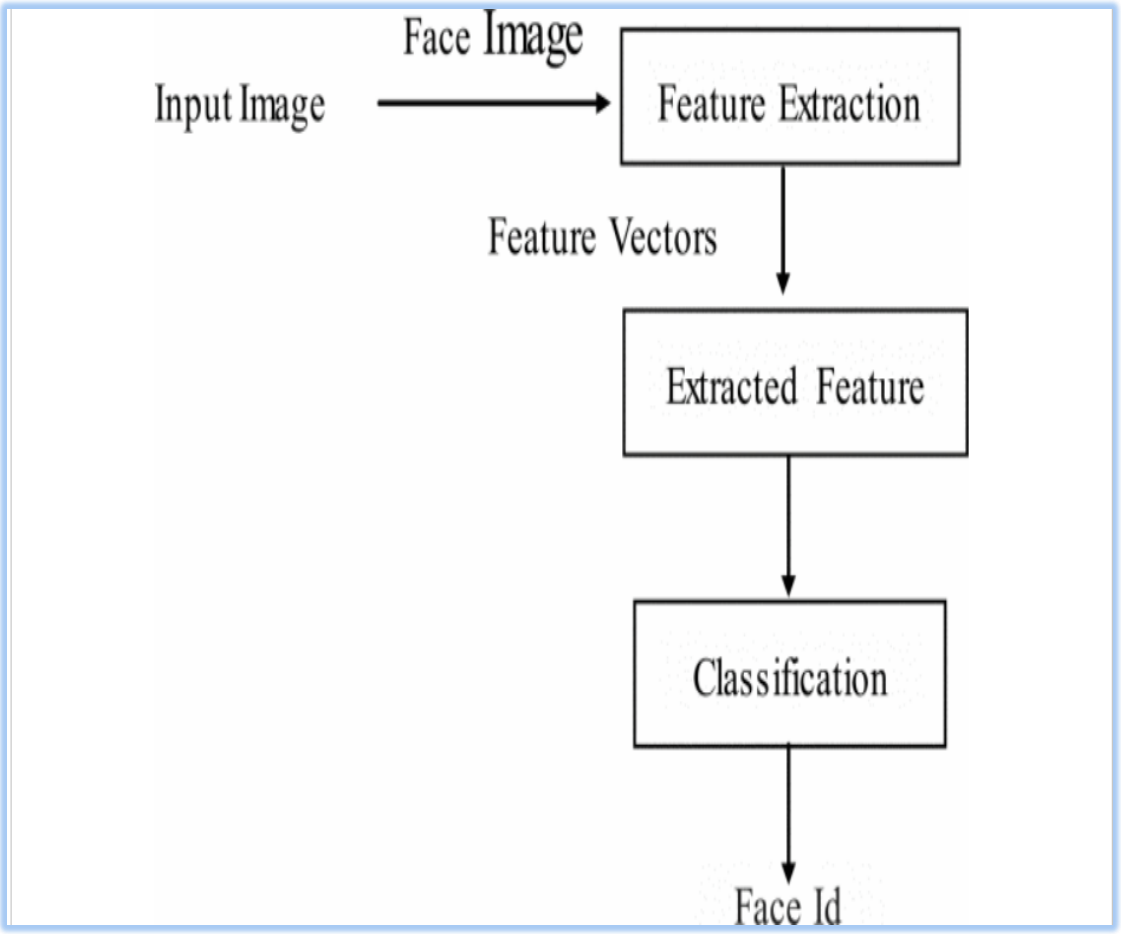

Fig 14. Flow Chart

- We have use libraries like numpy, PIL, tkinter, os, keras and tensorflow.

```
import numpy as np
import cv2
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from keras.layers import MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
from PIL import Image as Img
from PIL import ImageTk
import os
import tkinter as tk
from tkinter import *
```

Fig 15. Imported Libraries

- Loading the segregated dataset directly from the directories and then creating the training and testing generators to train and test the model.

```
train_dir = 'data/train'
val_dir = 'data/test'
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')

validation_generator = val_datagen.flow_from_directory(
        val_dir,
        target_size=(48,48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')
```

Fig 16. Train and Test Data

- Creating the Sequential() model of keras and also adding different layers.

```
emotion_model = Sequential()

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

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

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))
```

Fig 17. Sequential Model

- Compiling the model with the learning rate of 0.0001. This model has 50 epochs i.e. cycles and each cycle contains 448 steps. This is done in order to improve the overall accuracy of the model.

```
Epoch 1/50
448/448 [==============================] - 244s 544ms/step - loss: 1.7988 - accuracy: 0.2630 - val_loss: 1.7057 - val_accuracy: 0.3491
Epoch 2/50
448/448 [==============================] - 260s 580ms/step - loss: 1.6238 - accuracy: 0.3687 - val_loss: 1.5430 - val_accuracy: 0.4194
Epoch 3/50
448/448 [==============================] - 258s 574ms/step - loss: 1.5234 - accuracy: 0.4161 - val_loss: 1.4707 - val_accuracy: 0.4329
Epoch 4/50
448/448 [==============================] - 259s 578ms/step - loss: 1.4556 - accuracy: 0.4466 - val_loss: 1.4077 - val_accuracy: 0.4644
Epoch 5/50
448/448 [==============================] - 258s 575ms/step - loss: 1.3986 - accuracy: 0.4673 - val_loss: 1.3530 - val_accuracy: 0.4886
Epoch 6/50
448/448 [==============================] - 258s 576ms/step - loss: 1.3516 - accuracy: 0.4834 - val_loss: 1.3137 - val_accuracy: 0.5025
Epoch 7/50
448/448 [==============================] - 245s 548ms/step - loss: 1.3082 - accuracy: 0.5064 - val_loss: 1.2869 - val_accuracy: 0.5155
Epoch 8/50
448/448 [==============================] - 236s 527ms/step - loss: 1.2700 - accuracy: 0.5195 - val_loss: 1.2543 - val_accuracy: 0.5215
Epoch 9/50
448/448 [==============================] - 234s 523ms/step - loss: 1.2306 - accuracy: 0.5351 - val_loss: 1.2298 - val_accuracy: 0.5335
Epoch 10/50
448/448 [==============================] - 261s 583ms/step - loss: 1.2023 - accuracy: 0.5456 - val_loss: 1.2082 - val_accuracy: 0.5398
Epoch 11/50
448/448 [==============================] - 294s 656ms/step - loss: 1.1725 - accuracy: 0.5593 - val_loss: 1.1896 - val_accuracy: 0.5513
Epoch 12/50
448/448 [==============================] - 269s 599ms/step - loss: 1.1479 - accuracy: 0.5701 - val_loss: 1.1795 - val_accuracy: 0.5536
Epoch 13/50
448/448 [==============================] - 274s 611ms/step - loss: 1.1180 - accuracy: 0.5768 - val_loss: 1.1762 - val_accuracy: 0.5502
Epoch 14/50
448/448 [==============================] - 259s 579ms/step - loss: 1.0960 - accuracy: 0.5879 - val_loss: 1.1528 - val_accuracy: 0.5646
Epoch 15/50
448/448 [==============================] - 271s 606ms/step - loss: 1.0704 - accuracy: 0.5995 - val_loss: 1.1428 - val_accuracy: 0.5671
Epoch 16/50
448/448 [==============================] - 262s 585ms/step - loss: 1.0469 - accuracy: 0.6035 - val_loss: 1.1328 - val_accuracy: 0.5730
Epoch 17/50
448/448 [==============================] - 275s 615ms/step - loss: 1.0188 - accuracy: 0.6210 - val_loss: 1.1168 - val_accuracy: 0.5788
Epoch 18/50
448/448 [==============================] - 232s 519ms/step - loss: 0.9971 - accuracy: 0.6320 - val_loss: 1.1200 - val_accuracy: 0.5806
Epoch 19/50
448/448 [==============================] - 241s 538ms/step - loss: 0.9738 - accuracy: 0.6394 - val_loss: 1.0995 - val_accuracy: 0.5907
Epoch 20/50
448/448 [==============================] - 238s 532ms/step - loss: 0.9587 - accuracy: 0.6455 - val_loss: 1.1108 - val_accuracy: 0.5925
Epoch 21/50
448/448 [==============================] - 230s 514ms/step - loss: 0.9319 - accuracy: 0.6554 - val_loss: 1.0963 - val_accuracy: 0.5922
Epoch 22/50
448/448 [==============================] - 241s 538ms/step - loss: 0.9044 - accuracy: 0.6641 - val_loss: 1.0845 - val_accuracy: 0.6014
Epoch 23/50
448/448 [==============================] - 241s 537ms/step - loss: 0.8775 - accuracy: 0.6784 - val_loss: 1.0879 - val_accuracy: 0.5999
Epoch 24/50
448/448 [==============================] - 246s 549ms/step - loss: 0.8602 - accuracy: 0.6833 - val_loss: 1.0834 - val_accuracy: 0.6016
Epoch 25/50
448/448 [==============================] - 242s 540ms/step - loss: 0.8365 - accuracy: 0.6939 - val_loss: 1.0806 - val_accuracy: 0.6055
Epoch 26/50
448/448 [==============================] - 249s 555ms/step - loss: 0.8174 - accuracy: 0.6990 - val_loss: 1.0785 - val_accuracy: 0.6067
Epoch 27/50
448/448 [==============================] - 247s 552ms/step - loss: 0.7949 - accuracy: 0.7090 - val_loss: 1.0762 - val_accuracy: 0.6122
Epoch 28/50
448/448 [==============================] - 235s 525ms/step - loss: 0.7674 - accuracy: 0.7201 - val_loss: 1.0776 - val_accuracy: 0.6101
Epoch 29/50
448/448 [==============================] - 245s 547ms/step - loss: 0.7458 - accuracy: 0.7240 - val_loss: 1.0714 - val_accuracy: 0.6162
Epoch 30/50
448/448 [==============================] - 245s 547ms/step - loss: 0.7263 - accuracy: 0.7345 - val_loss: 1.0875 - val_accuracy: 0.6152
```

```
Epoch 31/50
448/448 [==============================] - 248s 554ms/step - loss: 0.7045 - accuracy: 0.7434 - val_loss: 1.0806 - val_accuracy: 0.6154
Epoch 32/50
448/448 [==============================] - 243s 542ms/step - loss: 0.6796 - accuracy: 0.7537 - val_loss: 1.0917 - val_accuracy: 0.6208
Epoch 33/50
448/448 [==============================] - 248s 554ms/step - loss: 0.6606 - accuracy: 0.7585 - val_loss: 1.0951 - val_accuracy: 0.6173
Epoch 34/50
448/448 [==============================] - 243s 542ms/step - loss: 0.6395 - accuracy: 0.7688 - val_loss: 1.0824 - val_accuracy: 0.6169
Epoch 35/50
448/448 [==============================] - 243s 542ms/step - loss: 0.6163 - accuracy: 0.7755 - val_loss: 1.1076 - val_accuracy: 0.6197
Epoch 36/50
448/448 [==============================] - 244s 544ms/step - loss: 0.5919 - accuracy: 0.7837 - val_loss: 1.1062 - val_accuracy: 0.6253
Epoch 37/50
448/448 [==============================] - 240s 536ms/step - loss: 0.5783 - accuracy: 0.7898 - val_loss: 1.1203 - val_accuracy: 0.6176
Epoch 38/50
448/448 [==============================] - 242s 539ms/step - loss: 0.5557 - accuracy: 0.7976 - val_loss: 1.1154 - val_accuracy: 0.6182
Epoch 39/50
448/448 [==============================] - 239s 533ms/step - loss: 0.5475 - accuracy: 0.8016 - val_loss: 1.1112 - val_accuracy: 0.6229
Epoch 40/50
448/448 [==============================] - 244s 544ms/step - loss: 0.5274 - accuracy: 0.8090 - val_loss: 1.1309 - val_accuracy: 0.6194

Epoch 41/50
448/448 [==============================] - 238s 530ms/step - loss: 0.4993 - accuracy: 0.8194 - val_loss: 1.1440 - val_accuracy: 0.6179
Epoch 42/50
448/448 [==============================] - 240s 535ms/step - loss: 0.4904 - accuracy: 0.8211 - val_loss: 1.1364 - val_accuracy: 0.6214
Epoch 43/50
448/448 [==============================] - 243s 541ms/step - loss: 0.4734 - accuracy: 0.8299 - val_loss: 1.1624 - val_accuracy: 0.6244
Epoch 44/50
448/448 [==============================] - 242s 540ms/step - loss: 0.4566 - accuracy: 0.8360 - val_loss: 1.1537 - val_accuracy: 0.6244
Epoch 45/50
448/448 [==============================] - 245s 548ms/step - loss: 0.4379 - accuracy: 0.8425 - val_loss: 1.1734 - val_accuracy: 0.6240
Epoch 46/50
448/448 [==============================] - 242s 541ms/step - loss: 0.4270 - accuracy: 0.8466 - val_loss: 1.1888 - val_accuracy: 0.6222
Epoch 47/50
448/448 [==============================] - 242s 540ms/step - loss: 0.4142 - accuracy: 0.8512 - val_loss: 1.2104 - val_accuracy: 0.6233
Epoch 48/50
448/448 [==============================] - 238s 530ms/step - loss: 0.4014 - accuracy: 0.8539 - val_loss: 1.1890 - val_accuracy: 0.6261
Epoch 49/50
448/448 [==============================] - 236s 526ms/step - loss: 0.3920 - accuracy: 0.8606 - val_loss: 1.2041 - val_accuracy: 0.6219
Epoch 50/50
448/448 [==============================] - 234s 523ms/step - loss: 0.3793 - accuracy: 0.8642 - val_loss: 1.2097 - val_accuracy: 0.6214
```

Fig18. Trained model Accuracy

- Saving the model weights and then deploying after the web cam feed.

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

```
cv2.ocl.setUseOpenCL(False)

emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}


  cap = cv2.VideoCapture(0)
  while True:
      # Find haar cascade to draw bounding box around face
      ret, frame = cap.read()
      if not ret:
          break
      bounding_box = cv2.CascadeClassifier
      ('C:/Users/Hp/anaconda3/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')
      gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      num_faces = bounding_box.detectMultiScale(gray_frame,scaleFactor=1.3, minNeighbors=5)

      for (x, y, w, h) in num_faces:
          cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
          roi_gray_frame = gray_frame[y:y + h, x:x + w]
          cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray_frame, (48, 48)), -1), 0)
          emotion_prediction = emotion_model.predict(cropped_img)
          maxindex = int(np.argmax(emotion_prediction))
          cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLEX, 1,
                      (255, 255, 255), 2, cv2.LINE_AA)

      cv2.imshow('Video', cv2.resize(frame,(1200,860),interpolation = cv2.INTER_CUBIC))
      if cv2.waitKey(1) & 0xFF == ord('q'):
          break

  cap.release()
  cv2.destroyAllWindows()
```

Fig19. Web Cam feed

5. Result And Conclusion:

- Human emotions are complex and simple. As being a smart species on the earth, humans can express emotions through various methods, such as voice, text, and facial expressions.
- In this project we are trying to capture the emotion through the facial expressions through live video and we have also tried to map it with emoji's.
- This project that can ease Teacher's work especially in kindergarten efforts in their starting phase of the schooling. EMOJIFY is an advanced project combined with different powerful python libraries.
- Further we need to enhance the UI/UX of the product.

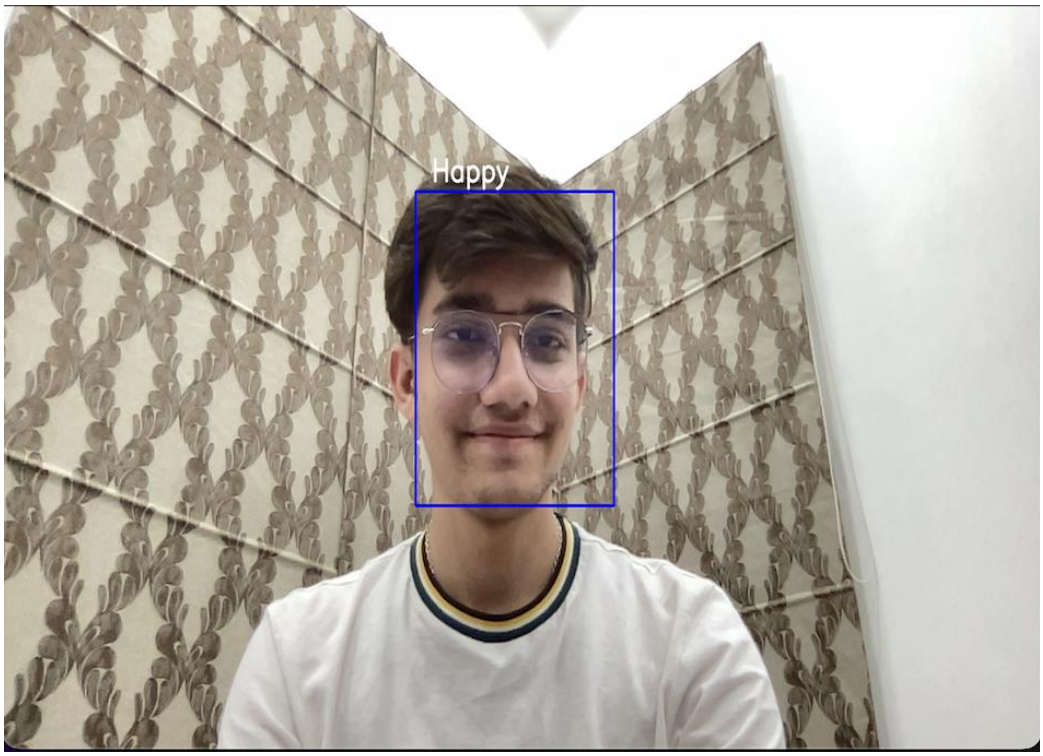

Fig 20. System Output

6. References

- IEEE Xplore. *(references)*
- Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions, by Iqbal H. Sarker, Article number: 420 (2021).
- frontiersin.org, Front. Psychol., 06 January 2022.
- https://keras.io/about/.
- https://analyticsindiamag.com/complete-guide-to-develop-an-interface-using-tkinter-python-gui-toolkit/.
- tensorflow.org/resources/libraries-extensions.
- OpenCV - CV2.