EMOJIFY

Dushant Singh
AIT-CSE
Chandigarh University
Chandigarh, India
dushantthakur890@gmail.com

Parth
AIT-CSE
Chandigarh University
Chandigarh, India
parth2003mahajan@gmail.com

Aditya Raj
AIT-CSE
Chandigarh University
Chandigarh, India
Rajaditya02520@gmail.com

Vansh
AIT-CSE
Chandigarh University
Chandigarh, India
janmedavansh@gmail.com

Emojis or avatars are ways to show facial expressions or nonverbal cues. These expressions have become a more and more essential part of chatting/texting, showing emotion and many more. With the 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 expressions to filter and map corresponding emojis or avatars.

I. INTRODUCTION (*HEADING 1*)

Deep learning (DL), a branch of machine learning (ML) and artificial intelligence (AI) is nowadays considered a core technology. Due to its learning capabilities from data, DL technology originated from an 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.

Human emotions are complex and simple. As 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 emotion through 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 the face was done by calculating unique facial characteristics such as nose size, brows width, and forehead area (can be done using mediapipe or cv2). Face recognition was introduced as an authentication tool in the latest gadgets. Smartphone vendors including Apple and Samsung have launched their new mobile phone variants with facial authentication functionality.

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 (which 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.

II. LITERATURE REVIEW

The main technology that will be used in our project is motion capture. Motion Capture is one of the concepts of Deep Learning and it is gaining popularity day by day. According to research done by scientists; The human body is a very complex system, consisting of more than 200

joints, and if we need to simulate the entire movement of the human body in the most vivid way, we will provide all the rotation angles. about the value and position of the human body. and other relevant information. Besides, we have emojis and avatars. Emojis are a way to display nonverbal cues. These signs have become an integral part of online chat, product reviews, brand logos, and more. There has also been an increase in data science research focusing on emoji-based storytelling. Thanks to advances in computer vision and deep learning, it is now possible to detect human emotions from images. 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 are 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)

III. TOOLS/LIBRARIES USED

A. *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 the same way.

A 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.

B. *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. 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 neighbourhood. The second one is non-linear image filtering, in which, the value of output is not a linear function of its input.

b) Image Transformation:

Image transformation generates a "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 an 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 angles. It will help make 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.

C. 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.

D. 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 large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

E. Tkinter

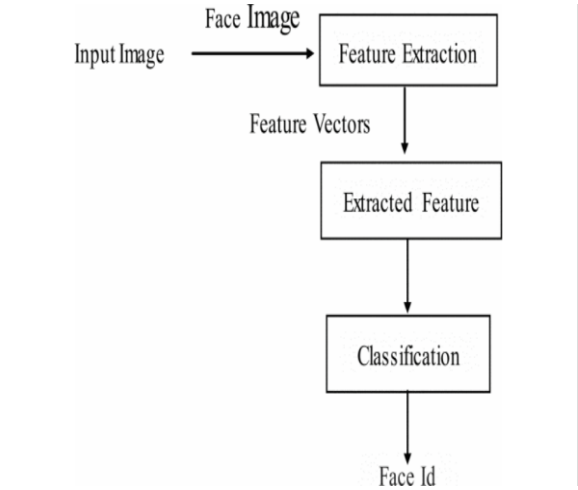
A Graphical User Interface allows the user to interact with the application created on different platforms. GUI interfaces use different indicators like audio indicators, graphical icons, and different widgets which makes them highly interactive and user friendly rather than Command-Line applications 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 the 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 code. Tkinter is famous for creating a GUI application because it opens up a new window where the user can interact with the application.

IV. MAKING KERAS ML MODEL

A) *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.*



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 with using linear discriminant analysis, multilayer perceptron, Naive Bayes, and support vector machine. In this project, we have used the Sequential() model of TensorFlow.

V. Data Set Used

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 consists of facial emotions in following categories:

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

VI. Conclusion

In this paper, an investigation is used to automatically detect the face of the person. The FER2013 dataset is used for performing experiments. This dataset is divided into training and testing sets. The training dataset contains approx. 28,900 images capturing different emotions and testing data contain approx. 8000 images. Then these two data sets are used to train the sequential model (of Keras). As a result, after prediction, the numerical value predicted is mapped to its corresponding emotion, using dictionaries. In last we tried to map the emoji's depicting the emotion of a person.

VII. Future Work

We will try to improve the accuracy of the model. And will use our dataset to improve the results. We will be updating this model from facial emotion recognition to full body language detection and we will be using live animation for the same.

This model can be used in online meeting platforms like zoom meetings google meet or any other platform.

Acknowledgement *(Heading 5)*

This paper and the research behind it would not have been possible without the exceptional support of our mentor, *Arun Mittal*. His enthusiasm, knowledge and exacting attention to detail have been an inspiration and kept our work on track.

I would also like to thank my group members: Parth, Aditya Raj and Vansh for their contribution to making the project and for helping me in this research.

.

References

[1] IEEE Xplore. *(references)*

[2] Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions, by Iqbal H. Sarker, Article number: 420 (2021).

[3] frontiersin.org, Front. Psychol., 06 January 2022.

[4] https://keras.io/about/.

[5] https://analyticsindiamag.com/complete-guide-to-develop-an-interface-using-tkinter-python-gui-toolkit/.

[6] tensorflow.org/resources/libraries-extensions.

[7] OpenCV - CV2.