# Emojinator

### Description

Emojis are ideograms and smileys used in electronic messages and web pages.

Emoji exist in various genres, including facial expressions, common objects, places and types of weather, and animals.

They are much like emoticons, but emoji are actual pictures instead of typographics

**Code Requirements**

we can use any python IDE where as in my case I have used **pycharm**.

MODULES USED ARE:

🡪NUMPY (**NumPy is a module for Python**. **NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better know under the heading of "big data". Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.)**

**🡪**SCIPY(**SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy needs Numpy, as it is based on the data structures of Numpy and furthermore its basic creation and manipulation functions. It extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others.)**

🡪CV2(**OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code**)

🡪 MATPLOTLIB(**Matplotlib is a**[**plotting**](https://en.wikipedia.org/wiki/Plotter)[**library**](https://en.wikipedia.org/wiki/Library_(computer_science))**for the**[**Python**](https://en.wikipedia.org/wiki/Python_(programming_language))**programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy). It provides an**[**object-oriented**](https://en.wikipedia.org/wiki/Object-oriented_programming)[**API**](https://en.wikipedia.org/wiki/API)**for embedding plots into applications using general-purpose**[**GUI toolkits**](https://en.wikipedia.org/wiki/GUI_toolkit)**like [Tkinter](https://en.wikipedia.org/wiki/Tkinter" \o "Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython" \o "WxPython),**[**Qt**](https://en.wikipedia.org/wiki/Qt_(software))**, or**[**GTK+**](https://en.wikipedia.org/wiki/GTK%2B)**. There is also a**[**procedural**](https://en.wikipedia.org/wiki/Procedural_programming)**"pylab" interface based on a**[**state machine**](https://en.wikipedia.org/wiki/State_machine)**(like**[**OpenGL**](https://en.wikipedia.org/wiki/OpenGL)**), designed to closely resemble that of**[**MATLAB**](https://en.wikipedia.org/wiki/MATLAB)**, though its use is discouraged**)

🡪PANDAS(*Pandas***is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in***C***or***Python*)

🡪KERAS(**Keras is an**[**open-source**](https://en.wikipedia.org/wiki/Open-source_software)[**neural-network**](https://en.wikipedia.org/wiki/Artificial_neural_network)**library written in**[**Python**](https://en.wikipedia.org/wiki/Python_(programming_language))**. It is capable of running on top of [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow" \o "TensorFlow),**[**Microsoft Cognitive Toolkit**](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit)**, [Theano](https://en.wikipedia.org/wiki/Theano_(software)" \o "Theano (software)), or [PlaidML](https://en.wikipedia.org/wiki/PlaidML" \o "PlaidML).**[**[1]**](https://en.wikipedia.org/wiki/Keras#cite_note-1)[**[2]**](https://en.wikipedia.org/wiki/Keras#cite_note-2)**Designed to enable fast experimentation with**[**deep neural networks**](https://en.wikipedia.org/wiki/Deep_learning)**, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System))**

**🡪**H5PY**(The h5py package is a Pythonic interface to the**[**HDF5**](http://hdfgroup.org/)**binary data format.)**

### Functionalities

Tensorflow's object detection API for training SSD with MobilnetV1

Filters to detect hand.

CNN for training the model.

### Python Implementation

🡪Network Used- Convolutional Neural NetworkProcedure

🡪First, generate images using *hand\_images.py.* (Make sure that you take images from different angles.)\_

🡪Data will be stored in /data folder

🡪Annotate the data using the *labelImg* program by *tzutalin*.

🡪For training, you can use model uploaded in GITHUB /*hand\_detection\_inference\_graph* (https://github.com/akshaybahadur21/Emojinator/tree/master/Emojinator\_V2)

🡪Finally, run *Emojinator2.py* via webcam