Implementation:

1)Emojify – Create your own emoji with CNN:

In Emojify, we take the input of the user through their webcam and return the emoji according to their input reaction.

For this particular part of our project, we will use CNN (Convolutional Neural Network). We will build and train a CNN model on FER2013 dataset CNN is a class of deep neural networks. CNN is usually used in processing data which has a grid-like topology, such as an image, which is the basic requirement in this part of our project.

Step1:

Import Libraries:

The most important library used for this part is CV2 in python. OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more.

After that we use keras library which will used to build a fully connected neural network.

Step 2:

We now go on to build a model and we will train the model. In this step we build a Convolutional Network Architecture We use relu activation function at every layer of the neural network.

After that, we combine the perceptron layers that we built and train the model.

After training the model, we save the model weights for the best possible output.

Step 3:

We use the 'haarcascade' xml function to find the boundaries of the face when the webcam is turned on. The image of the user's face is then passed on to the CNN model which we built and will predict the emotions of the user.

Step 4:

Our next step is to map the emotions and return an emoji according to the output of the CNN model and to build a GUI for a good presentation.

There are seven possible emotions in the dataset, so we have to map them to seven different emojis.

For this purpose, we can use tkinter library. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Using tkinter we can easily create and a GUI for our project using python.

2) Cartooning an image using OpenCV and python:

We will use Open cv for converting the input image to a simple cartoon image. OpenCV is a cross-platform library used for Computer Vision. It includes applications like video and image capturing and processing. It is majorly used in image transformation, object detection, face recognition, and many other stunning applications.

Step 1:

Importing necessary libraries,

- CV2: Imported to use OpenCV for image processing
- Easygui: Imported to open a file box. It allows us to select any file from our system.
- Numpy: Images are stored and processed as numbers. These are taken as arrays. We use NumPy to deal with arrays.
- Imageio: Used to read the file which is chosen by file box using a path.
- Matplotlib: This library is used for visualization and plotting. Thus, it is imported to form the plot of images.
- OS: For OS interaction. Here, to read the path and save images to that path.

Step 2:

Building a File Box to choose a particular file,

In this step, we will build the main window of our application, where the buttons, labels, and images will reside. We also give it a title by title() function.

In this step, we will build a method which will a pop-up box to choose an image from your hard-disk, which will open every time you run your code.

fileopenbox () is the method in easyGUI module which returns the path of the chosen file as a string.

Step 3:

In this step, we will store the image as Numpy array.

Imread is a method in cv2 which is used to store images in the form of numbers. This helps us to perform operations according to our needs. The image is read as a Numpy array, in which cell values depict R, G, and B values of a pixel.

Now comes the main part, to convert the image to a cartoon.

To convert an image to a cartoon, multiple transformations are done. Firstly, an image is converted to a Grayscale image. Yes, similar to the old day's pictures.! Then, the Grayscale image is smoothened, and we try to extract the edges in the image. Finally, we form a colour image and mask it with edges. This creates a beautiful cartoon image with edges and lightened colour of the original image.

To accomplish the task, we have to follow six steps to get the cartoon image,

- Convert the image to grayscale.
    - cvtColor (image, flag) is a method in cv2 which is used to transform an image into the colour-space mentioned as 'flag'. s
- Smoothening the grayscale image.
    - To smoothen an image, we simply apply a blur effect. This is done using medianBlur () function. Here, the centre pixel is assigned a mean value of all the pixels which fall under the kernel. In turn, creating a blur effect.
- Retrieving the edges of an image.
    - In this step, we will work on the first specialty. Here, we will try to retrieve the edges and highlight them. This is attained by the adaptive thresholding technique. The threshold value is the mean of the neighbourhood pixel values area minus the constant C. C is a constant that is subtracted from the mean or weighted sum of the neighborhood pixels.

- Preparing a mask image
    - So, let's combine the two specialties. This will be done using MASKING. We perform bitwise and on two images to mask them.
- And finally, giving a cartoon image.

After completing the above two components of our project, we combine the above two components of our project using tkinter. We create a GUI to dynamically run the above two parts in a single window on click of a single button.