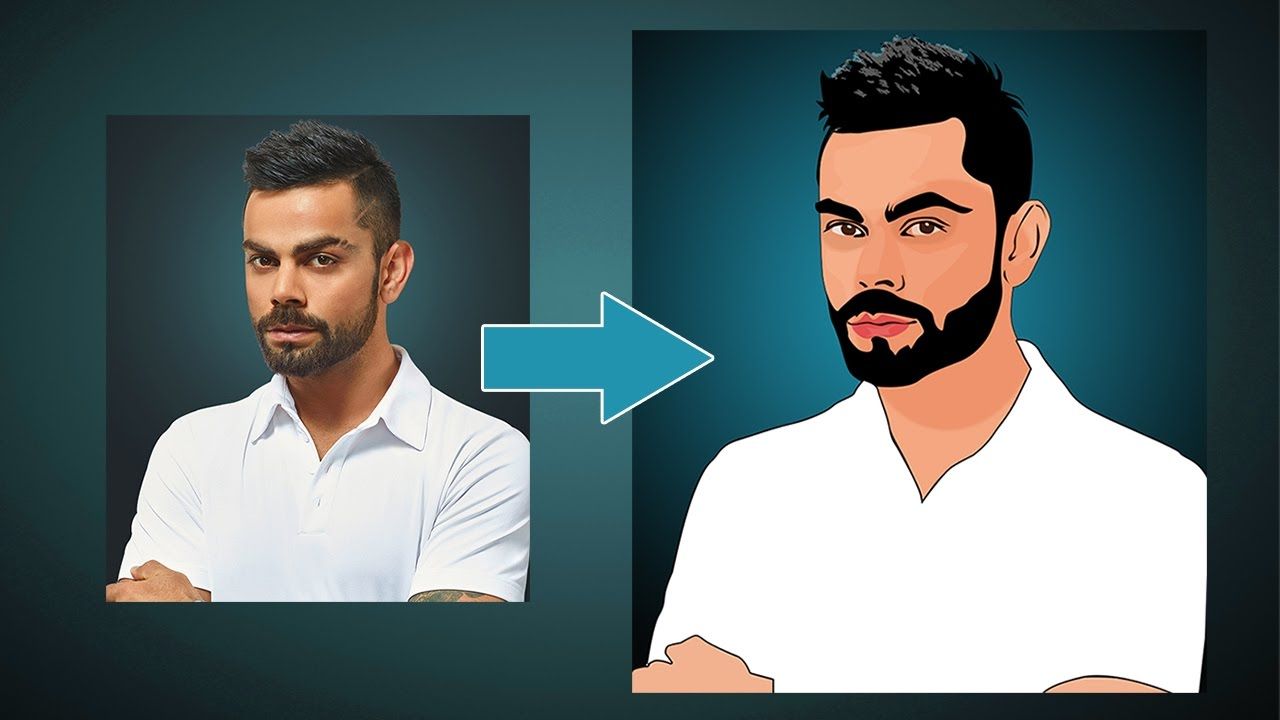
Cartoonfiy Image Using OpenCV and Python

Introduction:

We will build one interesting application that will cartoonify the imageprovided to it.To build this cartoonifyer application we will use python and openCV.

This is one of the exciting and thrilling application of Machine Learning.While building this application we will also see how to use libraries like asygui,Tkinter,and all.Here you have to select the image and then the application will convert that image into its cartoon form.mainly,we build this application using OpenCV and Python as the programming languages



OpenCV

OpneCV is an open-source library in python that is used mainly for computer vision tasks in the area of machine learning and artificial intelligence.Nowadays,openCV is playing a major role in the field of technology.Using OpenCV we can process images and videos for some tasks like object detection, face detection, object tracking, and all.

OpenCV has c,c++,java, and python interface and it supports all kinds of systems such as windows,Linux,Android,Mac OS,los,and all..

Requirements

Python: We use python as a programming language for building the application.

Cv2: We use cv2 for image processing.

NumPy: Mainly NumPy is used for dealing with arrays.Here the images that we use are stored in the form of arrays.So far that,we use Numpy.

Easygui: easygui is a module used for GUI programming in python. In our application easygui is used to open the file box to upload images from local system.

Imageio: Imageio is a python library that reads and writes the images.

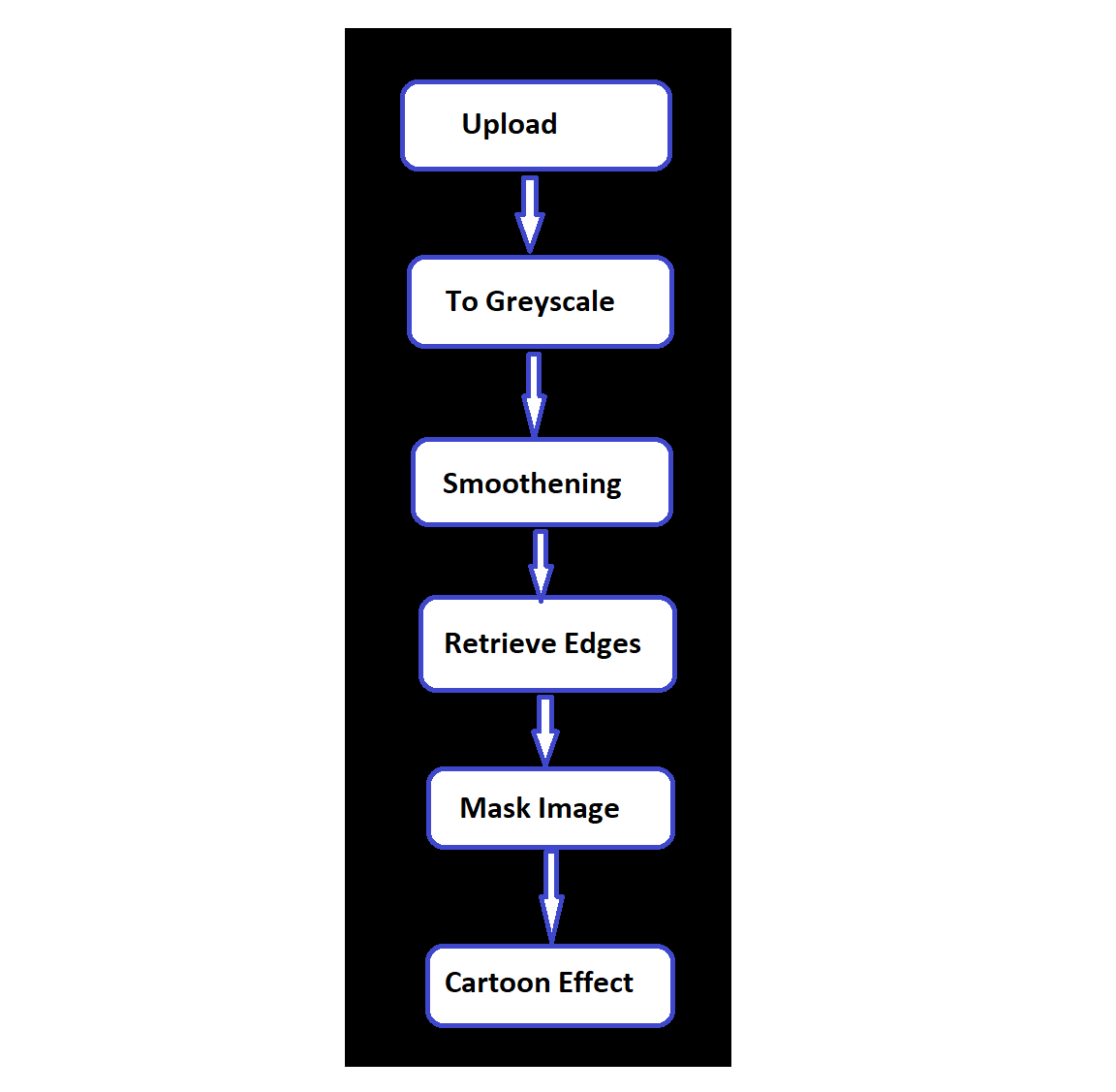
Matplotlib: Matplotlib is used for visualization purposes.

Here we plot the images using matplotlib.

OS: Here in our application OS is used for dealing with paths like reading images from the path and saving the image to the path.

Tkinter: Tkinter is a standard Graphical User Interface(GUI) package.

Methodology



This is the methodology that we are going to follow to build our cartoonify application. First of all using easygui, we will upload the image, and then the image is converted to a greyscale image.

The next two steps are the important steps to converting image into cartoon images. They are smoothening and then retrieving the edges. In this color of the image is smoothened to give the cartoon look and then we retrieve the edges and then highlight them in the final image.

Next, we will prepare a mask Image. In this, we use the bilateral filter with removes the noise and smoothen it to some extent. Now the final step is giving the cartoon effect. To the image which we got in the previous step, we combine our two important steps and finally give a mask-edged image that looks like a cartoon image.

Implementation

Start by importing all the required libraries such as NumPy, cv2, easygui, imageio, sys,matplotlib,os,and required Tkinter libraries.

TO upload the image from the local system, define an upload function and use a fileopenbox that opens the box to choose a file, and stores the file path.

Def upload():

ImagePath=easygui.fileopenbox()

cartoonify(ImagePath)

Define Cartoonify Function

Now create a function cartoonify which includes all the steps from converting to greyscale to the final cartoon image. The first step is to convert it into a greayscale image and then apply the blur to smoothen the image which is one of the main steps in cartooning the image. For smoothening, the blur effect is given using the median blur() function. and then retrieving the edges and highlighting for cartoon effect which is another important step of the application and then applying the bilateral filter which is in built that helps to remove the noise present in the image and provides the clean image and then masking the edged image and Finally we plot an image which contain all the six traditions throughout the process.

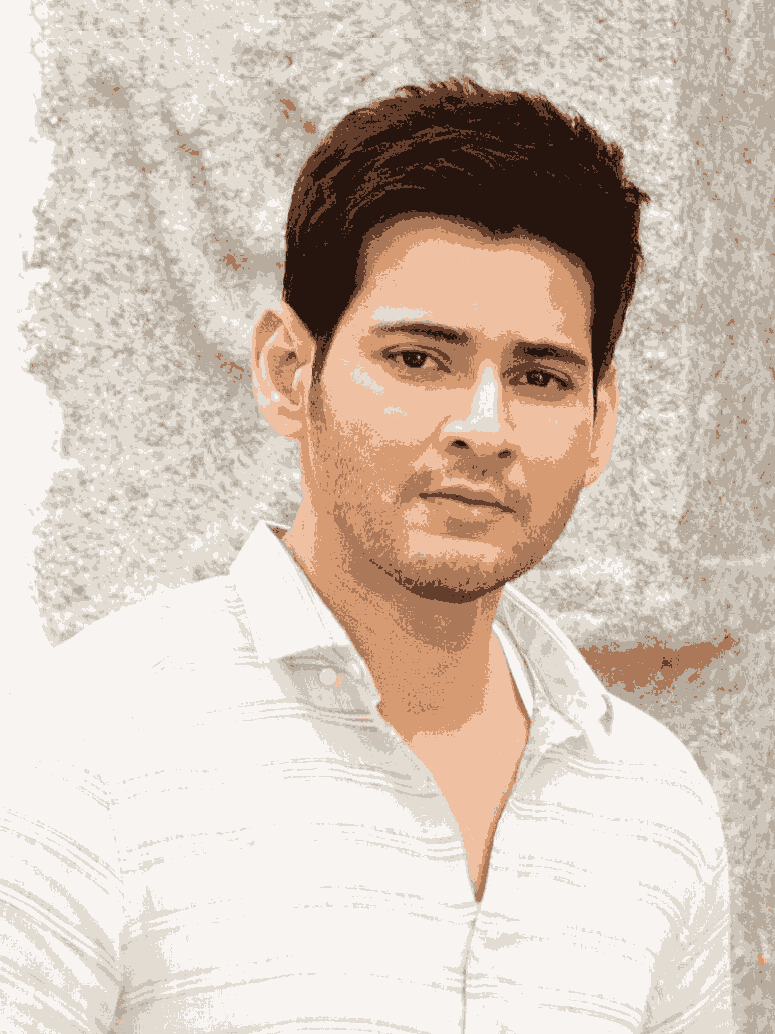
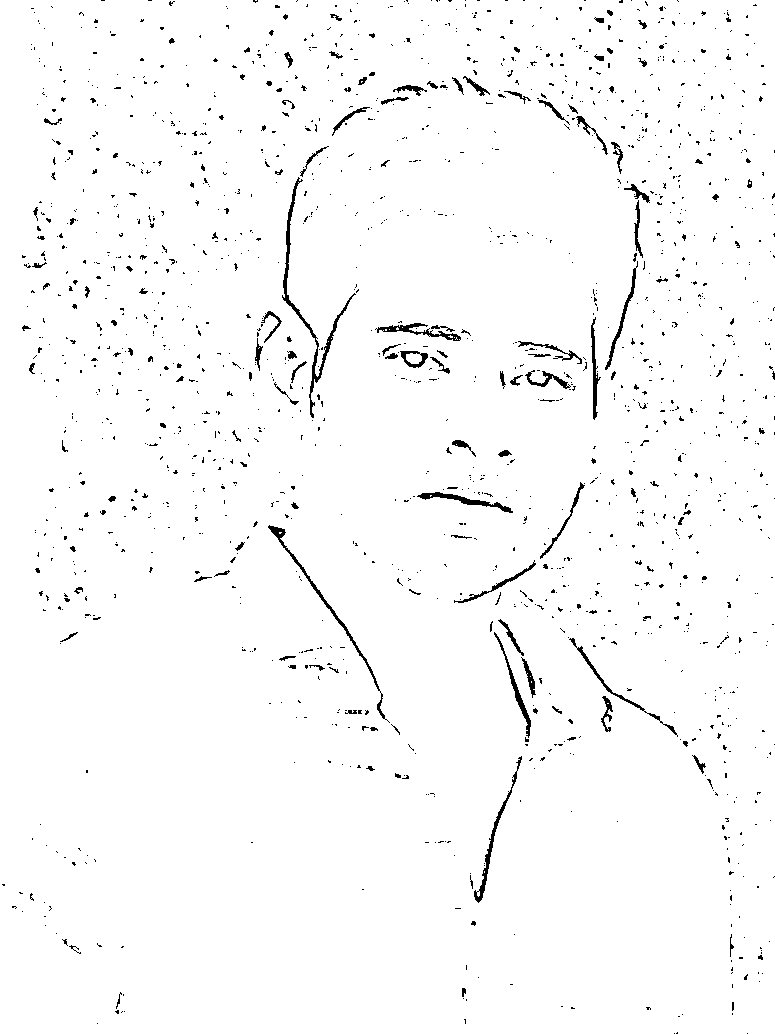
To plot all the transitions, we form a list of all images. We have saved all the images from first as resized1, resized2, resized3, resized4, resized ,resized, On a single plot, we will plot all these six. And then finally plt.show() will print all the images.

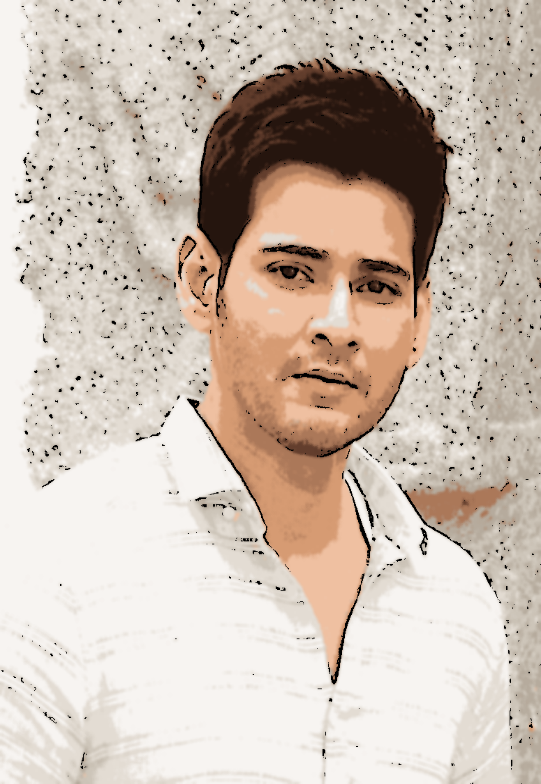
Steps we followed:

Step1:Upload image file



Step2: Run code to create cartoon Effect





Conclusion

Finally, we will get the output image as shown above.it contains all the transitions of the image. And the final image is the cartoon image.we enjoyed a lot this application