RAGHU INSTITUTE OF TECHNOLOGY

(Autonomous)

Accredited by NAAC and NBA, Affiliated to JNTU-Kakinada Dakamarri (V),

Bheemunipatnam (M),

Visakhapatnam – 531162,

2021

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

CARTOONING AN IMAGE USING OPENCV

A Mini Project report submitted at the end of sixth semester

K. RAVI TEJA

(Regd No: 183J1A0584)

M. GAYATRI

(Regd No: 183J1A05B2)

K. RAVI TEJA

(Regd No: 183J1A0591)

M.TARUN

(Regd No: 183J1A05A4)

M.S.K. VARMA

(Regd No: 183J1A05A3)

RAGHU INSTITUTE OF TECHNOLOGY

(Autonomous)

Accredited by NAAC and NBA, Affiliated to JNTU-Kakinada Dakamarri (V),

Bheemunipatnam (M),

Visakhapatnam

CERTIFICATE

This is to certify that this project entitled "**CARTOONING AN IMAGE USING OPENCV**" done by **M. GAYATRI, K. RAVI TEJA, M. TARUN, K. RAVI TEJA, M.S.K. VARMA** bearing **Regd. No: 183J1A05B2, 183J1A0584, 183J1A05A4, 183J1A0591, 183J1A05A3** respectively during the academic year 2020-2021 in partial fulfillment of the requirements for the completion of fifth semester of **Bachelor of Technology** in **Computer Science and Engineering.**

Internal Examiner **External Examiner**

DECLARATION

This is to certify that this project titled **“CARTOONING AN IMAGE USING OPENCV”** is bonafide work done by us, in partialfulfilment of the requirements for the completion of fifth semester of the degree B.Tech and submitted to the Department of Computer Science & Engineering, Raghu Institute of Technology, Dakamarri.

We also declare that this project is a result of our own effort and that has not been copied from anyone and we have taken only citations from the sources which are mentioned in the references.

M. GAYATRI	183J1A05B2
M. TARUN	183J1A05A4
K. RAVI TEJA	183J1A0584
K. RAVI TEJA	183J1A0591
M.S.K.VARMA	183J1A05A3

PLACE: RIT, Visakhapatnam **DATE: 16-08-2021**

TABLE OF CONTENTS

1. ABSTRACT

Since the client for animation pictures recovery framework focuses to get applicable pictures to inquiry picture from information base inside same item (for example a client has animation picture with object, for this situation the client will focus to get all significant picture with, along these lines A significant advance in animation picture recovery is characterizing the item inside animation picture .

In this project, a proficient technique for objects extraction from animation pictures is presented; it depends on broad suppositions identified with shading and areas of items in animation pictures, the items are commonly gravitated toward the focal point of the picture, the foundation tones is the all the more much of the time gravitated toward the edges of animation picture, and the item colors is less touch for the edges.

The cycles of shading quantization, seed filling and found the item apparition have been utilized. The aftereffects of led tests showed that the framework have promising effectiveness for extricating both single or multi object(s) lay in straightforward and complex foundations of animation pictures.

2. INTRODUCTION TO OPENCV

OpenCV is an open-source python library used for computer vision and machine learning. It is mainly aimed at real-time computer vision and image processing. It is used to perform different operations on images which transform them using different techniques.

It majorly supports all languages like Python, C++, Android, Java, etc. It is easy to use and in demand due to its features. It is used in creating image processing or rendering application using different languages.

In this project, we will try to perform some image transformation using the CV2 version of OpenCV. We will try to make sketches, cartoons, and other transformations on an ordinary image.

➢ Installing OpenCV

In order to install OpenCV, we need to run the following command given below in the command prompt.

```
pip install opencv-python
```

As soon as we install OpenCV we are good to go and start with our image transformation process. Open jupyter notebook and import the required libraries.

3. LIBRARIES USED

MODULE NAME	DESCRIPTION
tkinter	It is a Python binding to the Tk GUI toolkit
cv2	This is a module from the OpenCV library, it will be used for the image processing.
easygui	EasyGUI is a module for very simple, very easy GUI programming in python. Function fileopenbox returns the name of a file
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
SYS	This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. e.g exit().

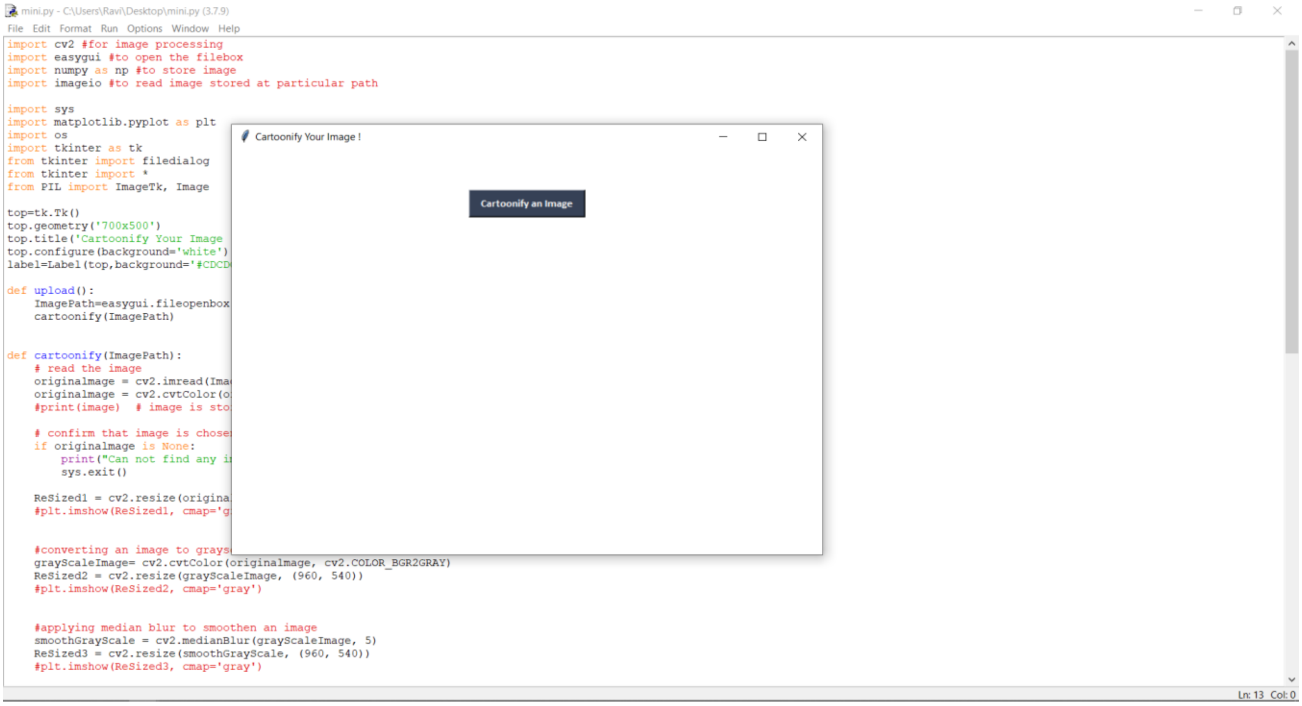
4.SOURCE CODE

```
import cv2 #for image processing
import easygui #to open the filebox
import numpy as np #to store image
import imageio #to read image stored at particular path

import sys
import matplotlib.pyplot as plt
import os
import tkinter as tk
from tkinter import filedialog
from tkinter import *
from PIL import ImageTk, Image

top=tk.Tk()
top.geometry('700x500')
top.title('Cartoonify Your Image !')
top.configure(background='white')
label=Label(top,background='#CDCDCD', font=('calibri',20,'bold'))

def upload():
    ImagePath=easygui.fileopenbox()
    cartoonify(ImagePath)



def cartoonify(ImagePath):
    # read the image
    originalmage = cv2.imread(ImagePath)
    originalmage = cv2.cvtColor(originalmage, cv2.COLOR_BGR2RGB)
    #print(image)  # image is stored in form of numbers

    # confirm that image is chosen
    if originalmage is None:
        print("Can not find any image. Choose appropriate file")
        sys.exit()
```

```
ReSized1 = cv2.resize(originalmage, (960, 540))
#plt.imshow(ReSized1, cmap='gray')


#converting an image to grayscale
grayScaleImage= cv2.cvtColor(originalmage, cv2.COLOR_BGR2GRAY)
ReSized2 = cv2.resize(grayScaleImage, (960, 540))
#plt.imshow(ReSized2, cmap='gray')


#applying median blur to smoothen an image
smoothGrayScale = cv2.medianBlur(grayScaleImage, 5)
ReSized3 = cv2.resize(smoothGrayScale, (960, 540))
#plt.imshow(ReSized3, cmap='gray')

#retrieving the edges for cartoon effect
#by using thresholding technique
getEdge = cv2.adaptiveThreshold(smoothGrayScale, 255,
    cv2.ADAPTIVE_THRESH_MEAN_C,
    cv2.THRESH_BINARY, 9, 9)

ReSized4 = cv2.resize(getEdge, (960, 540))
#plt.imshow(ReSized4, cmap='gray')

#applying bilateral filter to remove noise
#and keep edge sharp as required
colorImage = cv2.bilateralFilter(originalmage, 9, 300, 300)
ReSized5 = cv2.resize(colorImage, (960, 540))
#plt.imshow(ReSized5, cmap='gray')


#masking edged image with our "BEAUTIFY" image
cartoonImage = cv2.bitwise_and(colorImage, colorImage, mask=getEdge)

ReSized6 = cv2.resize(cartoonImage, (960, 540))
#plt.imshow(ReSized6, cmap='gray')
```

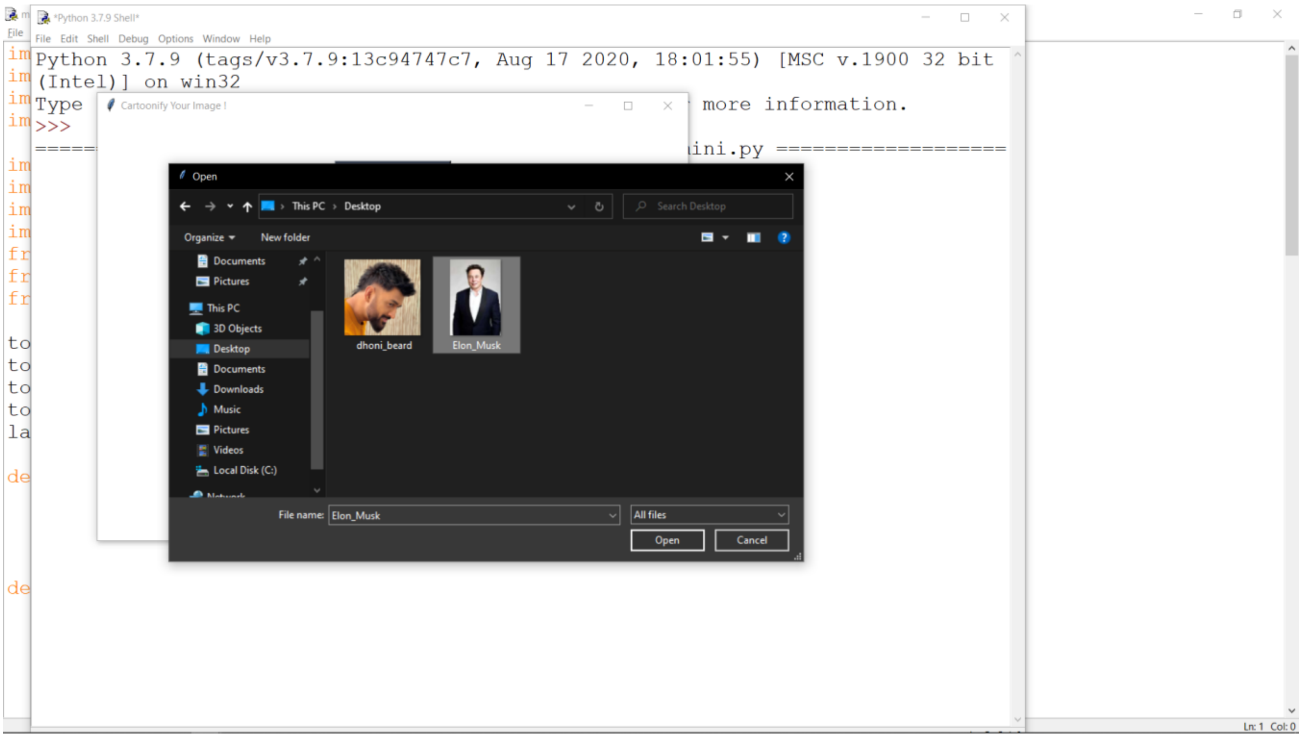
```
    # Plotting the whole transition
    images=[ReSized1, ReSized2, ReSized3, ReSized4, ReSized5, ReSized6]

    fig, axes = plt.subplots(3,2, figsize=(8,8), subplot_kw={'xticks':[], 'yticks':[]}, gridspec_kw=dict(hspace=0.1, wspace=0.1))
    for i, ax in enumerate(axes.flat):
        ax.imshow(images[i], cmap='gray')

    save1=Button(top,text="Save cartoon image",command=lambda: save(ReSized6, ImagePath),padx=30,pady=5)
    save1.configure(background='#364156', foreground='white',font=('calibri',10,'bold'))
    save1.pack(side=TOP,pady=100)

    plt.show()


def save(ReSized6, ImagePath):
    #saving an image using imwrite()
    newName="cartoonified_Image"
    path1 = os.path.dirname(ImagePath)
    extension=os.path.splitext(ImagePath)[1]
    path = os.path.join(path1, newName+extension)
    cv2.imwrite(path, cv2.cvtColor(ReSized6, cv2.COLOR_RGB2BGR))
    I= "Image saved by name " + newName +" at "+ path
    tk.messagebox.showinfo(title=None, message=I)

upload=Button(top,text="Cartoonify an Image",command=upload,padx=10,pady=5)
upload.configure(background='#364156', foreground='white',font=('calibri',10,'bold'))
upload.pack(side=TOP,pady=50)

top.mainloop()
```

5. SCREENSHOTS



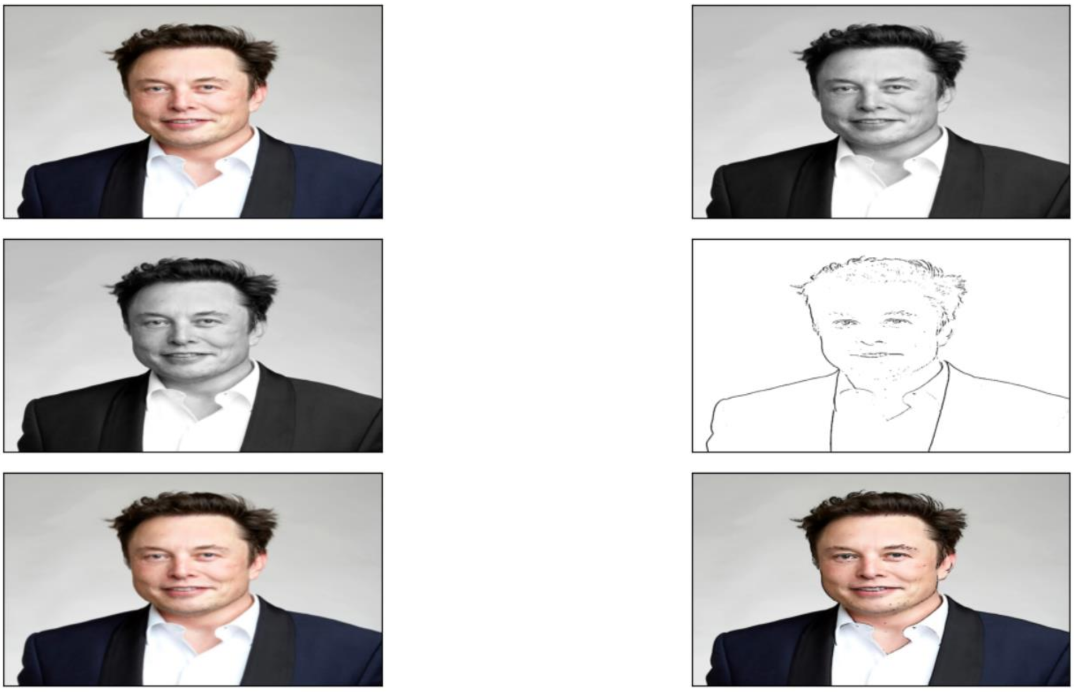
mini.py - C:\Users\Ravi\Desktop\mini.py (3.7.9)
File Edit Format Run Options Window Help
Cartoonify Your Image !
Cartoonify an Image
Ln: 13 Col: 0

```python
import cv2 #for image processing
import easygui #to open the filebox
import numpy as np #to store image
import imageio #to read image stored at particular path

import sys
import matplotlib.pyplot as plt
import os
import tkinter as tk
from tkinter import filedialog
from tkinter import *
from PIL import ImageTk, Image

top=tk.Tk()
top.geometry('700x500')
top.title('Cartoonify Your Image
top.configure(background='white')
label=Label(top,background='#CDCD

def upload():
    ImagePath=easygui.fileopenbox
    cartoonify(ImagePath)


def cartoonify(ImagePath):
    # read the image
    originalmage = cv2.imread(Ima
    originalmage = cv2.cvtColor(o
    #print(image)  # image is sto

    # confirm that image is chose
    if originalmage is None:
        print("Can not find any i
        sys.exit()

    ReSized1 = cv2.resize(origina
    #plt.imshow(ReSized1, cmap='g


    #converting an image to grays
    grayScaleImage= cv2.cvtColor(originalmage, cv2.COLOR_BGR2GRAY)
    ReSized2 = cv2.resize(grayScaleImage, (960, 540))
    #plt.imshow(ReSized2, cmap='gray')


    #applying median blur to smoothen an image
    smoothGrayScale = cv2.medianBlur(grayScaleImage, 5)
    ReSized3 = cv2.resize(smoothGrayScale, (960, 540))
    #plt.imshow(ReSized3, cmap='gray')
```

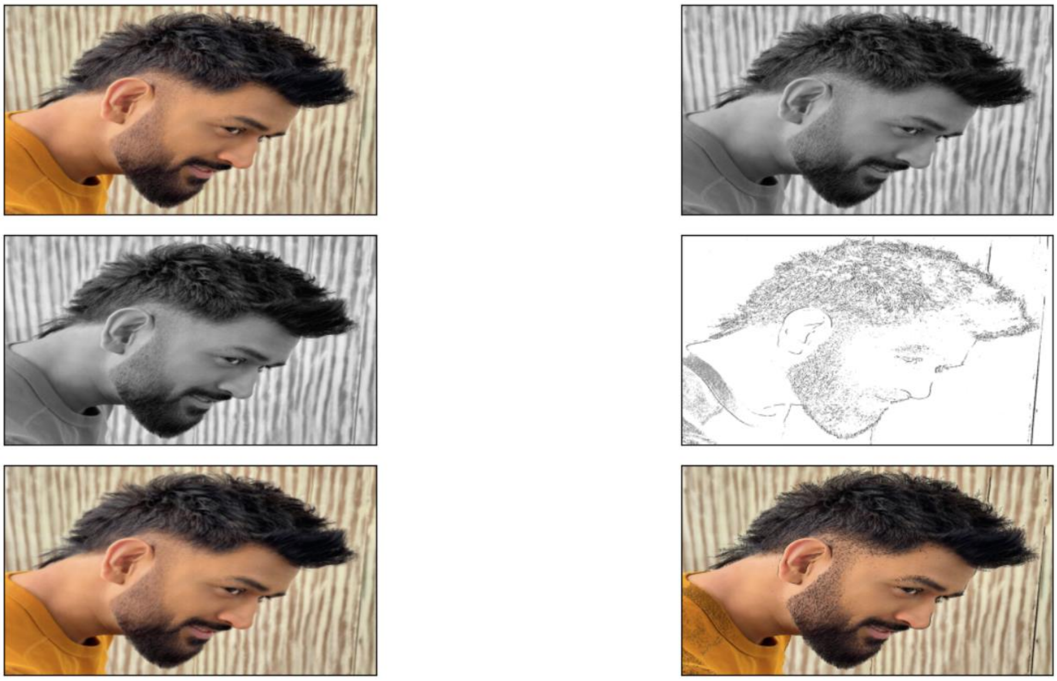


```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:01:55) [MSC v.1900 32 bit
(Intel)] on win32
Type                                                    more information.
>>>
=====                                                  ini.py ===================
```

OUTPUT-1:



OUTPUT-2:



6. REFERENCES

1) **https://www.geeksforgeeks.org/cartooning-an-image-using-opencv-python/**

2) **https://en.wikipedia.org/wiki/OpenCV**