**Image Cartoonifier**

**Mini Project**

Submitted in partial fulfillment of the requirement of University of Mumbai

For the Degree of

**(Computer Engineering)**

**By**

**Shweta Madhukar Kaware TU3F1920005**

**Karishma Apparao Bairi TU3F1920014**

**Abhishek Ashok Eitkar TU3F1920013**

**Under the Guidance of**

**Prof. Seema Biday**



**Department of Computer Engineering**

**TERNA ENGINEERING COLLEGE**

**Plot no.12, Sector-22, Opp. Nerul Railway station,**

**Phase-11, Nerul (w), Navi Mumbai 400706**

**UNIVERSITY OF MUMBAI**



**Terna Engineering College**

**NERUL, NAVI MUMBAI**

CERTIFICATE

*This is to certify that*

**Shweta Madhukar Kaware TU3F1920005**

**Karishma Apparao Bairi TU3F1920014**

**Abhishek Ashok Eitkar TU3F1920013**

*Has satisfactorily completed the requirements of the* ***Mini Project***

*Of subject*

**Artificial Intelligence**

*As prescribed by the* ***University of Mumbai*** *Under the guidance of*

Prof. Seema Biday

**HOD**

**Subject Incharge**

**Index**

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
|  |  |  |
| **Caption** |  | **Page No.** |
| Chapter 1 | Abstract | 04 |
| Chapter 2 | Introduction | 05 |
| Chapter 3 | Literature Survey | 06 |
| Chapter 4 | Methodology | 07 |
| Chapter 5 | Results | 11 |
| Chapter 6 | Conclusion | 12 |
|  | References | 13 |

**Chapter 1**

**Abstract**

In the field of the research processing of an image consisting of identifying an object in an image, identify the dimensions, no of objects, changing the images to blur effect and such effects are highly appreciated in this modern era of media and communication. There are multiple properties in the Image Processing. Each of the property estimates the image to be produced more with essence and sharper image. Each Image is examined to various grid. Each picture element together is viewed as a 2-D Matrix with each of the cell store different pixel values corresponding to each of the picture element.

**Chapter 2**

**Introduction**

The image processing plays a major role in all computers related applications. The image processing appears in many real-life applications, e.g., home security, banking system, education sector, defense system, Railway, and so on.

Cartooning of an image offers a rapid contribution to human interest. Each confined methodology helps in filtering the picture element that forms to an image. There are various factors that enables to produce the essence of an image. The concerns are contrasting and appropriate color mixing, matching between any two pixels connecting two cells, accurate placing of objects together combined to form image features. In the recent times there happened to be drastic changes in ample fields. The uplift of these fields enhances in betterment of the society. In the field of medicine, these processing of images enable to extract the fullest accuracy of the images. Image Processing is widely processed in the medical field such as in the MRI/ET scans. The amount of research in the image processing has helped to acquire early detection of tumors. There plays a vital role in the field of image processing and in the field of Biology. This research bound to save livelihood as early detection can be identified and effective treatment can be started off. These extended concepts have enabled to build better security systems which ensure safety. The security/surveillance systems have managed to build systems depending on the image processing algorithms. The recent technology of fingerprint unlock, face detection unlock has resulted in developing an efficient security. These Biometric systems perhaps have been now installed on to smaller devices as well for the simpler usage. With the recent success apprehended by the social media is duly with the techniques installed to enhance the user experience. E.g. Facebook confines with the auto tag mechanism to automatically suggest the person’s name and not by manually tagging each person on the image. The basic concept in this algorithm involves the technique of converting the RGB colour image to an accurate, cartooned image without multiple filtrations or blurred image without proper facilitation of edge detection. This user interface allows to apply the animation effects. This naturally provides an artistic effect and comics as well with wide range of pictures.

**Chapter 3**

**Literarture Survey**

|  |  |  |
| --- | --- | --- |
| **Year** | **Author Name** | **Paper Name** |
| 2020 | Mohapatra H.; Rath A. K. | A. K. Fundamentals of software engineering: Designed to provide an insight into thesoftware engineering concepts |
| 2019 | Kumar R.; Dey A | A study of neutrosophic shortest path problem. In Neutrosophic Graph Theory and Algorithms |
| 2018 | Kumar R.; Edaltpanh S. A.; Jha S.; Broumi S.; Dey A. | Neutrosophic shortest path problem. Neutrosophic Sets and Systems |
| 2017 | Nirgude V.; Mahapatra H.; Shivarkar S | Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method. Global Journal of Advanced Engineering Technologies and Sciences |
| 2015 | BM M.; Mohapatra H | Human centric software engineering. International Journal of Innovations & Advancement in Computer Science |

**Chapter 4**

**Methodology**

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('400x400')

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=50)

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)

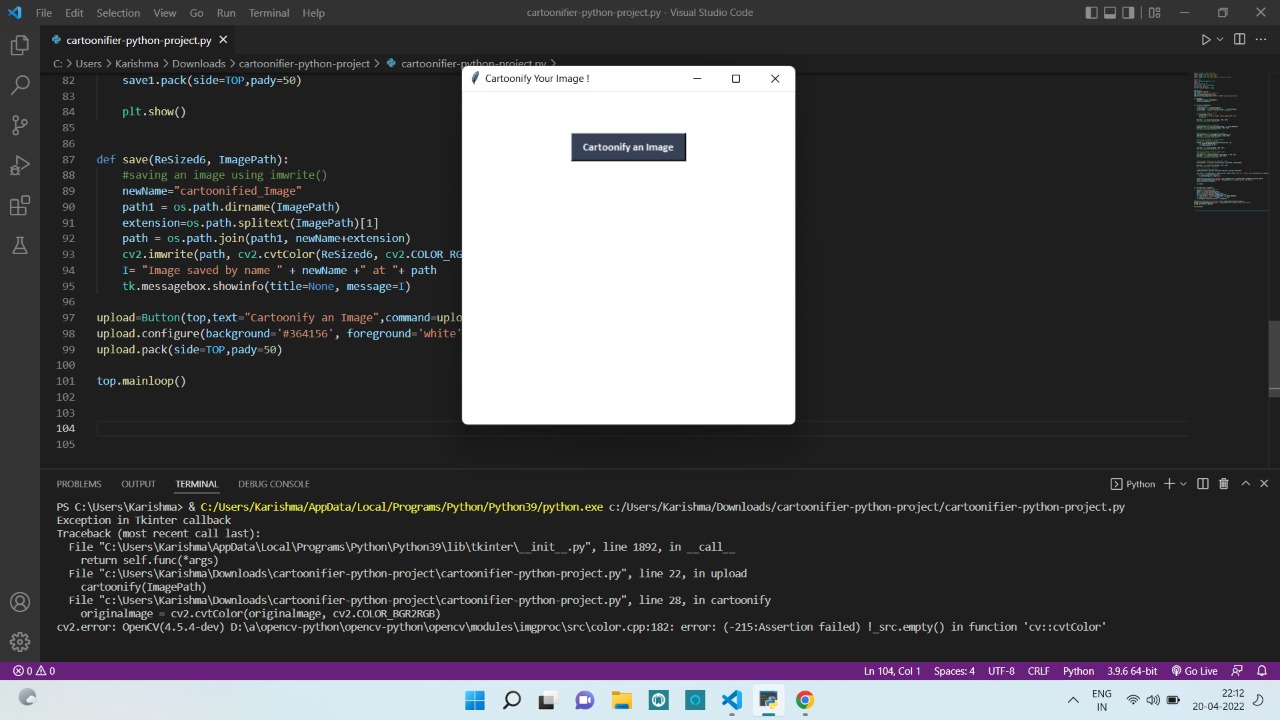
upload=Button(top,text="Cartoonify an Image",command=upload,padx=10,pady=5)

upload.configure(background='#364156', foreground='white',font=('calibri',10,'bold'))

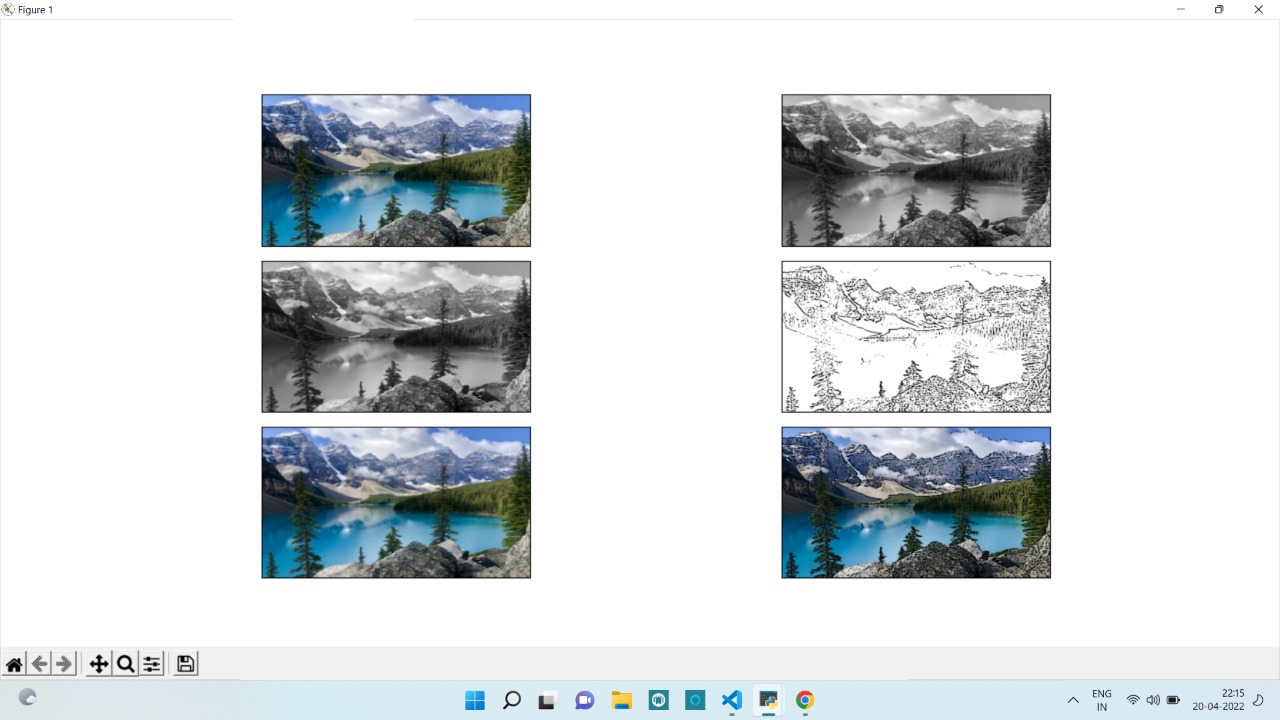
upload.pack(side=TOP,pady=50)

top.mainloop()

**Chapter 5**

**Results**

**Figure 1: Select an image to cartoonify**



**Figure 2: The image is cartoonified**

**Chapter 6**

**Conclusion**

First of all, the basic tools to handle the titled problems of the thesis are incorporated. It includes origin and history of image processing, different types of uncertain environment, existing methods for cartoon imaging. Amid the previous three decades, the topic of image processing has gained vital name and recognition among researchers because of their frequent look in varied and widespread applications within the field of various branches of science and engineering. As an example, image processing is helpful to issues in signature recognition, digital video processing, Remote Sensing and finance. Conclusion and Future Directions. Firstly, we use high-resolution camera to take picture of the internal structure of the wire. Secondly, we use OpenCV image processing functions to implement image pre-processing. Thirdly we use morphological opening and closing operations to segment image because of their blur image edges. The main attraction of the paper is to solve different types of images having one object, two object and three object which can’t be solved by any of the existing methods but can be solved by our proposed method.

**References**

* <https://www.researchgate.net/publication/264132632_A_Study_on_the_Cartoon_Style_in_Image_Contents>
* <https://www.researchgate.net/publication/333686497_Application_of_Cartoon_Like_Effects_to_Actual_Images>
* <https://www.irjet.net/archives/V7/i1/IRJET-V7I1376.pdf>
* <https://doi.org/10.46501/IJMTST0705010>