SNAPAMEND

A

Mini Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

T. Devi Srujana, 1602-20-737-011

D.Lalitha sowjanya, 1602-20-737-019

B. Suresh Kumar, 1602-20-737-051



Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahimbagh, Hyderabad-312022

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Hyderabad-500 031

Department of Information Technology



DECLARATION BY THE CANDIDATE

We, T. Devi Srujana, D. Lalitha Sowjanya and B. Suresh Kumar, bearing hall ticket numbers, 1602-20-737-011, 1602-20-737-019 and 1602-20-737-051, hereby declare that the project report entitled “SNAPAMEND” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

T. Devi Srujana

1602-20-737-011

D. Lalitha Sowjanya

1602-20-737-019

B. Suresh Kumar

1602-20-737-051

(Faculty In-Charge) (Head, Dept of IT)

ACKNOWLEDGMENT

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement.

We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators Divya Madam and RajyaLakshmi Madam, for guiding us in understanding the process of project development & giving us timely suggestions at every phase.

We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

ABSTRACT

Our project SNAPAMEND is a python based Image editing GUI application. It has 5 different editing options which are sketching, Black and white effect, Image Blur, Add border, Add text and inbuilt save, crop and size reducer options. And we also provided a exit button to get out of the application.

The user can easily access any photo of any type it may be a png image, jpg image, jpeg image, etc, from any folder as we have provided an access to every folder of the computer through ask file name method and can save at any location of your computer by using in-built save option provided by matplotlib

We alsohave crop, zoom in and zoom out options are also made available internally. It is a user friendly and accessible application.

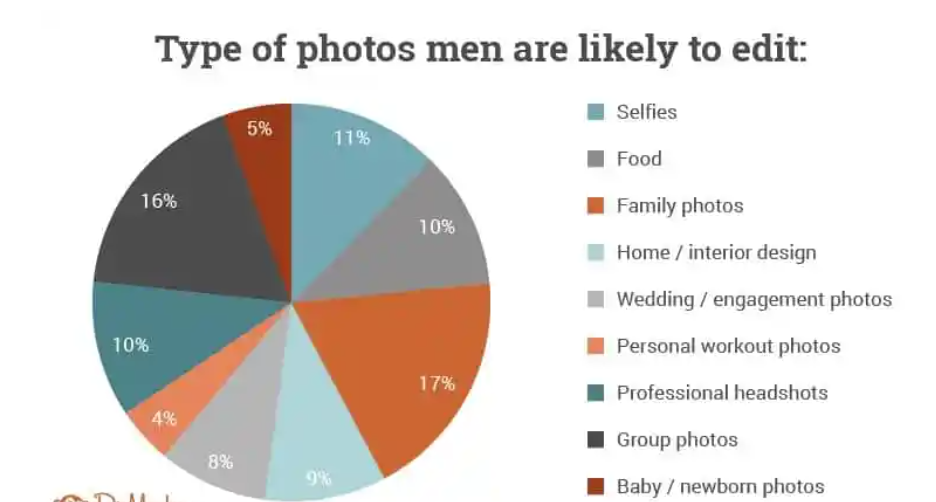
TABLE OF CONTENTS

|  |  |
| --- | --- |
| TITLE | PAGE NO |
| ABSTRACT | 4 |
| INTRODUCTION | 6 |
| TECHNOLOGY | 8 |
| PROPOSED WORK | 9 |
| RESULTS | 23 |
| ADDITIONAL LEARNINGS | 26 |
| CONCLUSION AND  FUTURE WORK | 27 |
| REFERENCES | 28 |

INTRODUCTION

Everyone of us like taking pictures they might be with friends, family, nature pictures, any occasions as we feel them like our memories.

According to a survey, 71% people edit their pictures before posting on any social media platforms.



Chart, pie chart

Description automatically generated

Chart, bar chart

Description automatically generated

OUR WORK

To satisfy these we have come up with out project SNAPAMEND.

We have provided 5 different edit options along with in-built save, crop and enchancer and an exit button.

1. Sketch converter- which converts image to sketch
2. Black and White effect- Adds Black and white effect to a picture
3. Image Blur- Blurs the image
4. Add Border- Adds border to the image with the colour chosen by the user
5. Add Title- Adds title to the image with the coloured text of the colour chosen by the user.

TECHNOLOGY

SOFTWARE TECHNOLOGIES:

Since our project works with images we used:

Opencv – The main source for adding effects to the pictures. Methods like copyboardmaker, blur, puttext etc were used.

Matplotlib – For reading and displaying images. Imread, imshow methods were used for reading and displaying pictures.

Tkinter – For building GUI. Buttons, labels, textfield, askopenfile, etc were used as a part of GUI.

HARDWARE TECHNOLOGIES:

The hardware requirements are quite low and there is no specific hardware required to run this program. We just need a processor with decent multi – tasking capabilities and minimum of 4GB ram.

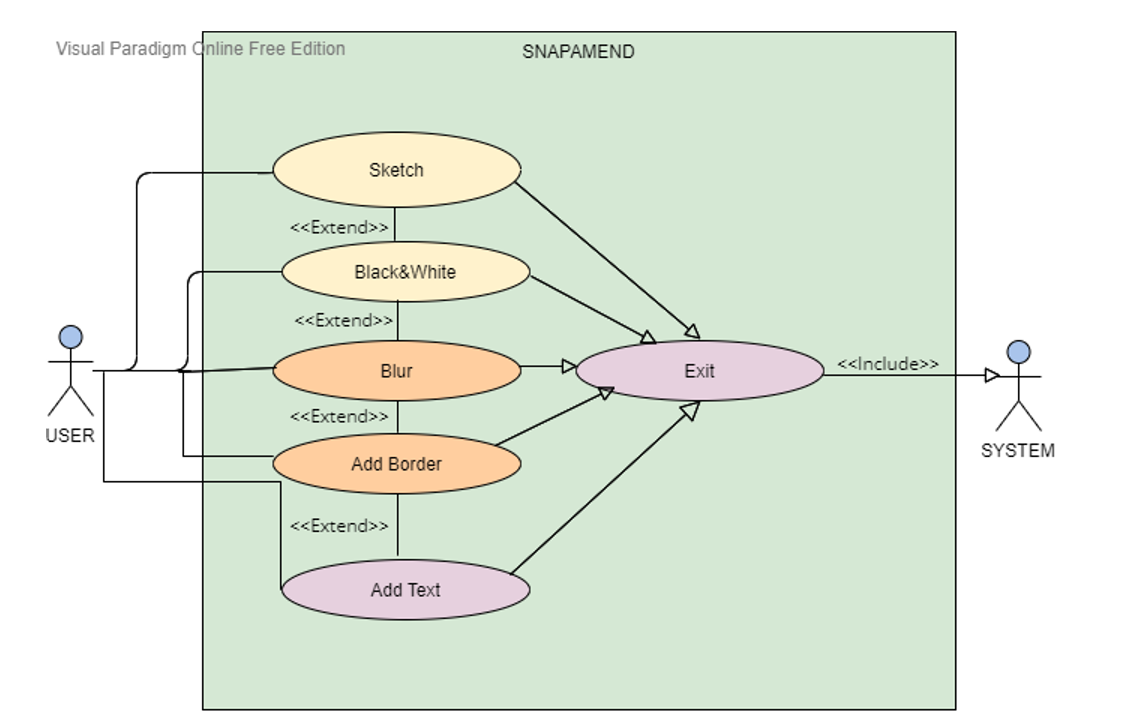
We developed and executed our application on a system running on Intel Core i5 (10th gen). The application was consuming around 70-80% performance of 2Cores out of the six and the average ram usage was around 3GB.

The above figures are taken when there was no other user opened program running in the background.

PROPOSED WORK

DESIGN

USE CASE DIAGRAM



Diagram

Description automatically generated

IMPLEMENTATION

CODE

import tkinter as tk

from tkinter import ttk

from tkinter import \*

from tkinter.filedialog import askopenfilename

import cv2

import matplotlib.pyplot as plot

win=tk.Tk()

win.title("snapAmend")

win.geometry("750x1000")

win.configure(bg='#4a7abc')

lbl=ttk.Label(win,text="SNAPAMEND",font=("Times",20),background='#4a7abc',foreground='pink').grid(column=0,row=0)

def Sketch():

Tk().withdraw()

filename=askopenfilename()

image=plot.imread(filename,1)

gray=cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

inverted=cv2.bitwise\_not(gray)

blurred=cv2.GaussianBlur(inverted,(111,111),0)

invertedBlur=255-blurred

pencilSketch=cv2.divide(gray,invertedBlur,scale=256.0)

ps=cv2.cvtColor(pencilSketch,cv2.COLOR\_BGR2RGB)

plot.imshow(ps)

plot.axis('off')

plot.show()

button1=Button(win,text="Sketch",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Sketch).grid(column=0,row=1)

def BlackandWhite():

Tk().withdraw()

filename = askopenfilename()

image=plot.imread(filename,1)

gray=cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

bw=cv2.cvtColor(gray,cv2.COLOR\_BGR2RGB)

plot.imshow(bw)

plot.axis('off')

plot.show()

button2=Button(win,text="Black&White",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=BlackandWhite).grid(column=0,row=2)

def ImageBlur():

Tk().withdraw()

filename = askopenfilename()

image=plot.imread(filename,1)

Blur=cv2.blur(image,(10,10))

plot.imshow(Blur)

plot.axis('off')

plot.show()

button3=Button(win,text="Blur",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=ImageBlur).grid(column=0,row=3)

def Border():

Tk().withdraw()

filename = askopenfilename()

image=plot.imread(filename,1)

def black():

image1=cv2.copyMakeBorder(image,20,20,20,20,cv2.BORDER\_CONSTANT,None,value=0)

plot.imshow(image1)

plot.axis('off')

plot.show()

def yellow():

image1=cv2.copyMakeBorder(image, 20, 20, 20, 20, cv2.BORDER\_CONSTANT, value=[255, 255, 0])

plot.imshow(image1)

plot.axis('off')

plot.show()

def pink():

image1=cv2.copyMakeBorder(image,20,20,20,20,cv2.BORDER\_CONSTANT,value=[255,0,255])

plot.imshow(image1)

plot.axis('off')

plot.show()

def red():

image1=cv2.copyMakeBorder(image,20,20,20,20,cv2.BORDER\_CONSTANT,value=[255,0,0])

plot.imshow(image1)

plot.axis('off')

plot.show()

def Darkblue():

image1=cv2.copyMakeBorder(image,20,20,20,20,cv2.BORDER\_CONSTANT,value=[0,0,255])

plot.imshow(image1)

plot.axis('off')

plot.show()

def Lightgreen():

image1=cv2.copyMakeBorder(image,20,20,20,20,cv2.BORDER\_CONSTANT,value=[0,255,0])

plot.imshow(image1)

plot.axis('off')

plot.show()

lbl1=ttk.Label(win,text="Choose color",font=("Times",18),background='#4a7abc',foreground='pink').grid(column=0,row=5)

button20=Button(win,text="Black",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=black).grid(column=0,row=6)

button21=Button(win,text="Pink",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=pink).grid(column=1,row=6)

button22=Button(win,text="DarkBlue",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Darkblue).grid(column=0,row=7)

button23=Button(win,text="Red",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=red).grid(column=1,row=7)

button24=Button(win,text="LightGreen",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Lightgreen).grid(column=0,row=8)

button25=Button(win,text="Yellow",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=yellow).grid(column=1,row=8)

button4=Button(win,text="Add Border",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Border).grid(column=0,row=4)

def AddText():

Tk().withdraw()

filename = askopenfilename()

image=plot.imread(filename,1)

font=cv2.FONT\_HERSHEY\_SIMPLEX

def black():

color=(0,0,0)

org=(10,20)

fontScale=1

thickness=4

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

def yellow():

color=(255,255,0)

org=(10,20)

fontScale=1

thickness=2

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

def pink():

color=(255,0,255)

org=(10,20)

fontScale=1

thickness=2

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

def Red():

color=(255,0,0)

org=(10,20)

fontScale=1

thickness=2

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

def Darkblue():

color=(0,0,255)

org=(10,20)

fontScale=1

thickness=2

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

def lightgreen():

color=(0,255,0)

org=(10,20)

fontScale=1

thickness=2

image1=cv2.putText(image,text.get(),org,font,fontScale,color,thickness,cv2.LINE\_AA)

plot.imshow(image1)

plot.axis('off')

plot.show()

button7=Button(win,text="Black",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=black).grid(column=0,row=11)

button8=Button(win,text="Yellow",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=yellow).grid(column=1,row=11)

button9=Button(win,text="Pink",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=pink).grid(column=0,row=12)

button10=Button(win,text="Red",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Red).grid(column=1,row=12)

button11=Button(win,text="DarkBlue",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=Darkblue).grid(column=0,row=13)

button12=Button(win,text="LightGreen",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=lightgreen).grid(column=1,row=13)

lbl2=ttk.Label(win,text="Enter title",font=("Times",20),background='#4a7abc',foreground='pink').grid(column=0,row=10)

text=StringVar()

txt=ttk.Entry(win,width=15,textvariable=text).grid(column=1,row=10)

button5=Button(win,text="Add Text",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=AddText).grid(column=0,row=9)

def close():

win.destroy()

win.quit()

button6=Button(win,text="Exit",font=('Times',12),height=0,width=10,padx=10,pady=10,bg='#4a7abc',fg='yellow',activebackground='green',activeforeground='white',command=close).grid(column=0,row=14)

win.mainloop()

GITHUB LINK: <https://github.com/Lalitha-Sowjanya/SNAPAMEND.git>

TESTING

Firstly after running the code we will get a layout appearing on our screen like this:

Graphical user interface

Description automatically generated with medium confidence

CASE-1:

And let the option chosen by the user be sketch then it will ask us to choose the picture any where on pc.

Graphical user interface, application

Description automatically generated

After choosing the picture it will sketch the image and displays it.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

There will be an in-built save option if we want to save it, we can save it any location on our pc.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

CASE-2:

Now let the option choose be Add Border,

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Now the user will have a choice to choose a colour for the border.

Let the colour chosen be blue.

Graphical user interface, application, Teams

Description automatically generated

RESULT

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

A picture containing text, screenshot, monitor, indoor

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word, PowerPoint

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

ADDITIONAL LEARNINGS

Apart from the deep understanding of python programming language we also learnt how to access libraries like opencv, matplotlib, tkinter deeply. We got to know how to access ask openfile to access any file on pc, how to set buttons, labels, textfields using tkinter, how to add effects to image using opencv

We have learned to meet deadlines. Throughout these months, we successfully completed and submitted the mini project abstract, the design document, the different modules in the project and not the report too. Thinking out of the box. We were constantly thinking about how to make this project better than the existing ones and user friendly.

CONCLUSION AND FUTURE WORK

Finally we like to conclude that our mini project SNAPAMEND is a good application for those who like editing pictures and adding filters to the pictures

Our project has easy access to any file on pc and can save the edited picture at any location.

In future we would like to add more options like cartoonize, background blur and even exchange of pictures.

We would also like to implement exchanging of the edited pictures like in snapchat.

REFERENCES

<https://www.geeksforgeeks.org/opencv-overview/>

<https://www.pyimagesearch.com/2016/05/23/opencv-with-tkinter/>

<https://www.geeksforgeeks.org/how-to-display-an-opencv-image-in-python-with-matplotlib/>