from tkinter import \*  
import tkinter.filedialog as tk  
import tkinter.messagebox as tk2  
import numpy as np  
import cv2  
  
  
class Application(Frame):  
  
 def \_\_init\_\_(self, master):  
 super(Application, self).\_\_init\_\_(master)  
 self.img = ""  
 self.grid(rowspan=5, columnspan=4)  
 self.addButton = Button(self, text='Add', command=self.add)  
 self.generateButton = Button(self, text='Generate', command=self.generate)  
 self.generateButton.grid(row=4, column=0)  
 self.addButton.grid(row=4, column=2)  
 self.pack()  
  
 def add(self):  
 file = tk.askopenfilenames(initialdir='C:/Users/babbu/Downloads')  
 imgTuple = root.splitlist(file) # turn user's opened filenames into tuple  
 self.img = "".join(list(imgTuple))  
  
 def generate(self):  
 img\_obj = cv2.imread(self.img)  
  
 # img\_obj.shape -- to check the shape of image  
  
 scale\_percent = 0.60  
 width = int(img\_obj.shape[1] \* scale\_percent)  
 height = int(img\_obj.shape[0] \* scale\_percent)  
  
 dim = (width, height)  
 resized = cv2.resize(img\_obj, dim, interpolation=cv2.INTER\_AREA) # resizing the image  
  
 kernel\_sharpening = np.array([[-1, -1, -1],  
 [-1, 9, -1],  
 [-1, -1, -1]])  
 sharpened = cv2.filter2D(resized, -1, kernel\_sharpening) # shape the image  
  
 gray = cv2.cvtColor(sharpened, cv2.COLOR\_BGR2GRAY) # convert in black and white  
 object\_detection = cv2.cvtColor(sharpened, cv2.COLOR\_BGR2HSV) # convert in image detection formate  
  
 inv = 255 - gray # convert in inverse form  
 gauss = cv2.GaussianBlur(inv, ksize=(15, 15), sigmaX=0, sigmaY=0) # convert in gauss form  
  
 pencil = cv2.divide(gray, 255 - gauss, scale=256)  
  
 # to display these four images  
 cv2.imshow('resized', resized)  
 # cv2.imshow('sharp',sharpened)  
 # cv2.imshow("gray", gray)  
 cv2.imshow('pencile', pencil)  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
  
root = Tk()  
root.title('Sketch-Maker')  
root.geometry('500x200')  
app = Application(root)  
app.mainloop()