MATHEMATICS WITH COMPUTER GRAPHICS (HS09)

IMAGE PROCESSING

SC32303

SEMESTER 1 2022/2023

INDIVIDUAL ASSIGNMENT 2

LECTURER:

PROFESSOR ABDULLAH BIN BADE

NAME:

YAP JIA JUN

MATRIC NUMBER:

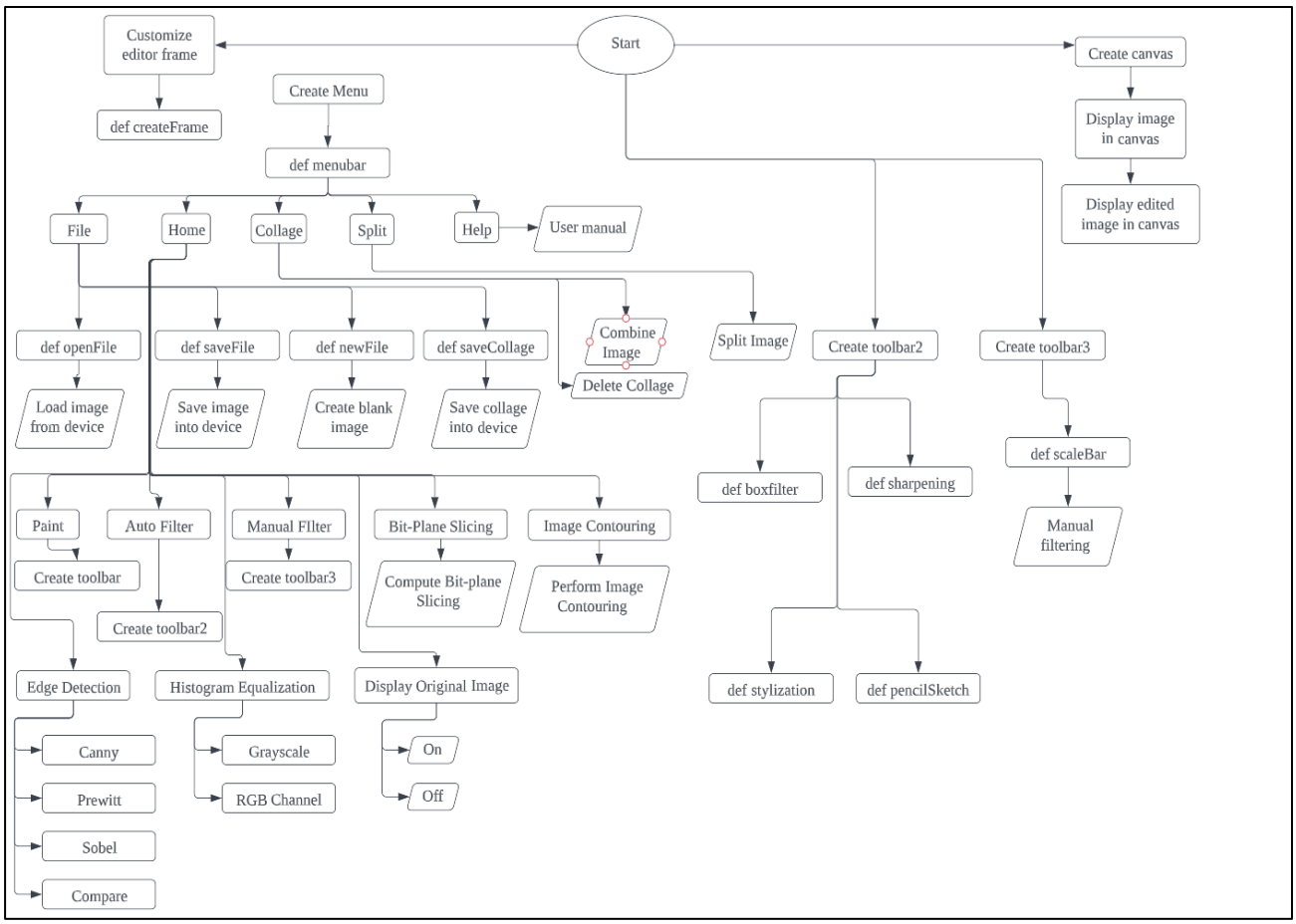
BS20110220

Table of Content

1.0 Introduction

In this assignment 2, I upgraded my assignment 1 2D interactive image editor. Basically, I had further enhanced my editor by adding more features such as split and combine several images, compute histogram equalization for both single channel (grayscale) and RGB channels, adding more filters to enhance quality of image, perform bit-plane slicing, adding few types of edge detection techniques and perform image contouring process. Furthermore, user also can manually adjust the filtering of image in my editor. Besides, I also added a function for user to compare their edited image with original image where user can switch on or off for it. In order for user can differentiate each type of filters, I implement comparison function in my editor so that user can clearly see the difference among all the filters.

2.0 Flow Diagram / Algorithm



3.0 System Architecture

In order to create all the filters and functions in the requirements, I have used several libraries such as OpenCV, matplotlib and numpy. For split and combine image, I have changed my function from loading one image once only to several images and the image after split and combine can be save as well. With the help of matplotlib, my editor enable user to compare between all the filtered image and edited image.

4.0 Strength and Uniqueness

In order to build a user-friendly interface, I added more menu for user can use the function in my editor based on each category. For example, when user just want to do drawing and painting, user can choose paint from menu home and the functions of paint will be displayed. But when user want to do filtering image, a different function menu will be replaced the previous menu. This ensure that user will not get confuse with all the functions in my editor.

Next, an overall comparison among all the filtered and edited image is available in this editor so that user does not need to keep on changing the filtering to see the differences. User can turn on and off any time during editing image to compare the differences. Besides auto filtering, user can also filter image manually by adjusting the value with scale bar provided.

5.0 Sample Source Code

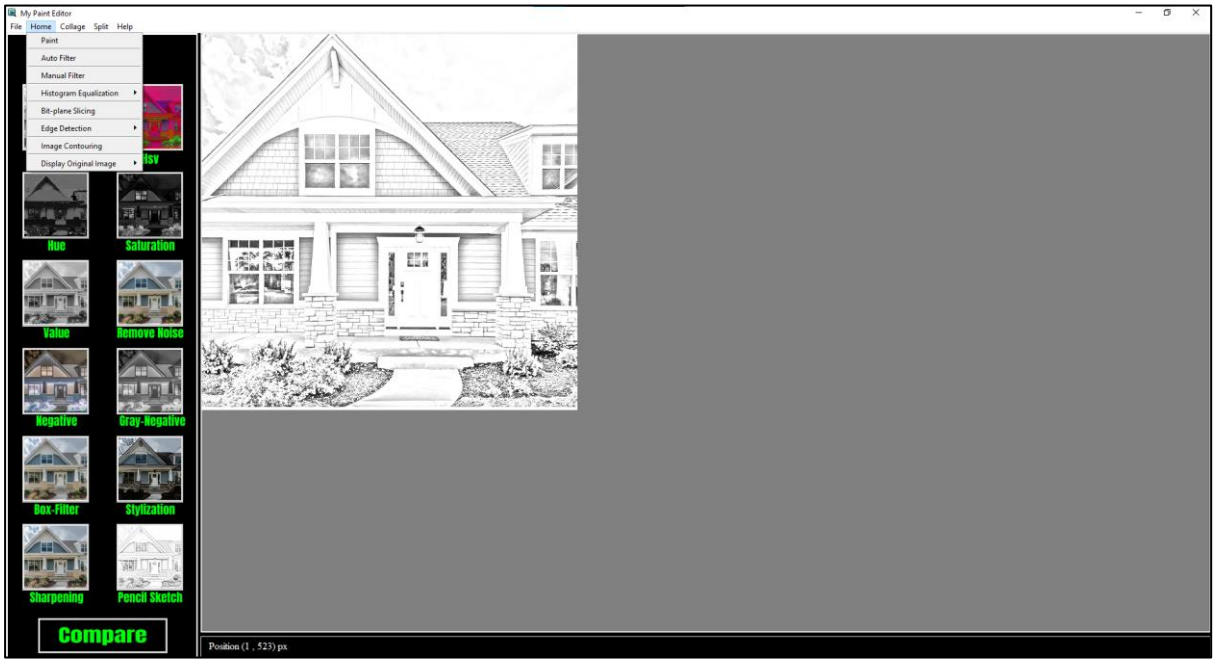
```
855
856     def boxFilter(self):
857         self.editedImage=cv2.boxFilter(self.editImage,-1,(7,7))
858         self.showImage(self.editedImage)
859
860     def enchanment(self):
861         img = self.editImage
862         blur = cv2.GaussianBlur(self.editImage,(9,9),0)
863         self.editedImage = cv2.addWeighted(img, 2.5, blur, -1.5, 0)
864         self.showImage(self.editedImage)
865
866     def stylisation(self):
867         self.editedImage=cv2.stylization(self.editImage, sigma_r=0.25, sigma_s=150)
868         self.showImage(self.editedImage)
869
870     def pencilSketch(self):
871         grey = cv2.cvtColor(self.editImage, cv2.COLOR_BGR2GRAY)
872         invert = cv2.bitwise_not(grey)
873         blur = cv2.GaussianBlur(invert, (49,49),0)
874         invertblur = cv2.bitwise_not(blur)
875         self.editedImage = cv2.divide(grey, invertblur, scale=256.0)
876         self.showImage(self.editedImage)
877
```

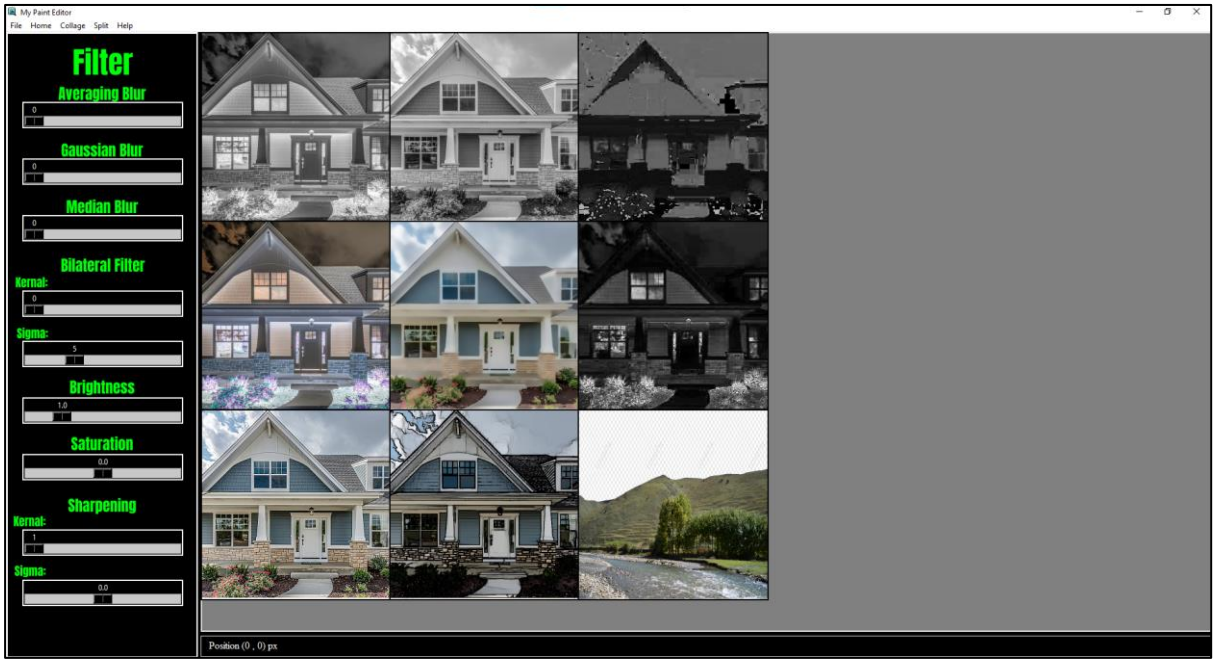
```
984     def averaging(self, value):
985         value = int(value)
986         if value % 2 == 0:
987             value += 1
988         self.editImage = cv2.blur(self.oriImage, (value, value))
989         self.editedImage = self.editImage
990         self.showImage(self.editedImage)
991
992     def gaussian(self, value):
993         value = int(value)
994         if value % 2 == 0:
995             value += 1
996         self.editImage = cv2.GaussianBlur(self.oriImage, (value, value), 0)
997         self.editedImage = self.editImage
998         self.showImage(self.editedImage)
999
1000    def median(self, value):
1001        value = int(value)
1002        if value % 2 == 0:
1003            value += 1
1004        self.editImage = cv2.medianBlur(self.oriImage, value)
1005        self.editedImage = self.editImage
1006        self.showImage(self.editedImage)
1007
1008    def bilateral(self, value):
1009        kernal = self.bilateral2_slider.get()
1010        value = self.bilateral_slider.get()
1011        if kernal % 2 == 0:
1012            kernal += 1
1013
1014        self.editImage = cv2.bilateralFilter(self.oriImage,kernal,value,value)
1015        self.editedImage = self.editImage
1016        self.showImage(self.editedImage)
1017
1018    def brightness(self, value):
1019        self.editedImage = cv2.convertScaleAbs(self.editImage, alpha=self.brightness_slider.get())
1020        self.showImage(self.editedImage)
```

```
    def collage12(self):

        self.filename = fd.askopenfilename(multiple=True,initialdir="envproject/Assignment2/images",
            title="Select A File", filetypes=(("png files", "*.png"),("all files","*")) )

        self.filenamelist.append(self.filename)

        image1 = cv2.imread(self.filename[0])
        image1 = cv2.cvtColor(image1,cv2.COLOR_BGR2RGB)
        image1 = cv2.resize(image1,(450,900))
        self.img1 = ImageTk.PhotoImage(Image.fromarray(image1))
        self.imageLabel1 = Label(self.Dpanel,image=self.img1,bg='black')
        self.imageLabel1.place(x=0,y=0)

        image2 = cv2.imread(self.filename[1])
        image2 = cv2.cvtColor(image2,cv2.COLOR_BGR2RGB)
        image2 = cv2.resize(image2,(450,900))
        self.img2 = ImageTk.PhotoImage(Image.fromarray(image2))
        self.imageLabel2 = Label(self.Dpanel,image=self.img2,bg='black')
        self.imageLabel2.place(x=450,y=0)

    def collage21(self):

        self.filename = fd.askopenfilename(multiple=True,initialdir="envproject/Assignment2/images",
            title="Select A File", filetypes=(("png files", "*.png"),("all files","*")) )

        self.filenamelist.append(self.filename)

        image1 = cv2.imread(self.filename[0])
        image1 = cv2.cvtColor(image1,cv2.COLOR_BGR2RGB)
        image1 = cv2.resize(image1,(900,450))
        self.img1 = ImageTk.PhotoImage(Image.fromarray(image1))
        self.imageLabel1 = Label(self.Dpanel,image=self.img1,bg='black')
        self.imageLabel1.place(x=0,y=0)

        image2 = cv2.imread(self.filename[1])
```

```
    def hisEqua1(self):
        grayscale = cv2.cvtColor(self.editImage,cv2.COLOR_BGR2GRAY)
        hisEqualizedImg = cv2.equalizeHist(grayscale)
        histogram1 = cv2.calcHist([grayscale], [0], None, [256], [0, 256])
        histogram2 = cv2.calcHist([hisEqualizedImg], [0], None, [256], [0, 256])
        plt.subplot(2,2,1)
        plt.imshow(grayscale, cmap='gray')
        plt.title("Grayscale")
        plt.xticks([])
        plt.yticks([])

        plt.subplot(2,2,2)
        plt.imshow(hisEqualizedImg, cmap='gray')
        plt.title("Equalized")
        plt.xticks([])
        plt.yticks([])

        plt.subplot(2,2,3)
        plt.plot(histogram1, color='k')
        plt.title("Grayscale Histogram")

        plt.subplot(2,2,4)
        plt.plot(histogram2, color='k')
        plt.title("Equalized Histogram")
        plt.show()

    def hisEqua2(self):
        original = cv2.cvtColor(self.editImage,cv2.COLOR_BGR2RGB)
        plt.subplot(2,2,1)
        plt.title("Original Image")
        plt.imshow(original)

        for i, col in enumerate(['b', 'g', 'r']):
            histogram1 = cv2.calcHist([original], [i], None, [256], [0, 256])
            plt.subplot(2,2,3)
            plt.title("Original Histogram")
```

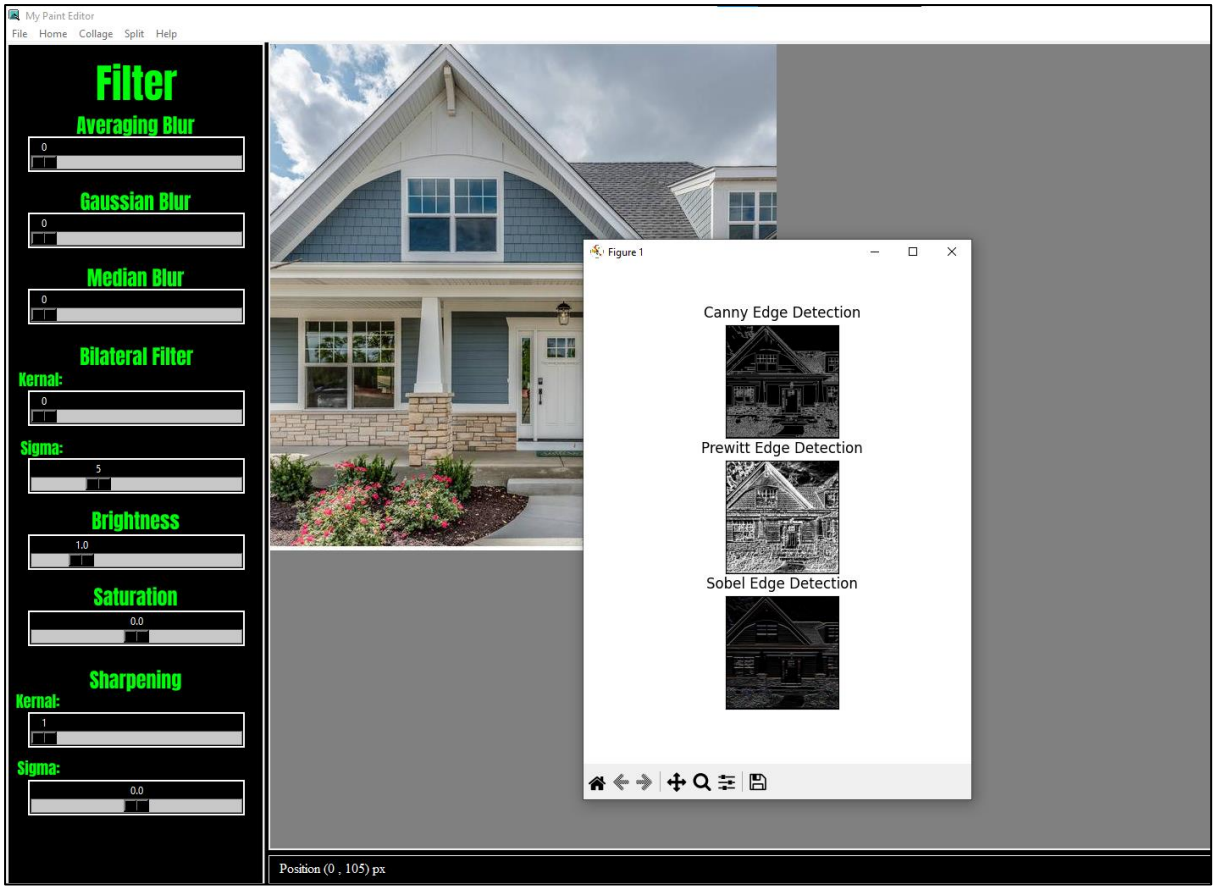
6.0 Sample Outputs






My Paint Editor
File Home Collage Split Help
Filter
Averaging Blur
Gaussian Blur
Median Blur
Bilateral Filter
Kernal:
Sigma:
Brightness
Saturation
Sharpening
Kernal:
Sigma:
Figure 1
Original Image
Equalized Image
Original Histogram
Equalized Histogram
Position (127 , 91) px


My Paint Editor
File Home Collage Split Help
Filter
Averaging Blur
Figure 1
Bit-plane 1
Bit-plane 2
Bit-plane 3
Bit-plane 4
Bit-plane 5
Bit-plane 6
Bit-plane 7
Bit-plane 8
Grayscale
x=393. y=329.
[0.0]
Position (606 , 504) px


My Paint Editor
File Home Collage Split Help
Filter
Averaging Blur
Gaussian Blur
Median Blur
Bilateral Filter
Kernal:
Sigma:
Brightness
Saturation
Sharpening
Kernal:
Sigma:
Figure 1
Canny Edge Detection
Prewitt Edge Detection
Sobel Edge Detection
Position (0 , 105) px


My Paint Editor
File Home Collage Split Help
Filter
Averaging Blur
Gaussian Blur
Median Blur
Bilateral Filter
Kernal:
Sigma:
Brightness
Saturation
Sharpening
Kernal:
Sigma:
Position (23 , 1) px