import tkinter as tk

from tkinter.filedialog import askopenfilename

from PIL import ImageTk,Image

# importing packages

#from face\_landmark import faceLandmark

#from face\_landmark\_makeup import faceMakeup

#from img\_face\_detection import imageFaceDetection

#from image\_face\_recognition import imageFaceRecognition

#from image\_age\_gender import imageAgeGenderDetect

#from realtime\_face\_detection\_blur import realtimeFaceBlur

#from realtime\_face\_detection import realtimeFaceDetect

#from realtime\_face\_recognition import realtimeFaceRecognition

class main\_page:

    def \_\_init\_\_ (self, window):

        self.window = window

        self.window.title("Enhancement of Ancient Manuscripts")

        self.window.geometry("1350x700")

        self.window.resizable(width=False, height=False)

        # Background Image

        # self.bg\_icon = ImageTk.PhotoImage(file = "Program\_images/bg\_image.jpg")

        # bg\_lbl = Label(self.window, image = "Program\_images/bg\_image.jpg")

        # All variables here

        self.topics\_names = ["Background Normalisation", "Image Face Detection", "Image Face Recognition", "Image Age and Gender",

                        "Realtime Face Blur", "Realtime Face Detection", "Realtime  Face Recognition"]

        self.btn\_commands = [lambda: screen.background\_normalisation(screen.dummy()),

                        lambda: screen.image\_face\_detection(screen.dummy()),

                        lambda: screen.image\_face\_recognition(screen.dummy()),

                        lambda: screen.image\_age\_gender(screen.dummy()),

                        lambda: screen.realtime\_face\_blur(screen.dummy()),

                        lambda: screen.realtime\_face\_detection(screen.dummy()),

                        lambda: screen.realtime\_face\_recognition(screen.dummy())]

        self.image\_path = "Images.jpg"

        # ---- Lables -----------------------

        title = tk.Label(self.window, text="Enhancement of Ancient Manuscripts", font=("times new roman", 40, "bold"),

                      bg="#BFCEFF", fg="red", bd=10, relief=tk.GROOVE)

        title.place(x=0, y=0, relwidth=1)

    def open\_file(self):

        """Open and select image for processing"""

        filepath = askopenfilename(filetypes=[("Image Files", ("\*.jpg", "\*.png")), ("All Files", "\*.\*")])

        if not filepath:

            return

        return filepath

    def dummy(self):

        # frame 1

        frm1 = tk.Frame(self.window, bd=4, relief=tk.RIDGE, bg="white")

        frm1.place(x=25, y=160, width=645, height=510)

        # frame 2

        frm2 = tk.Frame(self.window, bd=4, relief=tk.RIDGE, bg="white")

        frm2.place(x=670, y=160, width=645, height=510)

        return (frm1, frm2)

    def lowerPart(self):

        frm = tk.Frame(self.window, bd=4, relief=tk.RIDGE, bg="#007AA8")

        frm.place(x=20, y=155, width=1300, height=520)

    def options\_bar(self):

        # Frame options

        frm\_options = tk.Frame(self.window, bd=4, relief=tk.RIDGE, bg="#007AA8")

        frm\_options.place(x=20, y=100, width=1300, height=50)

        # making all option buttons

        for index, text in enumerate(self.topics\_names):

            # creating label widget with the text form

            options\_btn = tk.Button(master=frm\_options, text=text, width=25, height=2, command=self.btn\_commands[index])

            options\_btn.grid(row=0, column=index, sticky=tk.NS)

        # calling lower part

        screen.lowerPart()

        #screen.image\_landmark(screen.dummy())

    def btn\_select\_image(self, frm2):

        self.image\_path = screen.open\_file()

        if self.image\_path:

            image\_name = self.image\_path.split("/")[-1]

            # frm 2

            tk.Label(master=frm2, text=image\_name, bg="#007AA8", fg="white",

                     font=("times new roman", 20, "bold")).place(x=0, y=0, relwidth=1)

            # image

            photo = Image.open(self.image\_path)

            photo = photo.resize((600, 400), Image.ANTIALIAS)

            photo.save("ArtWrk.ppm")

            img = ImageTk.PhotoImage(Image.open("ArtWrk.ppm"))

            image\_frame = tk.Label(frm2, bd=4, image=img)

            image\_frame.image = img

            image\_frame.place(x=0, y=40)

    def btn\_run(self,image\_path,index):

        if index == 0:

            landmark = faceLandmark(image\_path)

            landmark.landmarks()

        elif index == 1:

            detect = imageFaceDetection(self.image\_path)

            detect.detect\_face()

        elif index == 2:

            recognition = imageFaceRecognition(self.image\_path)

            recognition.encoding()

        elif index == 3:

            detect = imageAgeGenderDetect(self.image\_path)

            detect.detect()

        elif index == 4:

            face\_blur = realtimeFaceBlur()

            face\_blur.blur()

        elif index == 5:

            detect = realtimeFaceDetect()

            detect.detect()

        elif index == 6:

            recognise = realtimeFaceRecognition()

            recognise.recognise()

        elif index == 7:

            makeup = faceMakeup(self.image\_path)

            makeup.makeup()

    def image\_landmark(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Image Landmark", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(fill=tk.X)

        # frm 2

        tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white", font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        btn\_select = tk.Button(master=btns\_frame, text="Select Image", width=25, height=2,

                               command=lambda: screen.btn\_select\_image(frm2))

        btn\_select.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

        btn\_run = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                            command=lambda: screen.btn\_run(self.image\_path, 0))

        btn\_run.grid(row=0, column=1, sticky=tk.NS, padx=15, pady=10)

        # makeup

        makeup\_label = tk.Label(master=btns\_frame, text="Makeup Selected Image", bg="white", fg="Black", font=("times new roman", 20, "bold"))

        makeup\_label.grid(row=1, column =1,sticky=tk.N, padx=15, pady=10)

        # run makeup

        btn\_run = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                            command=lambda: screen.btn\_run(self.image\_path, 7))

        btn\_run.grid(row=2, column=1, sticky=tk.NS, padx=15, pady=10)

    def image\_face\_detection(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Image Face Detection", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # frm 2

        tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

                 font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Select Image", width=25, height=2,

                                command=lambda: screen.btn\_select\_image(frm2))

        options\_btn.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 1))

        options\_btn.grid(row=0, column=1, sticky=tk.NS, padx=15, pady=10)

    def image\_face\_recognition(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Image Face Recognition", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # frm 2

        tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

                 font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Select Image", width=25, height=2,

                                command=lambda: screen.btn\_select\_image(frm2))

        options\_btn.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 2))

        options\_btn.grid(row=0, column=1, sticky=tk.NS, padx=15, pady=10)

    def image\_age\_gender(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Image Age Gender Detection", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # frm 2

        tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

                 font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Select Image", width=25, height=2,

                                command=lambda: screen.btn\_select\_image(frm2))

        options\_btn.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 3))

        options\_btn.grid(row=0, column=1, sticky=tk.NS, padx=15, pady=10)

    def realtime\_face\_blur(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Realtime Face Blur", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # # frm 2

        # tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

        #          font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 4))

        options\_btn.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

    def realtime\_face\_detection(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Realtime Face Detection", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # frm 2

        # tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

        #          font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 5))

        options\_btn.grid(row=0, column=0, sticky=tk.NS, padx=15, pady=10)

    def realtime\_face\_recognition(self, dummy):

        frm1 = dummy[0]

        frm2 = dummy[1]

        # frm 1

        tk.Label(master=frm1, text="Realtime Face Recognition", bg="#007AA8", fg="white", font=("times new roman", 20)).pack(

            fill=tk.X)

        # frm 2

        # tk.Label(master=frm2, text="Selected Image", bg="#007AA8", fg="white",

        #          font=("times new roman", 20, "bold")).pack(fill=tk.X)

        # creating btns

        btns\_frame = tk.Frame(master=frm1, relief=tk.RIDGE, bg="white")

        btns\_frame.pack(fill=tk.BOTH)

        options\_btn = tk.Button(master=btns\_frame, text="Run", width=25, height=2,

                                command=lambda: screen.btn\_run(self.image\_path, 6))

        options\_btn.grid(row=0, column=1, sticky=tk.NS, padx=15, pady=10)

window = tk.Tk()

screen = main\_page(window)

screen.options\_bar()

window.mainloop()