PROJECT NAME : Snap Chat Filters

AIM

To create tkinter interface to apply snapchat filters to the photos which was selected by user and also in live cam.

PROJECT OUTCOMES

* How face detections and landmarks are used in opencv.
* This project helps to know how to link frontend (tkinter) and backend.

APPLIED FILTERS

1) Face mesh filter on images
2) Face mesh filter on set of images
3) tug life filter on image
4) Face mesh filter on live cam
5) Animated Eyes and fire smoke from mouth filter on live cam
6) Animated Eyes, Face mesh and fire smoke from mouth filter on live cam
7) tug life filter on live cam

1 . FRONTEND

```
Importing the required modulus

Required modulus are:
1. tkinter module for interface
2. dbm module for storing and manipulate data
3. filter files (backend) (Face_mesh_filter_image,
Face_mesh_filter_set_of_images,
                          tug_life_filter_image, Face_mesh_live,
Eyes_smoke_filter_cam
                          Face_mesh_eyes_smoke, tug_life_filter_cam)

Frontend can be seen in fvbb.py file
```

Class name `main`

Constructor of class is to `login` the user by username and password.

If user is new then user will `signup` the details by entring username and required password. Singup will be seen in `appen` function.

Details will be added in `database` file of name `data` file by function `add` in class main.

If user is new but user try to login before signup then there will be display of message `Invaild Credentials` in new window. This can be seen in function `Home` .

If username is present in database but password is `not matched` then there will be a display of message `*Incorrect password` . This can be seen in function `Home` .

If username is `matchs` with corresponding password then home page of `snapchat filter` window will be open and also `destroys` login window. This can be seen in function `Home`

```
def __init__(self):          # class constructor

    self.Authen_root = Tk()

    self.x = self.Authen_root.winfo_screenwidth()
    self.y = self.Authen_root.winfo_screenheight()
    ww = 500
    wh = 500
    self.Authen_root.geometry(f"{ww}x{wh}+{(self.x-ww)//2}+{(self.y-wh)//2}")

    self.Authen_root.title("Login")
    self.Authen_root.configure(bg='yellow')
    self.Authen_root.resizable(width=False, height=False)

    self.blank()

    fr = Frame(self.Authen_root,bg='yellow')

    graphical_image = PhotoImage(file="Touch.png")
    canvas = Canvas(self.Authen_root, width=300, height=200,bg='yellow',
                    highlightbackground='yellow',highlightthickness=5)
    canvas.create_image(0, 0, anchor=NW, image=graphical_image)
    canvas.pack()

    self.blank()

    Label(fr, text="Username : ",font=('Times',18),bg='yellow').grid(row=0,
column=0)
    Label(fr, text="Password : ",font=('Times',18),bg='yellow').grid(row=1,
column=0)

    self.entry_username = Entry(fr)
    self.entry_username.grid(row=0, column=1)
    self.entry_password = Entry(fr, show="*")
    self.entry_password.grid(row=1, column=1)

    Button(fr, text="Login", command=self.Home).grid(row=2,column=1,sticky='ns')
    Label(fr, text="",bg='yellow').grid(row=3, column=0)
    Label(fr,text='Don\'t have an account?',font=('Helvetica',
12),bg='yellow').grid(row = 4,column = 0)
    Button(fr,text='Sign up',command=self.appen).grid(row = 4,column=1,sticky=W)

    fr.pack()
    self.Authen_root.mainloop()
```

In `append` function new window is created with title Sign up

Label `username` , `password` is created and respective space is created for entry

Password is hidden by `*` for security purpose

After clicking the button Signup, it will redirect to `add` function

```
def appen(self):
    self.data = Tk()
    self.data.title('Sign up')
    self.data.configure(bg='yellow')
    ww = 500
    wh = 500
    self.data.geometry(f"{ww}x{wh}+{(self.x-ww)//2}+{(self.y-wh)//2}")


    fr1 = Frame(self.data,bg='yellow')
    for i in range(3):
        Label(fr1,text="",bg='yellow').grid(row=i,column=0)

    Label(fr1, text="Username : ",font=('Times',18),bg='yellow').grid(row=3,
column=0)
    self.entry_user = Entry(fr1)
    self.entry_user.grid(row=3, column=1)

    Label(fr1, text="Password : ",font=('Times',18),bg='yellow').grid(row=4,
column=0)
    self.entry_pass = Entry(fr1, show="*")
    self.entry_pass.grid(row=4, column=1)
    Button(fr1,text='Sign up',command=self.add).grid(row = 6,column=1,sticky=W)

    fr1.pack()
    self.data.mainloop()
```

In `add` function username and password is get back which was entered at signup page.

This username and password is stored in `database` file, `username` as `key` and `password` as `value` .

After storing close the database file and it will `destory` Sign up window.

```
def add(self):
    username = self.entry_user.get()
    password = self.entry_pass.get()
    dbo = dbm.open('data','w')
    dbo[f'{username}'] = f'{password}'
    dbo.close()
    self.data.destroy()
```

In `Home` function username and password is get back which was entered at `login page`

If username is in database but `not matchs` with corresponding key then there will be display of messagebox `*Incorrect password`

If username is `not` in database then there will be display a messagebox `Invalid Credentials` and `suggestion` message `'Try to sign up If you already sign up check username'` .

If username and password is `matchs` with key and corresponding value in `database` file then the `window` will `redirect` to `Snapchat Filter` window

In `Snapchat Filter` window two buttons are avalilable:

- `Apply on image`
- `Apply on cam`

By click on `Apply on image` button a new window will be opened with title `Filters on image` in function `Filter_on_Image` .
By click on `Apply on cam` button a new window will be opened with title `Filter by cam` in function `Filter_on_cam` .

```
def Home(self):

    username = self.entry_username.get()
    password = self.entry_password.get()

    dbo = dbm.open('data','r')
    if username in dbo:
        k = dbo[f'{username}'].decode()
        if k==password:
            la = Label(self.Authen_root,text='Correct
Password',fg='green',bg='yellow')
            la.pack()

            self.Authen_root.destroy()
            self.Home_root = Tk()
            self.Home_root.resizable(width=False, height=False)

            x = self.Home_root.winfo_screenwidth()
            y = self.Home_root.winfo_screenheight()
            ww = 500
            wh = 500
            self.Home_root.geometry(f"{ww}x{wh}+{(x-ww)//2}+{(y-wh)//2}")

            self.Home_root.title("Snapchat Filter")
            self.Home_root.configure(bg='yellow')
            c = 'yellow'

            self.splt(self.Home_root,2,c)

            l1 = Label(self.Home_root,text="SNAPCHAT FILTERS",font=('Times',20),
                    fg = 'black',bg='yellow')
            l1.pack()

            graphical_image = PhotoImage(file="logo.png")
            canvas = Canvas(self.Home_root, width=200, height=113,bg='yellow',
                            highlightbackground='yellow',highlightthickness=5)
            canvas.create_image(0, 0, anchor=NW, image=graphical_image)
            canvas.pack()


            self.splt(self.Home_root,2,c)
            b1 = Button(self.Home_root,text="Apply on Image",fg='black',
                        width=15,command=self.Filter_on_Image)
            b1.pack()

            self.splt(self.Home_root,2,c)
            b3 = Button(self.Home_root,text='Apply by cam',fg='black',width=15,
                        command=self.Filter_on_Cam)
            b3.pack()
            self.Home_root.mainloop()

        else:
            new = Tk()
            new.title('Message')
            for _ in range(2):
                label_blank = Label(new,text="")
```

```
                label_blank.pack()
            new.geometry(f"{300}x{150}+{(self.x-300)//2}+{(self.y-150)//2}")
            la = Label(new,text='*Incorrect Password',font=('Times',20),fg='red')
            la.pack()
            dbo.close()
            new.mainloop()
    else:
        new = Tk()
        new.title('Message')
        for _ in range(2):
            label_blank = Label(new,text="")
            label_blank.pack()
        new.geometry(f"{300}x{150}+{(self.x-300)//2}+{(self.y-150)//2}")
        la = Label(new,text='Invaild Credentials',font=('Times',20),fg='red')
        la.pack()
        la1 = Label(new,text='Try to sign up\n If you already sign up check
username')
        la1.pack()
        new.mainloop()
        dbo.close()
        return
```

In `Filter_on_Image` function new window will be opened with title `Filter on image` .

In `Filter on Image window` three buttons are avalilabe:

- `Face mesh filter on image`
- `Face mesh filter on set of image`
- `TUG LIFE on image`

By click on `Face mesh filter` button a new window will be opened to select `image` from user device and will be go to function `f1` .
By click on `Face mesh filter` button a new window will be opened to select `directory` from user device and will be go to function `f2` .
By click on `TUG LIFE on image` button a new window will be opened to select `image` from user device and will be go to function `f3` .

```
def Filter_on_Image(self):

    self.filter_image = Tk()
    self.filter_image.title('Filters on image')
    self.filter_image.configure(bg='yellow')
    self.filter_image.resizable(width=False, height=False)

    ww = 387
    wh = 400
    self.filter_image.geometry(f"{ww}x{wh}+0+0")

    self.splt(self.filter_image,2,'yellow')
    ba = Button(self.filter_image,text="Face mesh filter on image",fg='black',
                command=self.f1)
    ba.pack()

    self.splt(self.filter_image,2,'yellow')
    ba1 = Button(self.filter_image,text="Face mesh filter on set of
image",fg='black',
                 command=self.f2)
    ba1.pack()

    self.splt(self.filter_image,2,'yellow')
```

```
    ba2 = Button(self.filter_image,text="TUG LIFE on image",fg='black',
                 command=self.f3)
    ba2.pack()

    self.filter_image.mainloop()
```

In `blank` function, three empty line frame is created

```
def blank(self):

    blank = Frame(self.Authen_root)

    for _ in range(3):
        label_blank = Label(blank,text="",bg='yellow')
        label_blank.pack()

    blank.pack()
```

In `splt` function, `n` empty line Label, `root` respective window, `c` is for background color is created

```
def splt(self,root,n,c):

    for _ in range(n):
        t = Label(root,text="",bg=c)
        t.pack()
```

In function `f1` , object to class `mesh_image` is created which is in `Face_mesh_filter_image` module and object is calling function `apply`

```
def f1(self):
    k = Face_mesh_filter_image.mesh_image()
    k.apply()
```

In function `f2` , object to class `mesh_set_image` is created which is in `Face_mesh_filter_set_of_images` module and object is calling function `apply`

```
def f2(self):
    k = Face_mesh_filter_set_of_images.mesh_set_image()
    k.apply()
```

In function `f3` , object to class `tuglife_image` is created which is in `tug_life_filter_image` module and object is calling function `apply`

```
def f3(self):
    k = tug_life_filter_image.tuglife_image()
    k.apply()
```

In `Filter_on_Cam` function new window will be opened with title `Filter by cam` .

In `Filter on cam window` three buttons are avalilabe:

- `Face Mesh`
- `Animated eyes with fire smoke`
- `Face mesh Animated eyes with fire smoke`

- TUG LIFE on cam

By click on Face Mesh button a new window will be open which is camera of user device which will go to function f4 .
By click on Animated eyes with fire smoke button a new window will be open which is camera of user device which will go to function f5 .
By click on Face mesh Animated eyes with fire smoke button a new window will be open which is camera of user device which will go to function f6 .
By click on TUG LIFE on cam button a new window will be open which is camera of user device which will go to function f7 .

```
def Filter_on_Cam(self):

    self.filter_image = Tk()
    self.filter_image.configure(bg='yellow')
    self.filter_image.title('Filters by cam')

    ww = 387
    wh = 400
    x = self.filter_image.winfo_screenwidth()-ww
    self.filter_image.geometry(f"{ww}x{wh}+{x}+0")
    self.filter_image.resizable(width=False, height=False)

    self.splt(self.filter_image,2,'yellow')
    ba = Button(self.filter_image,text="Face Mesh",fg='black',
                command=self.f4)
    ba.pack()

    self.splt(self.filter_image,2,'yellow')
    ba1 = Button(self.filter_image,text="Animated eyes with fire
smoke",fg='black',
                command=self.f5)
    ba1.pack()

    self.splt(self.filter_image,2,'yellow')
    ba2 = Button(self.filter_image,text="Face mesh Animated eyes with fire
smoke",fg='black',
                command=self.f6)
    ba2.pack()

    self.splt(self.filter_image,2,'yellow')
    ba3 = Button(self.filter_image,text="TUG LIFE on cam",fg='black',
                 command=self.f7)
    ba3.pack()

    self.filter_image.mainloop()
```

In function f4 , object to class mesh_cam is created which is in Face_mesh_live module and object is calling function apply

```
def f4(self):
    k = Face_mesh_live.mesh_cam()
    k.apply()
```

In function f5 , object to class smoke_cam is created which is in Eyes_smoke_filter_cam module and object is calling function apply

```
def f5(self):
```

```
    k = Eyes_smoke_filter_cam.smoke_cam()
    k.apply()
```

In function `f6` , object to class `mesh_smoke_cam` is created which is in `Face_mesh_eyes_smoke` module and object is calling function `apply`

```
def f6(self):
    k = Face_mesh_eyes_smoke.mesh_smoke_cam()
    k.apply()
```

In function `f7` , object to class `tuglife_cam` is created which is in `tug_life_filter_cam` module and object is calling function `apply`

```
def f7(self):
    k = tug_life_filter_cam.tuglife_cam()
    k.apply()
```