INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL
DEPARTEMENT TECHNOLOGIES DE L'INFORMATIQUE

RAPPORT PYTHON

THEME DE PROJET

RECONNAISSANCE FACIAL

PERIODE

06 JANVIER 2022

REALISE PAR

AYA BEN KHEDHER

WISSAL DAOUED

PROFESSEUR

NAOUFEL FAKER

SESSION JANVIER 2022

Sommaire

Liste des figures

Introduction

La reconnaissance faciale est une technologie de plus en plus répandue, basée sur l'intelligence artificielle, permettant d'identifier une personne sur une photo ou une vidéo en comparant son visage avec ceux sauvegardés dans une base de données.
Aujourd'hui, La reconnaissance de visage est un domaine mature qui fait d'ailleurs l'objet de plusieurs librairies, Framework mais aussi et surtout de services cognitifs.

Les Outils

Création de l'interface graphique :

pour la création de l'interface graphique :nous avons utiliser Tkinter c'est une bibliothéque graphique libre d'origine pour le langage Python.

Système de reconnaissance faciale :

Un système de reconnaissance faciale est une application logicielle visant a reconnaitre une personne grâce a son visage de manière automatique. C'est un domaine de la vision par ordinateur consistant a reconnaitre automatiquement une personne a partir d'une image de son visage.

Création de la base de données et l'insertion des données :

pour la création de la base de données et l'insertion des données ,nous avons utilisé SQlite qui est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL SQLite.

Algorithme de reconnaissance facial :

Pour travailler avec le systéme de reconnaissance facial nous avons installer Open Cv c'est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.

Envoi des mails :

Pour envoyer des mails , nous avons utiliser le module smtplib qui définit un objet de session client SMTP qui peut être utilisé pour envoyer du courrier a n'importe quelle machine Internet avec un démon d'écoute SMTP ou ESMTP.

Les Captures

```
def webdet():
  cap = cv2.VideoCapture(0)
  sfr = SimpleFacerec()
  sfr.load_encoding_images("images/")

  while True:
    _,frame = cap.read()


    face_locations, face_names = sfr.detect_known_faces(frame)

    for face_loc, name in zip(face_locations, face_names):
        y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2], face_loc[3]

        cv2.putText(frame, name,(x1, y1 - 10), cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 200), 2)
        cv2.rectangle(frame, (x1,y1), (x2,y2), (0, 0, 200), 4)

    cv2.imshow("Frame", frame)

    print("press q to capture", frame)
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        cv2.imwrite("capture image opencv.jpg", frame)
        print("image captured")
        break
    if key == 27:
        break

  cap.release()
  cv2.destroyAllWindows()
```

Figure 1 : Capter visage

```
def send():
  dateTimeNow = datetime.datetime.now()

  message = MIMEMultipart()
  message['from'] = "ayabenkhedher84@gmail.com"
  message['to'] = "daoudwissal2000@gmail.com "
  Password = "Wissalpqt1"
  message['Subject'] = "Face recognition app"
  body = "Welcome to my Face recognition app  " + str(dateTimeNow)

  image_open = open('capture image opencv.jpg' ,'rb').read()

  message.attach(MIMEText(body, 'html'))
  message.attach(MIMEImage(image_open, 'jpg', name= 'aya.jpg'))



  server = smtplib.SMTP("smtp.gmail.com", 587)
  server.starttls()
  server.login(message['From'], Password)
  server.sendmail(message['From'], message['To'], message.as_string())
  server.quit()

  os.system('face recognition')
  cv2.destroyAllWindows()
```

Figure 2 : Envoyer un email

```
def data():
    try:
        sqliteConnection = sqlite3.connect('SQLite_data.db')
        cursor = sqliteConnection.cursor()
        print("Connected to SQLite")

        cursor.execute("SELECT * from Person")
        record = cursor.fetchall()
        profil = 0
        for row in record:
            First_name = row[0]
            Last_name = row[1]
            image = row[2]
            mail = row[3]
            print("First_name =", row[0], "Last_name =", row[1], "img = " , row[2], "mail =" , row[3])
            print("Storing Person image on disk \n")

        cursor.close()

    except sqlite3.Error as error:
        print("Failed to read blob data from sqlite table", error)
    finally:
        if sqliteConnection:
            sqliteConnection.close()
            print("sqlite connection is closed")
```

Figure 3 : Contenu de la base de données

```
root=Tk()
root.geometry('500x570')
frame = Frame(root, relief=RIDGE, borderwidth=2)
frame.pack(fill=BOTH,expand=1)
root.title('Reconnaissance Facial')
frame.config(background='light pink')
label = Label(frame, text="Reconnaissance Facial",bg='light pink',font=('Times 35 bold'))
label.pack(side=TOP)
filename = ImageTk.PhotoImage(Image.open("C:\\miniprojet\\demo4.jpg"))
background_label = Label(frame,image=filename)
background_label.pack(side=TOP)
```

Figure 4 : Utilisation Tkinter

```
but1=Button(frame,padx=5,pady=5,width=39,bg='white',fg='black',relief=GROOVE,command=web,text='Open Cam',font=('helvetica 15 bold'))
but1.place(x=5,y=104)

but2=Button(frame,padx=5,pady=5,width=39,bg='white',fg='black',relief=GROOVE,command=webrec,text='Open Cam & Detect',font=('helvetica 15 bold'))
but2.place(x=5,y=176)

but3=Button(frame,padx=5,pady=5,width=39,bg='white',fg='black',relief=GROOVE,command=webdet,text='Open Cam & Capture',font=('helvetica 15 bold'))
but3.place(x=5,y=250)

but4=Button(frame,padx=5,pady=5,width=39,bg='white',fg='black',relief=GROOVE,command=send,text='Send Email',font=('helvetica 15 bold'))
but4.place(x=5,y=322)

but5=Button(frame,padx=5,pady=5,width=39,bg='white',fg='black',relief=GROOVE,command=data,text='Read From Database',font=('helvetica 15 bold'))
but5.place(x=5,y=400)

but6=Button(frame,padx=5,pady=5,width=5,bg='white',fg='black',relief=GROOVE,text='EXIT',command=exitt,font=('helvetica 15 bold'))
but6.place(x=210,y=478)
```

Figure 5 : Création Bouton avec Tkinter

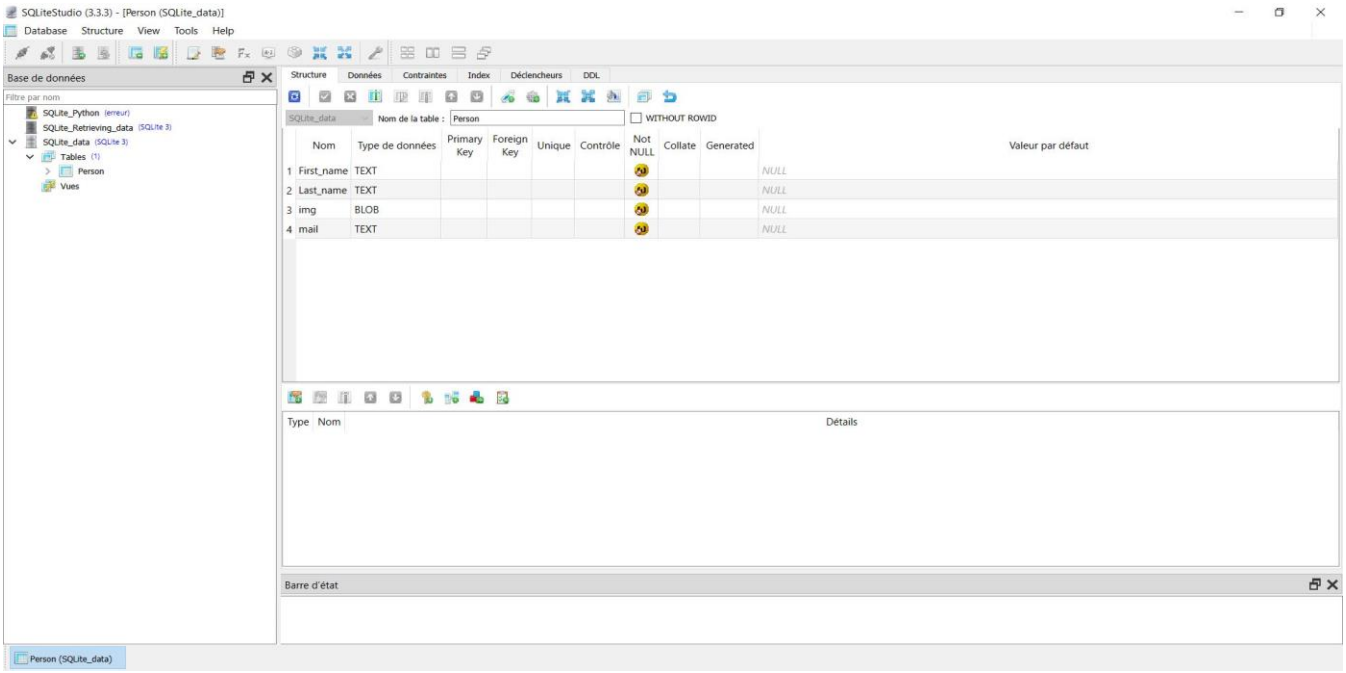

Figure 6 : Base de données

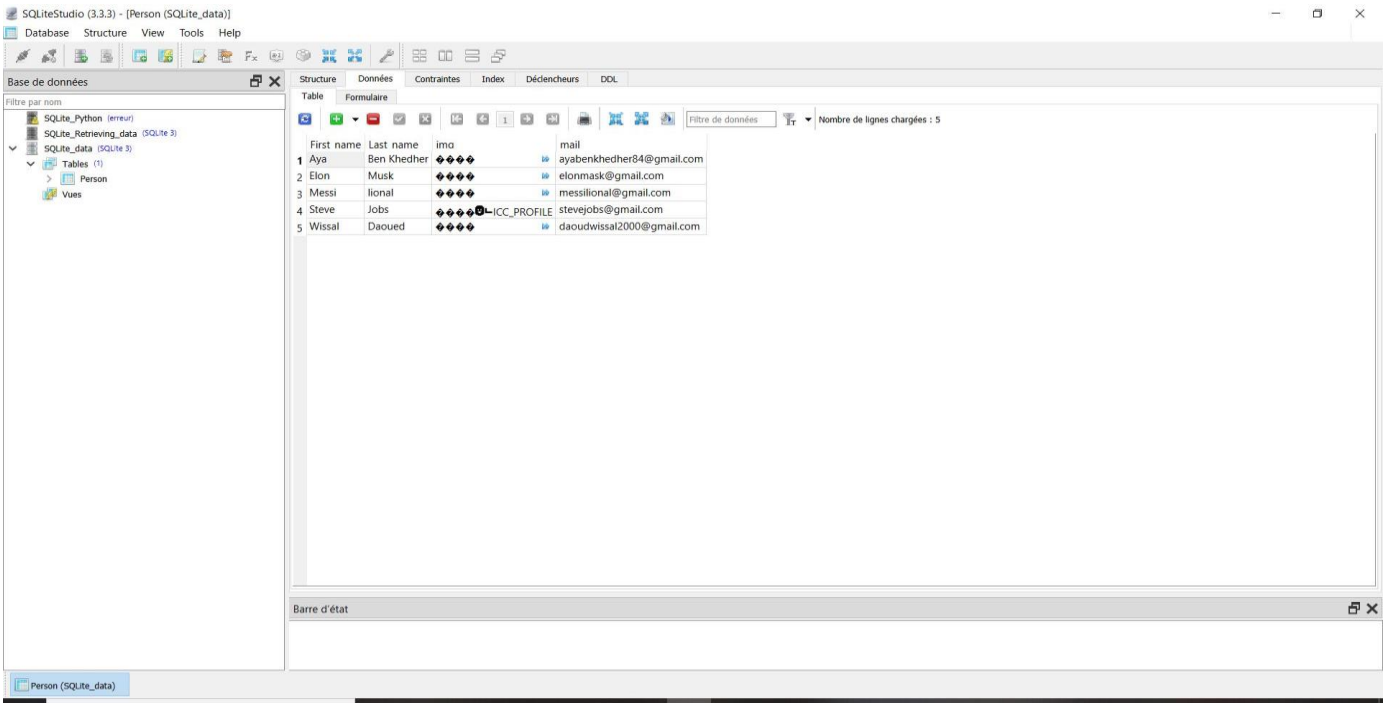

Figure 7 : L'insertion dans la base de données

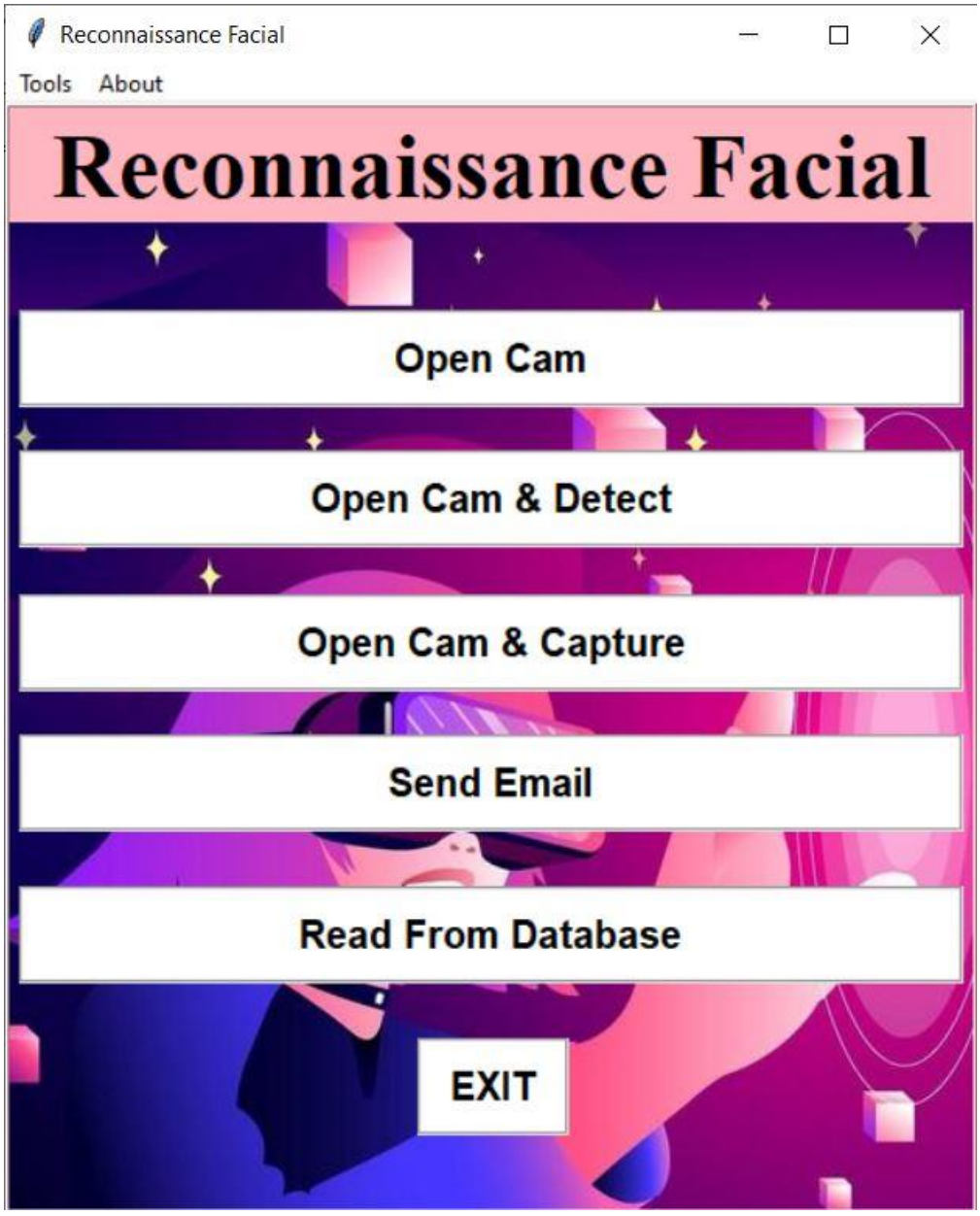

Figure 8 : Résultat final

Conclusion

Ce projet nous a permis d'acquérir une bonne base sur le langage python.

Nous avons utilisé des nouvelles bibliothèques de reconnaissance facial et analyse de données pour les extraire.

Aussi nous nous sommes concentrés sur les aspects essentiels à notre projet : nous avons ainsi appris à envoyer des mails en python , insérer les données avec SQLite.

Enfin, Nous avons beaucoup appris au cours de toutes les étapes de la réalisation de notre projet à travers les problèmes rencontrés et résolus, l'apprentissage du langage Python, l'utilisation des nouvelles bibliothèque de la reconnaissance facial Au final, le chemin parcouru aura été aussi important que le résultat final.

Références

Reconnaissance Facial :

https://www.youtube.com/watch?v=5yPeKQzCPdI

Renvoi des mails :

https://www.youtube.com/watch?v=wjrFK2XKgZs

Insertion des données :

https://www.youtube.com/watch?v=6gWS2CdtZrs