



Face Recognition

Adnane Mehdaoui

Rachid Baiou

Plan:

Partie I:

1-Introduction

2-Objectif

3-Démarche

Partie II:

1- La structure de l'application

2-les résultats

3-Discussion

Partie I:

1-Introduction :

La reconnaissance faciale, également appelée reconnaissance biométrique faciale, est une technologie qui permet d'identifier une personne à partir de son visage. Cette technologie utilise des algorithmes d'apprentissage automatique pour analyser les caractéristiques du visage d'une personne, telles que la distance entre les yeux, la forme du nez et la structure de la mâchoire, afin de créer une empreinte faciale unique pour cette personne.

L'importance de la reconnaissance faciale réside dans ses applications potentielles dans divers domaines, notamment la sécurité, la surveillance, la reconnaissance des criminels, la gestion des identités, et même la reconnaissance émotionnelle. Dans le domaine de la sécurité, la reconnaissance faciale pourrait être utilisée pour identifier les suspects de manière plus efficace et rapide, pour contrôler l'accès aux lieux sensibles, ou pour détecter les comportements suspects dans les lieux publics.

En outre, la reconnaissance faciale peut également être utilisée pour améliorer l'expérience utilisateur dans diverses applications, telles que le déverrouillage des téléphones portables ou l'accès aux comptes bancaires. Elle peut également être utilisée pour améliorer l'identification des personnes dans les photos, les vidéos et les réseaux sociaux, offrant ainsi des avantages dans la gestion des identités et la recherche de personnes disparues.

2-Objectif:

L'objectif de votre application de reconnaissance faciale est d'utiliser des algorithmes d'apprentissage automatique pour identifier les caractéristiques uniques du visage d'une personne et pouvoir la reconnaître ultérieurement. Pour atteindre cet objectif, vous devez collecter des données d'entraînement, entraîner l'algorithme de reconnaissance faciale, intégrer l'algorithme dans votre application, tester et améliorer l'application en tenant compte des préoccupations de sécurité et de confidentialité.

3-Démarche:

1. Utiliser une bibliothèque de reconnaissance faciale : Vous pouvez utiliser une bibliothèque de reconnaissance faciale telle que Face recognition , Dlib ou pillow pour créer votre application.

2. Collecter les données d'entraînement : Vous devez collecter et préparer une grande quantité de données d'entraînement pour votre algorithme de reconnaissance faciale. Ces données peuvent inclure des images de visages provenant de différentes sources et avec différentes expressions faciales.
3. Entraîner l'algorithme : Vous devez entraîner votre algorithme de reconnaissance faciale avec les données d'entraînement que vous avez collectées. Cela implique de choisir les bons paramètres pour votre algorithme et d'optimiser les performances de l'algorithme pour de meilleurs résultats.
4. Intégrer l'algorithme dans votre application : Vous devez intégrer l'algorithme de reconnaissance faciale que vous avez créé dans votre application, en utilisant les bibliothèques et les API appropriées.
5. Tester et améliorer l'application : Vous devez tester l'application pour vous assurer qu'elle fonctionne correctement et fournir des résultats précis..

Partie I I:

1-Structure de l'application:

Dans notre application, on a utilisé comme architecture mvc(modèle,view controller).

-view:c'est la partie du code qui est responsable de l'affichage de

l'interface graphique, dans laquelle l'utilisateur peut mettre l'image du personne dans laquelle il veut connaître son visage.

-modèle: c'est la partie la plus importante de l'application qui est responsable de faire les traitement nécessaire pour la détection du visage, affichage du nom de la personne si est connue.

-controller: c'est le responsable pour lier le modèle et la vue.

Voila le code correspondant a chaque partie:

Vue:

```
<!doctype html>
<html>
  <head>
    <title>Upload an Image</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.0.2/css/bootstrap.min.css">
    <style>
      .image-container {
        position: absolute;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
        border: 2px solid black;
        padding: 10px;
      }
      .image-container img {
        max-width: 100%;
      }
    </style>
  </head>
  <body class="container">
    <h1 class="text-center">Upload an Image</h1>
    <div class="row">
      <div class="col-md-6 mx-auto">
        <form method="POST" action="/" enctype="multipart/form-data">
          <div class="form-group mb-3">
            <input type="file" name="file" class="form-control" accept="image/*">
          </div>
          <div class="d-flex justify-content-center">
            <button type="submit" class="btn btn-primary">Upload</button>
          </div>
        </form>
      </div>
    </div>
  </body>
</html>
```

```

{% if filename %}
<h2 class="text-center">Uploaded Image:</h2>
<div class="image-container">
  <!--<div class="col-md-7 mx-auto"></div>-->
  
</div>

{% endif %}
{% if name %}
<h2 class="text-center">Name :{{name}} </h2>

{% endif %}

<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.0.2/js/bootstrap.min.js"></script>
</body>
</html>

```

Controller:

```

from flask import Flask, render_template, request
from face_detection import face_detection
from PIL import Image

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        image = request.files['file']
        image.save('static/' + image.filename)
        results,name = face_detection('static/' + image.filename)
        nouvelle_taille = (500, 400)
        nouvelle_image = results.resize(nouvelle_taille, resample=Image.LANCZOS)
        nouvelle_image.save('static/r'+image.filename)

        return render_template('home.html', filename='r'+image.filename,name=name)
    return render_template('home.html')

if __name__ == '__main__':
    app.run(debug=True)

```

Modèle:

```

import dlib
import face_recognition
from PIL import Image, ImageDraw, ImageFont
import numpy as np
import math

def face_detection(lien):
    def comparedistance(t, t1):
        d=0
        for i in range(len(t)):
            d = d + abs(t[i]-t1[i])
        return d
    def Knn(face_encoding, known_face_name, known_face_encoding):
        face_encoding_array = face_encoding.flatten()
        dictdistance = {}
        for name, encoding in zip(known_face_name, known_face_encoding):
            d = comparedistance(encoding, face_encoding_array)
            if name in dictdistance:
                dictdistance[name].append(d)
            else:
                dictdistance[name] = [d]
        print(dictdistance)

        sorted_dict = {k: sorted(v) for k, v in dictdistance.items()}
        print(sorted_dict)

        closest_face_distance = None
        closest_face_name = None
        for name, distances in sorted_dict.items():
            if closest_face_distance is None or distances[0] < closest_face_distance:
                closest_face_distance = distances[0]
                closest_face_name = name

```



```

    if closest_face_distance - 4.50 <= 0.10:
        return closest_face_name
    else:
        return "Unknown"

# maria detection
image_maria = face_recognition.load_image_file("photo/maria.jpg")
maria_encoding_face = face_recognition.face_encodings(image_maria)[0]

image_maria1 = face_recognition.load_image_file("photo/maria1.jpg")
maria_encoding_face1 = face_recognition.face_encodings(image_maria1)[0]
#shakira
image_shakira = face_recognition.load_image_file("photo/shakira.jpg")
shakira_encoding_face = face_recognition.face_encodings(image_shakira)[0]

image_shakira1 = face_recognition.load_image_file("photo/shakira2.jpg")
shakira_encoding_face1 = face_recognition.face_encodings(image_shakira1)[0]
#neymar

image_neymar = face_recognition.load_image_file("photo/neymar.jpg")
image_neymar_face = face_recognition.face_encodings(image_neymar)[0]

image_neymar1 = face_recognition.load_image_file("photo/neymar1.jpg")
image_neymar1_encoding_face1 = face_recognition.face_encodings(image_neymar1)[0]
#ronaldo

image_ronaldo = face_recognition.load_image_file("photo/ronaldo3.jpg")
image_ronaldo_face = face_recognition.face_encodings(image_ronaldo)[0]

image_ronaldo1 = face_recognition.load_image_file("photo/ronaldo1.jpg")
image_ronaldo1_encoding_face1 = face_recognition.face_encodings(image_ronaldo1)[0]
# messi

image_messi = face_recognition.load_image_file("photo/messi.jpg")
image_messi_face = face_recognition.face_encodings(image_messi)[0]

```

```

image_messi1 = face_recognition.load_image_file("photo/messi1.jpg")

image_messi1_encoding_face1 = face_recognition.face_encodings(image_messi1)[0]

# moutouali

image_mohssine1 = face_recognition.load_image_file("photo/mohssine1.jpg")
image_mohssine1_face = face_recognition.face_encodings(image_mohssine1)[0]

image_mohssine3 = face_recognition.load_image_file("photo/mohssine3.jpg")

image_mohssine3_encoding_face1 = face_recognition.face_encodings(image_mohssine3)[0]

#boufal
image_soufiane = face_recognition.load_image_file("photo/soufiane.jpg")
image_soufiane_face = face_recognition.face_encodings(image_soufiane)[0]

image_soufiane1 = face_recognition.load_image_file("photo/soufiane1.jpg")
image_soufiane1_encoding_face1 = face_recognition.face_encodings(image_soufiane1)[0]
#youssef
image_youssef = face_recognition.load_image_file("photo/youssef.jpg")
image_youssef_face = face_recognition.face_encodings(image_youssef)[0]

image_youssef2 = face_recognition.load_image_file("photo/youssef2.jpg")
image_youssef2_encoding_face1 = face_recognition.face_encodings(image_youssef2)[0]
#saadLamjarred
image_saadLamjarred = face_recognition.load_image_file("photo/saadLamjarred.jpg")
image_saadLamjarred_face = face_recognition.face_encodings(image_saadLamjarred)[0]

image_saadLamjarred2 = face_recognition.load_image_file("photo/saadLamjarred1.jpg")
image_saadLamjarred2_encoding_face1 = face_recognition.face_encodings(image_saadLamjarred2)[0]
#Mbappe
image_Mbappe = face_recognition.load_image_file("photo/Mbappe.jpg")
image_Mbappe_face = face_recognition.face_encodings(image_Mbappe)[0]

image_Mbappe2 = face_recognition.load_image_file("photo/Mbappe1.jpg")
image_Mbappe2_encoding_face1 = face_recognition.face_encodings(image_Mbappe2)[0]

```

```

#badrHari
image_badrHari = face_recognition.load_image_file("photo/badrHari.jpg")
image_badrHari_face = face_recognition.face_encodings(image_badrHari)[0]

image_badrHari2 = face_recognition.load_image_file("photo/badrHari1.jpg")
image_badrHari2_encoding_face1 = face_recognition.face_encodings(image_badrHari2)[0]
#hakimZiyech
image_hakimZiyech = face_recognition.load_image_file("photo/ziyech1.jpg")
image_hakimZiyech_face = face_recognition.face_encodings(image_hakimZiyech)[0]

image_hakimZiyech2 = face_recognition.load_image_file("photo/ziyech2.jpg")
image_hakimZiyech2_encoding_face1 = face_recognition.face_encodings(image_hakimZiyech2)[0]

known_face_encoding = [maria_encoding_face ,maria_encoding_face1 ,shakira_encoding_face,
                        shakira_encoding_face1,image_neymar_face,
                        image_neymar1_encoding_face1,image_ronaldo_face,image_ronaldo1_encoding_face1,
                        image_messi_face ,image_messi1_encoding_face1, image_mohssine1_face,
                        image_mohssine3_encoding_face1,image_soufiane_face,image_soufiane1_encoding_face1,image_youssef_face
                        ,image_youssef2_encoding_face1,image_saadLamjarred_face, image_saadLamjarred2_encoding_face1,
                        image_Mbappe_face,image_Mbappe2_encoding_face1,image_badrHari_face,image_badrHari2_encoding_face1,
                        image_hakimZiyech_face,image_hakimZiyech2_encoding_face1
                        ]
known_face_name = ["maria", "maria","shakira","shakira" ,"neymar","neymar","ronaldo","ronaldo","messi", "messi",
                  "mohssine", "mohssine","soufiane","soufiane","youssef","youssef","saadLamjarred","saadLamjarred",
                  "Mbappe","Mbappe","badrHari","badrHari","hakimZiyech","hakimZiyech"
                  ]
unknown_image = face_recognition.load_image_file(lien)
face_locations = face_recognition.face_locations(unknown_image)
face_encodings = face_recognition.face_encodings(unknown_image, face_locations)

known_face_encoding_array = [i.flatten() for i in known_face_encoding]
face_names = [Knn(face_encoding, known_face_name, known_face_encoding_array) for face_encoding in face_encodings]

```

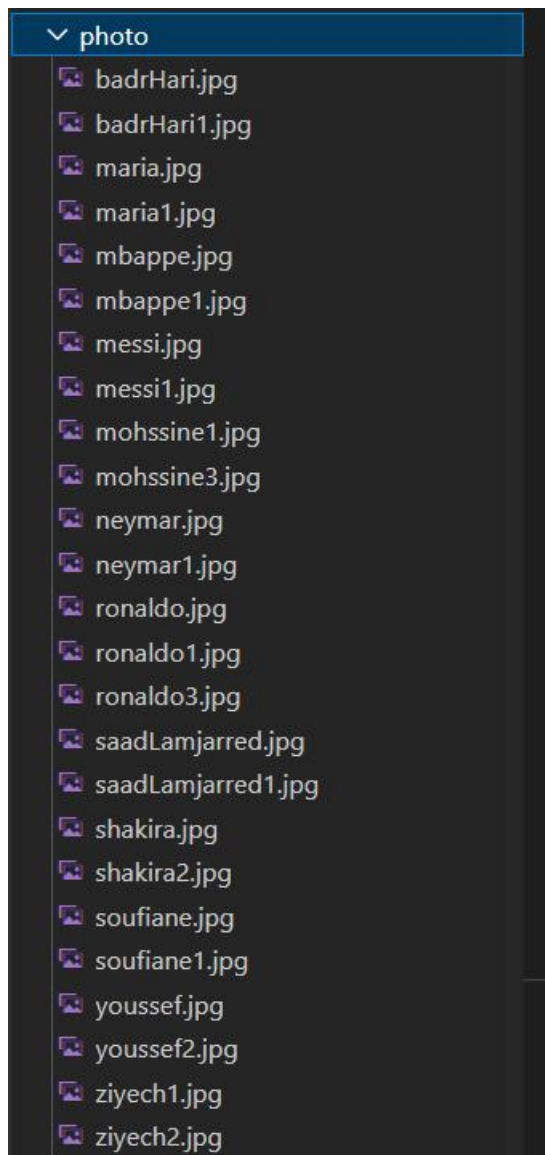
```

pil_image = Image.fromarray(unknown_image)
draw = ImageDraw.Draw(pil_image)
for (top, right, bottom, left), name in zip(face_locations, face_names):
    bold_font = ImageFont.truetype("C:\Windows\Fonts\Arial.ttf", size=30)
    draw.rectangle(((left, top), (right, bottom)), outline=(0, 255, 0))
    text_width, text_height = draw.textsize(name)
    draw.text((left + 6, bottom - text_height - 5), name, fill=(255, 255, 255), font=bold_font)
del draw

return pil_image,name

```

En plus de cela, nous avons utilisé un dossier contenant des photos pour l'entraînement, et pour les tests, l'utilisateur peut importer une image à partir de n'importe quel dossier .



2-Les résultat:

Voila des test qui ont fait par cette application,on a donné des image des personne , après notre application doit trouvé le nom de la personne choisie .

Upload an Image

Choisir un fichier

Aucun fichier n'a été sélectionné

Upload

Uploaded Image:

Name :badrHari



Upload an Image

Choisir un fichier

Aucun fichier n'a été sélectionné

Upload

Uploaded Image:
Name :Mbappe



Upload an Image

Choisir un fichier

Aucun fichier n'a été sélectionné

Upload

Uploaded Image:
Name :Unknown



3-Discussion:

On a utilisé comme modèle dans notre application le KNN. Il nous a permis de bien déterminer le visage le plus ressemblant au visage dont on voulait connaître le nom. Le modèle a donné une précision de 83,33%, ce qui peut être considéré comme une bonne précision, mais cela ne garantit pas une distinction précise entre les visages qui se ressemblent beaucoup, comme deux jumeaux. Bien que nous ayons traité ce point dans notre code, il est parfois difficile même pour l'être humain de distinguer entre deux personnes qui se ressemblent beaucoup. Malgré cela, le modèle KNN reste utile pour

notre application car il fournit une bonne approximation pour identifier les visages.

[Lien de l'application:](#)

https://github.com/Adnane-me/machine_1.git

