



Find Yourself (AI)

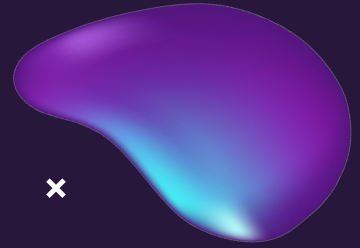
Un proyecto que te ayude a encontrar todas las fotos en las que salga tu cara

Minerva Quintero Flores



x

Tabla de contenidos



x

- 01 Descripción**
- 02 Código**
- 03 Ejecución**
- 04 Problemas encontrados**
- 05 PROJECT STAGES**



01 - Descripción

Esta aplicación tiene como objetivo encontrar personas determinadas entre una colección de fotos a partir una foto de partida.

02 - Código

```
def search_face(img1_p, directory):
    metrics = ["cosine", "euclidean", "euclidean_l2"]
    print(directory)
    #face recognition
    df = DeepFace.find(img_path = img1_p,
                       db_path = directory,
                       distance_metric = metrics[2],
                       enforce_detection=False
    )

    img_paths = df[0]['identity']
    img_url = []

    for url in img_paths:
        # print(url)
        img_url.append(url)

    return img_url

def search_two_faces(img1_p, img2_p, directory):
    url_im1 = search_face(img1_p, directory)
    url_im2 = search_face(img2_p, directory)
    resultado = elementos_repetidos(url_im1, url_im2)
    return resultado
```

```
#Función para extraer imágenes en las que salen dos personas a la vez
def elementos_repetidos(arr1, arr2):
    set_arr1 = set(arr1)
    set_arr2 = set(arr2)
    elementos_comunes = set_arr1.intersection(set_arr2)
    return list(elementos_comunes)
```

02 - Código

```
def seleccionar_imagen(self):
    archivo_imagen = filedialog.askopenfilename(
        title="Seleccionar archivo de imagen",
        filetypes=[("Archivos de imagen", "*.jpg;*.jpeg;*.png")]
    )
    if archivo_imagen:
        self.photo_path=archivo_imagen
        self.etiqueta_imagen.config(text=f"Imagen seleccionada: {archivo_imagen}")
        # Puedes llamar a tu función de extraer imagen aquí
```

```
for path in img_paths:
    label = Label(self.frame_labels, text=f"Imagen encontrada: {path}")
    label.pack(pady=5)
    label.bind("<Button-1", lambda event, path=path: self.abrir_imagen_label(event, path))
    label.bind("<Enter", lambda event, label=label: self.cambiar_color_label(label, "blue", True))
    label.bind("<Leave", lambda event, label=label: self.cambiar_color_label(label, "black", False))
```

02 - Código

```
# Botones
self.btn_seleccionar_directorio = Button(ventana, text="Seleccionar Directorio", command=self.seleccionar_directorio)
self.btn_seleccionar_directorio.pack(pady=10)
self.etiqueta_directorio = Label(ventana, text="Directorio seleccionado:")
self.etiqueta_directorio.pack(pady=5)

self.btn_seleccionar_imagen = Button(ventana, text="Seleccionar Imagen", command=self.seleccionar_imagen)
self.btn_seleccionar_imagen.pack(pady=10)
self.etiqueta_imagen = Label(ventana, text="Imagen seleccionada:")
self.etiqueta_imagen.pack(pady=5)

self.btn_buscar_persona = Button(ventana, text="Buscar", command=self.buscar_persona)
self.btn_buscar_persona.pack(pady=10)
self.label = Label(self.ventana, text="Imagen encontrada: ")
```

```
#Función que permite abrir la imagen cuando clican sobre ella
def abrir_imagen_label(self, event, img_path):
    if os.path.exists(img_path):
        img = Image.open(img_path)
        img.show()

def cambiar_color_label(self, label, color, subrayar):
    label.config(fg=color, underline=subrayar)
```

03 - Ejecución

×

04 - Problemas encontrados

Búsqueda Brad Pitt

×

Buscar

Imagen encontrada: ./images//brad.jpg

Imagen encontrada: ./images//brad_ang_7.jpg

Imagen encontrada: ./images//brad_ang_.jpg

Imagen encontrada: ./images//brad_ang_2.jpg

Imagen encontrada: ./images//brad_ang_6.jpg

Imagen encontrada: ./images//brad_ang_5.jpg

Búsqueda Angelina Jolie

Buscar

Imagen encontrada: ./images//ang.jpg

Imagen encontrada: ./images//face-1252996_1280.jpg

Imagen encontrada: ./images//brad_ang_4.jpg

Imagen encontrada: ./images//face-1262801_1280.jpg

Imagen encontrada: ./images//child-2162408_1280.jpg

Imagen encontrada: ./images//child-2162410_1280.jpg

Imagen encontrada: ./images//man-1252993_1280.jpg

Imagen encontrada: ./images//face-1253009_1280.jpg

¡Gracias!

