

FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL

INFORME DE AVANCES

TEMA:

Programa de Reconocimiento Facial

GRUPO DE TRABAJO:

- *Edgar David Martínez*
- *Jandri Giovanni Villavicencio*

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el reconocimiento facial se ha convertido en una herramienta esencial en diversos ámbitos, desde la seguridad hasta la gestión de identidades en aplicaciones cotidianas. Este proyecto se propone desarrollar una arquitectura de reconocimiento facial enfocada en la identificación precisa de rostros, excluyendo la evaluación de expresiones faciales. A medida que estas tecnologías avanzan, surgen desafíos específicos relacionados con la optimización de la precisión y eficiencia en la identificación de rostros en un conjunto de datos preentrenados.

El reto central consiste en diseñar una arquitectura capaz de analizar con precisión las características distintivas de un rostro, considerando factores como variaciones en la iluminación, cambios en la orientación y posibles obstáculos visuales. La efectividad de la arquitectura dependerá de su habilidad para discriminar y reconocer rostros en diversas condiciones, garantizando una alta tasa de precisión.

El uso de un conjunto de datos preentrenados plantea interrogantes sobre la adaptabilidad de la arquitectura a diversas características faciales y la generalización del reconocimiento a nuevas instancias, sin incluir la evaluación de expresiones emocionales. Abordar estos aspectos es esencial para asegurar la confiabilidad y aplicabilidad de la aplicación en entornos donde la identificación de rostros es la principal tarea.

OBJETIVOS

General: Desarrollar una arquitectura de reconocimiento facial que optimice la precisión y eficiencia en la identificación de rostros en un conjunto de datos preentrenados, enfocándose en la identificación precisa de rostros, con el propósito de responder eficazmente a la creciente demanda de tecnologías biométricas precisas y eficientes, prescindiendo de la evaluación de expresiones faciales.

Específicos:

- Diseñar una arquitectura que analice con precisión las características de un rostro, incorporando algoritmos y técnicas de procesamiento de imágenes para abordar variaciones en la iluminación, cambios en la orientación y obstáculos visuales, para garantizar la identificación precisa de rostros en diversas condiciones.

- Optimizar la capacidad de discriminación y reconocimiento mediante la implementación de técnicas de aprendizaje profundo que permitan adaptarse a diversas características faciales y escenarios, asegurando una alta tasa de precisión en la identificación de rostros en entornos dinámicos.
- Evaluar y abordar la adaptabilidad de la arquitectura a características faciales, realizando pruebas con datos que representen variabilidad facial y evaluando el rendimiento en nuevas instancias, para garantizar la generalización del reconocimiento facial a nuevas situaciones sin expresiones emocionales.
- Asegurar la confiabilidad y aplicabilidad de la aplicación mediante medidas de seguridad y privacidad, y la realización de pruebas éticas para mitigar sesgos y discriminaciones, para garantizar su utilidad en entornos donde la identificación de rostros es la tarea principal.

DESCRIPCIÓN

Este proyecto se enfoca en el desarrollo de una arquitectura de reconocimiento facial centrada en la identificación precisa de rostros, excluyendo la evaluación de expresiones faciales. La iniciativa surge en un contexto donde el reconocimiento facial desempeña un papel crucial en la seguridad y la gestión de identidades en diversas aplicaciones cotidianas. La arquitectura propuesta busca superar desafíos relacionados con la variabilidad en la iluminación, cambios de orientación y obstáculos visuales, priorizando la simplicidad y eficacia en la identificación de rostros.

La importancia de este proyecto radica en la creación de una herramienta biométrica altamente confiable y especializada. Al evitar la evaluación de expresiones faciales, se busca optimizar la arquitectura, aumentando su rendimiento y fomentando su aplicación en entornos donde la identificación precisa de rostros es primordial. Asimismo, este enfoque contribuye a la mejora de la seguridad y la eficiencia en diversos escenarios cotidianos, abordando la demanda creciente de tecnologías de reconocimiento facial más especializadas y adaptables a condiciones variables.

MARCO TEÓRICO

Conceptos

- **Aprendizaje Profundo (Deep Learning):** Técnica que permite que la arquitectura de reconocimiento facial mejore su capacidad de identificación mediante la exposición a conjuntos de datos extensos.
- **Red Neuronal Convolucional (CNN):** Tipo de red neuronal diseñada para procesar y analizar datos de imágenes, fundamental en el reconocimiento facial para la extracción de características.
- **Datos Biométrico:** Se refiere a las características físicas o comportamentales únicas utilizadas para la identificación de individuos, siendo el reconocimiento facial un ejemplo de biometría.
- **Preentrenamiento de Modelos:** Proceso en el cual la arquitectura se entrena inicialmente en un conjunto de datos general antes de ser afinada para la tarea específica de reconocimiento facial.
- **Generalización:** Capacidad de la arquitectura para aplicar el reconocimiento facial de manera efectiva a nuevas instancias y escenarios no vistos durante el entrenamiento.

- **Varianza en la Iluminación:** Desafío común en el reconocimiento facial relacionado con las variaciones en la cantidad y dirección de la luz que incide sobre los rostros.
- **Segmentación Facial:** Proceso de dividir una imagen para aislar y analizar específicamente la región facial, facilitando la identificación precisa.
- **Algoritmo de Emparejamiento:** Utilizado para comparar y verificar la similitud entre características faciales extraídas, determinando la identidad de un individuo.
- **Sistemas de Seguridad:** Área de aplicación del reconocimiento facial, donde se utiliza para el control de accesos y la autenticación biométrica, contribuyendo a la mejora de la seguridad en diferentes entornos.

Estado del arte

- **DeepFace de Facebook (2014):** El proyecto DeepFace de Facebook fue un hito importante, utilizando una red neuronal profunda para mapear y reconocer rostros en imágenes. Logró una tasa de precisión cercana al reconocimiento humano, marcando un avance significativo en la eficacia del reconocimiento facial.
- **Proyecto FaceNet de Google (2015):** FaceNet, desarrollado por Google, introdujo el concepto de embeddings faciales, representando las características de un rostro en un espacio tridimensional. Esto permitió una comparación más eficiente y precisa de rostros, incluso en grandes conjuntos de datos.
- **Clearview AI (2016):** Clearview AI desarrolló una polémica tecnología de reconocimiento facial que recopila y compara rostros utilizando una amplia base de datos de imágenes de Internet. Aunque controversial, su impacto en la discusión sobre ética y privacidad en el reconocimiento facial ha sido significativo.
- **Proyecto NtechLab's FindFace:** NtechLab's FindFace es una plataforma de reconocimiento facial que se ha utilizado en diversas aplicaciones, desde la seguridad pública hasta la identificación de individuos en entornos corporativos. Su enfoque en la precisión y la adaptabilidad lo ha hecho relevante en el panorama actual.

ALGORITMO SELECCIONADO

Algoritmo OpenCV DNN:

El algoritmo de reconocimiento facial del OpenCV DNN (Deep Neural Network) utiliza redes neuronales profundas para el reconocimiento facial. OpenCV DNN permite cargar modelos preentrenados, como el modelo de reconocimiento facial, para realizar tareas avanzadas de visión por computadora, incluido el reconocimiento de rostros.

Ventajas:

- Utiliza modelos de aprendizaje profundo, lo que permite un reconocimiento facial más preciso y robusto.
- Puede aprovechar la potencia de las GPU para acelerar el procesamiento.
- Es flexible y puede ser utilizado para otros fines de visión por computadora además del reconocimiento facial.

Algoritmo HaarCascade:

HaarCascade es un algoritmo de detección de objetos que utiliza patrones de características llamados Haar-like features. Se suele utilizar para el reconocimiento facial y otros objetos en imágenes.

Ventajas:

- Es eficiente y rápido, lo que lo hace adecuado para aplicaciones en tiempo real.
- Puede entrenarse para detectar diferentes tipos de objetos, no solo rostros.
- Funciona bien en condiciones de iluminación variable y diferentes orientaciones de objetos.

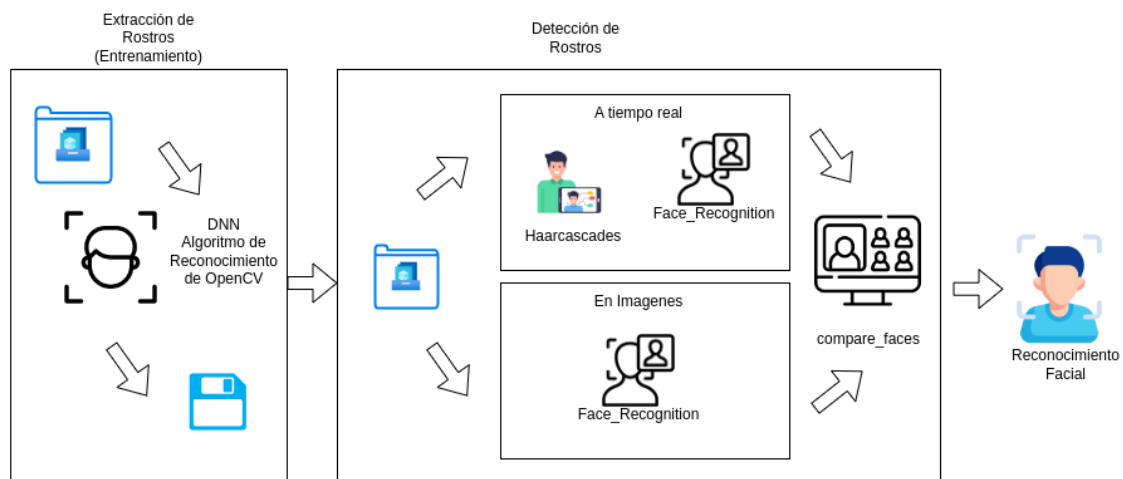
Librería Face_Recognition:

La librería Face_Recognition es una biblioteca de Python que simplifica la implementación del reconocimiento facial. Utiliza el modelo de aprendizaje profundo de face_recognition para detectar y reconocer caras en imágenes.

Ventajas:

- Fácil de usar y proporciona una interfaz sencilla para el reconocimiento facial.
- Utiliza modelos preentrenados, lo que facilita su implementación sin la necesidad de entrenamiento adicional.
- Es compatible con varias bibliotecas de procesamiento de imágenes en Python, como Pillow y OpenCV.

ARQUITECTURA



DESARROLLAR EL PROTOTIPO

Codificación del sistema de reconocimiento facial

```

~/Documents/UTPL/Septimo Ciclo/Inteligencia Artificial/DeteccionFacialV2/ExtraerRostros.py - Subli...  x
~/Documents/UTPL/Septimo Ciclo/Inteligencia Artificial/DeteccionFacialV2/DetecciónRostros.py - Su...  x
File Edit Selection Find View Goto Tools Project Preferences Help
File Edit Selection Find View Goto Tools Project Preferences Help

from opencv2 import OpenAI
import cv2
import os

imagesPath = "/home/edmartinez/Documents/UTPL/Septimo Ciclo/Inteligencia Artificial/De..."
if not os.path.exists("RostrosDetectados"):
    os.makedirs("RostrosDetectados")
    print("Nueva carpeta: RostrosDetectados")

# ----- READ DNN MODEL -----
prototxt = "model/deploy.prototxt" # Arquitectura
model = "model/res10_300x300_ssd_iter_140000.caffemodel" # Pesos
net = cv2.dnn.readNetFromCaffe(prototxt, model) # Cargamos el modelo de la red

count = 0
for imageName in os.listdir(imagesPath):
    print(imageName)
    image = cv2.imread(imagesPath + "/" + imageName)
    height, width, _ = image.shape
    image_resized = cv2.resize(image, (300, 300))

    # Create a blob
    blob = cv2.dnn.blobFromImage(image_resized, 1.0, (300, 300), (104, 117, 123))
    print(blob.shape) # (1, 3, 300, 300)
    blob_to_show = cv2.merge([blob[0][0], blob[0][1], blob[0][2]])
    cv2.imshow("blob", blob_to_show)

    net.setInput(blob)
    detections = net.forward()
    print(detections.shape) # (1, 1000, 5)

    for idx, detection in enumerate(detections[0][0]):
        if detection[2] > 0.5:
            box = detection[3:7] * [width, height, width, height]
            x_start, y_start, x_end, y_end = int(box[0]), int(box[1]), int(box[2]), int(box[3])
            face = image[y_start:y_end, x_start:x_end]
            face = cv2.resize(face, (150, 150))

            # Save the detected face as an image
            face_filename = os.path.join("RostrosDetectados/", f"face_{idx}.jpg")
            cv2.imwrite(face_filename, face)

            cv2.rectangle(image, (x_start, y_start), (x_end, y_end), (0, 255, 0), 2)
            cv2.putText(image, "Conf: {:.2f}".format(detection[2] * 100), (x_start,
            cv2.destroyAllWindows()

import cv2
import os
import face_recognition

# Codificar los rostros extraidos
imageFacesPath = "/home/edmartinez/Documents/UTPL/Septimo Ciclo/Inteligencia Artificial..."
facesEncodings = []
facesNames = []

for file_name in os.listdir(imageFacesPath):
    image = cv2.imread(imageFacesPath + "/" + file_name)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    f_coding = face_recognition.face_encodings(image, known_face_locations=[(0, 150, 1
    facesEncodings.append(f_coding)
    facesNames.append(file_name.split('.')[0])

#print(facesEncodings)
#print(facesNames)

cap = cv2.VideoCapture(0, cv2.CAP_ANY) # Activamos la camara para poder ver el
# Detector facial
faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_de

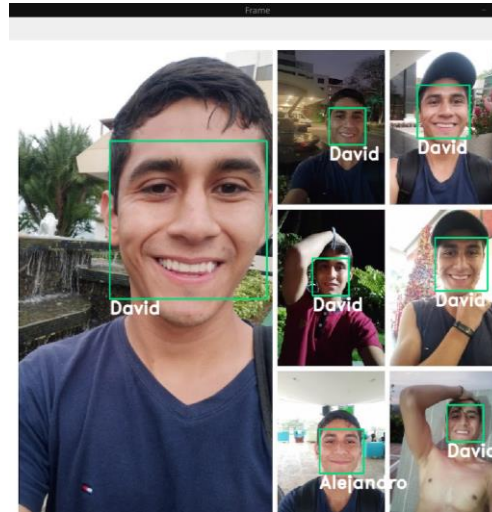
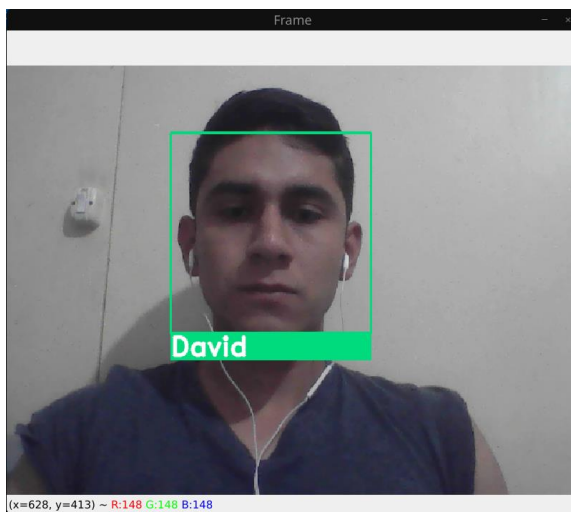
while True:
    ret, frame = cap.read()
    if ret == False:
        break
    frame = cv2.flip(frame, 1)
    orig = frame.copy()
    faces = faceClassif.detectMultiScale(frame, 1.1, 5)

    for (x, y, w, h) in faces:
        face = orig[y:y+h, x:x+w]
        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
        actual_face_encoding = face_recognition.face_encodings(face, known_face_locat
        result = face_recognition.compare_faces(facesEncodings, actual_face_encoding)
        #print(result)
        if True in result:
            index = result.index(True)
            name = facesNames[index]
            color = (125, 220, 0)
        else:
            name = "Desconocido"
            color = (50, 50, 255)

        cv2.rectangle(frame, (x, y, h), (x + w, y + h + 30), color, -1)
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, name, (x, y + h + 25), 2, 1, (255, 255, 255), 2, cv2.LINE_

```

Funcionamiento en tiempo real y por medio de una imagen



Inserción de una GUI al sistema de reconocimiento facial con Django

```
def detect_people(request):
    if request.method == 'POST' and request.FILES['detect_image']:
        # Obtiene la imagen desde el formulario
        uploaded_image = request.FILES['detect_image']
        image_path = os.path.join(settings.MEDIA_ROOT, 'Detecciones', uploaded_image.name)

        # Guarda la imagen en la carpeta "uploads"
        with open(image_path, 'wb') as destination:
            for chunk in uploaded_image.chunks():
                destination.write(chunk)

        faces_encodings, faces_names = get_faces_encodings()

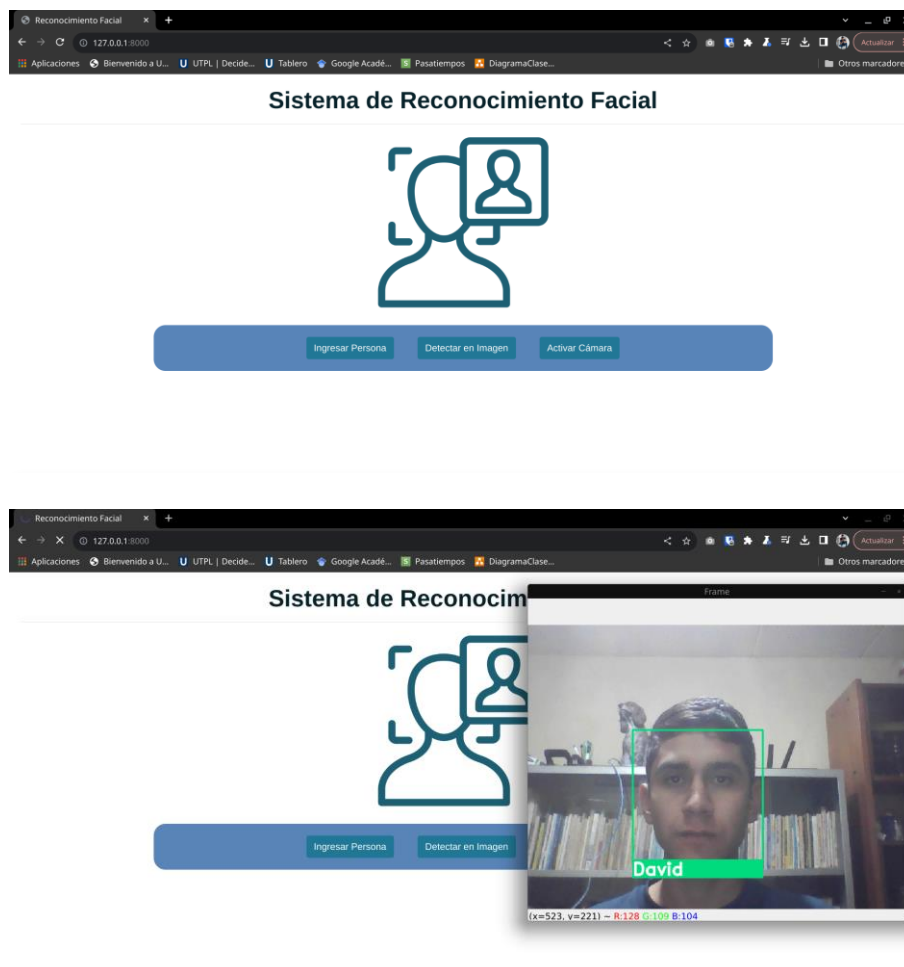
        frame = cv2.imread(image_path)
        orig = frame.copy()
        faces = face_recognition.face_locations(frame)

        for (top, right, bottom, left) in faces:
            face = orig[top:bottom, left:right]
            face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
            actual_face_encoding = face_recognition.face_encodings(face)

            if actual_face_encoding:
                # Si se detecta una cara, realiza la comparación
                result = face_recognition.compare_faces(faces_encodings, actual_face_encoding[0])

                if True in result:
                    index = result.index(True)
```

Proyecto desplegado y en funcionamiento



BIBLIOGRAFÍA

- Velázquez, A. (2018, agosto 15). ¿Cómo plantear un problema de investigación? QuestionPro. <https://www.questionpro.com/blog/es/como-plantear-un-problema-de-investigacion/>
- Alonso, M. (2023, febrero 9). Qué son los objetivos generales y específicos y cómo redactarlo [2023] •. Asana. <https://asana.com/es/resources/general-and-specific-objectives>
- Reconocimiento Facial. (2022, febrero 18). Innovatrics. <https://www.innovatrics.com/es/glosario/reconocimiento-facial/>
- Burns, E. (2021, septiembre 6). Aprendizaje profundo (deep learning). ComputerWeekly.es; TechTarget. <https://www.computerweekly.com/es/definicion/Aprendizaje-profundo-deep-learning>
- ¿Qué son las redes neuronales convolucionales? (s/f). Ibm.com. Recuperado el 9 de enero de 2024, de <https://www.ibm.com/es-es/topics/convolutional-neural-networks>
- ¿Qué es la segmentación de imágenes? ¿Cuáles son sus aplicaciones? (2023, abril 5). CaseGuard; CaseGuard Video Redaction Software. <https://caseguard.com/es/articles/que-es-la-segmentacion-de-imagenes-cuales-son-sus-aplicaciones/>
- Lahuerta, M. (2014, marzo 18). DeepFace, software de reconocimiento facial de Facebook. Computer Hoy. <https://computerhoy.com/noticias/software/deepface-software-reconocimiento-facial-facebook-10405>
- Informática, I., Diaz Y Diaz, J. F., Hector, R., Pasquale, D., Networks, N., Mcculloch, W., & Pitts, W. (s/f). Facultad de Ingeniería y Ciencias Agrarias. Edu.ar. Recuperado el 9 de enero de 2024, de <https://repositorio.uca.edu.ar/bitstream/123456789/11212/1/reconocimiento-facial-facenet-aws.pdf>
- Resiliente Digital. (2021, febrero 9). Qué es Clearview.ai y a qué se dedican. Resiliente Digital. <https://resilientdigital.com/que-es-clearview-ai-y-a-que-se-dedican/>
- Face recognition security software. (2019, mayo 6). NtechLab. <https://ntechlab.com/findface-multi/>