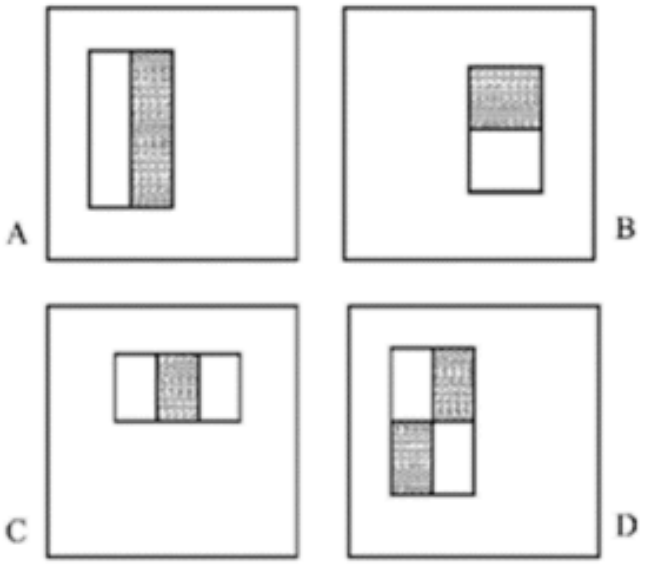
Haar Cascades (Continuación)

Esta técnica, desarrollada por Viola y Jones, se basa en la concatenación de varios clasificadores débiles para analizar diferentes partes de una imagen o frame de vídeo. Los clasificadores débiles tienen una alta probabilidad de dar falsos positivos individualmente, pero son potentes cuando se combinan. Se organizan las imágenes según el valor de las características (features), que son rectángulos o bloques de píxeles encontrados en ventanas de detección cerradas. Se utilizan datasets de imágenes positivas y negativas para el entrenamiento, seleccionando características relevantes y utilizando Adaboost para encontrar secciones relevantes de las imágenes. Se introduce el concepto de clasificación de cascada, donde las features se dividen en etapas para una evaluación eficiente. Se utiliza la función recursiva de la imagen integral para una rápida evaluación de las features, y si una etapa falla en la detección, se descarta la ventana. El orden de las etapas prioriza las características más relevantes del objeto. Menos etapas implican un mayor consumo de recursos para la detección del objeto.



Features de entrada definida por Viola y Jones

Adaboost es un algoritmo de aprendizaje automático que mejora la precisión de los clasificadores débiles al asignar pesos a las instancias de entrenamiento y combinar estos clasificadores para formar un clasificador fuerte. Se enfoca en las instancias mal clasificadas por clasificadores anteriores, asignándoles mayor peso, y los combina para lograr una alta precisión en la clasificación. Es resistente al sobreajuste y puede manejar datos de alta dimensionalidad.

El reconocimiento facial mediante Haar Cascades implica los siguientes pasos:

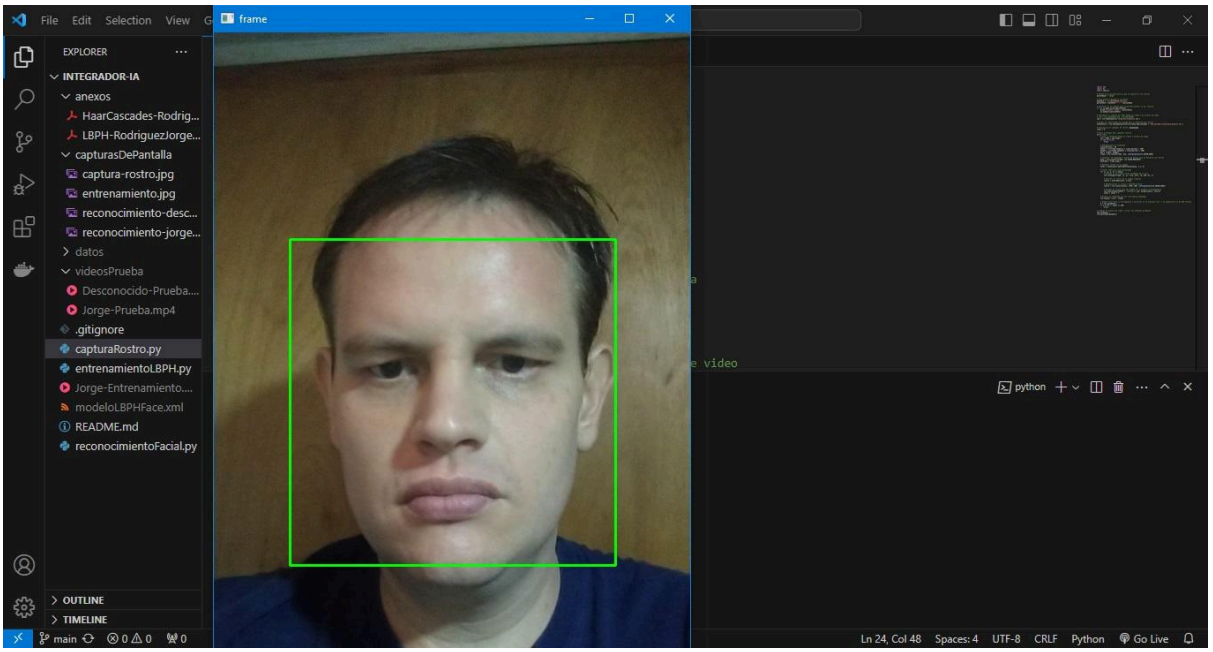
- Entrenamiento del clasificador: En esta etapa, se utiliza un conjunto de imágenes de entrenamiento que contienen ejemplos positivos de rostros (imágenes que contienen caras) y ejemplos negativos (imágenes que no contienen caras). El algoritmo utiliza

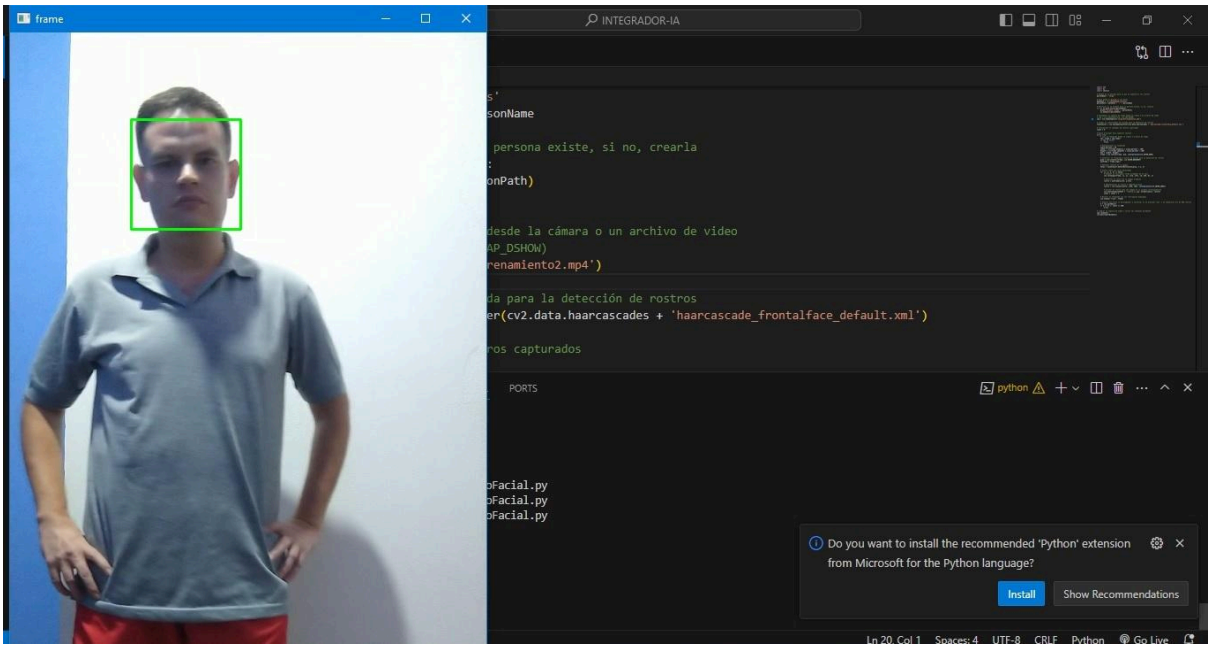
estas imágenes para aprender patrones visuales que distinguen entre rostros y no rostros.

- Extracción de características: Se seleccionan y calculan características importantes de la imagen, que son patrones rectangulares simples conocidos como "filtros Haar". Estos filtros detectan cambios de intensidad de píxeles en regiones específicas de la imagen, como bordes, líneas y áreas claras u oscuras.
- Creación de clasificadores débiles: Se construyen una serie de clasificadores débiles que son capaces de distinguir entre regiones de la imagen que podrían contener un rostro y regiones que no lo hacen. Estos clasificadores se basan en las características extraídas en el paso anterior.
- Clasificación en cascada: Los clasificadores débiles se organizan en una estructura de cascada, donde las regiones de la imagen se evalúan secuencialmente utilizando una serie de criterios. Si una región no pasa un criterio de clasificación, se descarta como no contenedora de un rostro. Solo las regiones que pasan todos los criterios son consideradas como posibles rostros y pasan a la siguiente etapa de la cascada.
- Refinamiento y detección: Las regiones que han pasado por todas las etapas de la cascada se someten a una evaluación más detallada para determinar si realmente contienen un rostro. Esto puede implicar el uso de técnicas adicionales, como la verificación de la presencia de características faciales específicas (ojos, nariz, boca) o la comparación con un modelo de referencia de rostro.

En el proyecto, el archivo 'haarcascade_frontalface_default.xml' (se hace referencia en reconocimiento-rostro-ia/capturaRostro.py, línea 22) es un archivo XML que contiene los datos de un clasificador de Haar Cascade preentrenado para la detección de caras frontales en imágenes. Este archivo es parte de la biblioteca OpenCV (Open Source Computer Vision Library) y se utiliza en aplicaciones de reconocimiento facial y detección de caras. El clasificador Haar Cascade incluido en 'haarcascade_frontalface_default.xml' ha sido entrenado utilizando un conjunto de datos de imágenes faciales que contienen ejemplos de caras frontales y no frontales. Durante el entrenamiento, el algoritmo Haar Cascade aprende patrones visuales que son característicos de las caras humanas, como la disposición de los ojos, la nariz y la boca, así como los cambios de contraste en diferentes partes de la cara. Una vez que el clasificador Haar Cascade ha sido entrenado y almacenado en el archivo XML, puede ser utilizado en aplicaciones de detección de caras. Cuando se aplica a una imagen, el clasificador examina la imagen utilizando una ventana deslizante y realiza una serie de operaciones para detectar regiones que puedan contener una cara frontal. Si encuentra una región que cumple con ciertos criterios de coincidencia con los patrones aprendidos durante el entrenamiento, marca esa región como una posible cara frontal.

Pruebas realizadas con video:





Pruebas realizadas con imágenes:

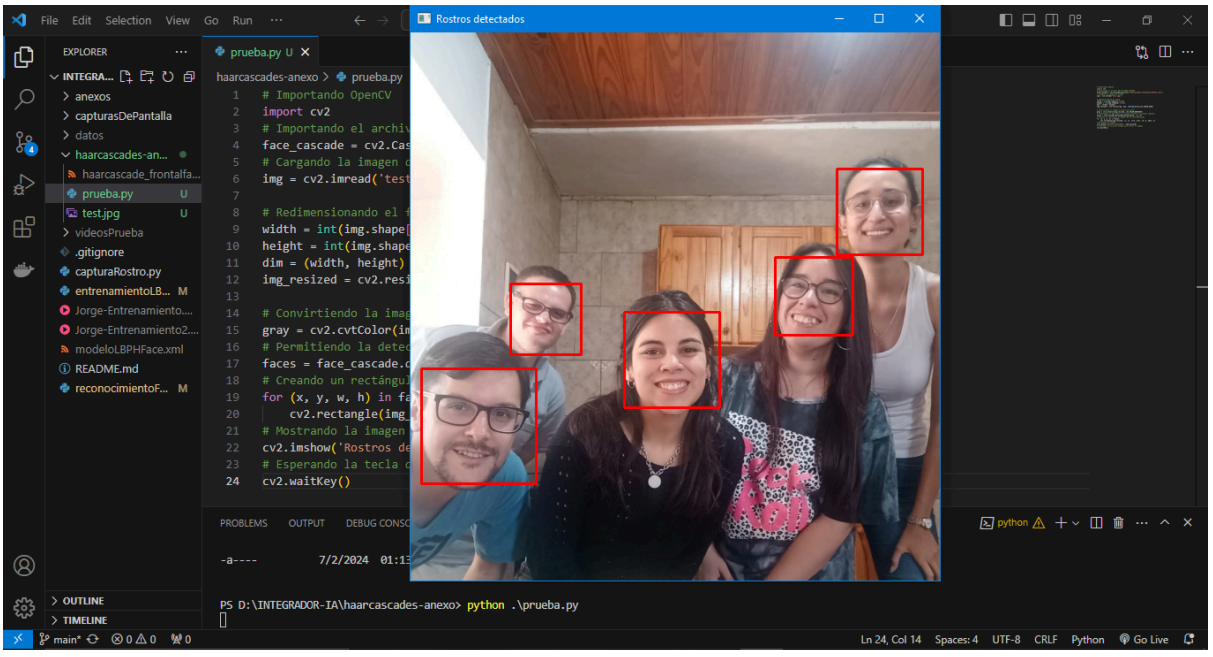
Código utilizado:

```python
# Importando OpenCV
import cv2
# Importando el archivo XML de HARR CASCADE
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# Cargando la imagen de prueba
img = cv2.imread('test.jpg')

# Redimensionando el frame
width = int(img.shape[1] * 0.5)
height = int(img.shape[0] * 0.5)
dim = (width, height)
img_resized = cv2.resize(img, dim, interpolation=cv2.INTER_AREA)

# Convirtiendo la imagen a escala de grises
gray = cv2.cvtColor(img_resized, cv2.COLOR_BGR2GRAY)
# Permitiendo la detección a múltiples escalas (múltiples tamaños)
faces = face_cascade.detectMultiScale(gray, 1.1, 6)
# Creando un rectángulo alrededor de la cara detectada
for (x, y, w, h) in faces:
    cv2.rectangle(img_resized, (x, y), (x+w, y+h), (0, 0, 250), 2)
# Mostrando la imagen
cv2.imshow('Rostros detectados', img_resized)
# Esperando la tecla de escape para cerrar la imagen
cv2.waitKey()
```

Detección de rostros:



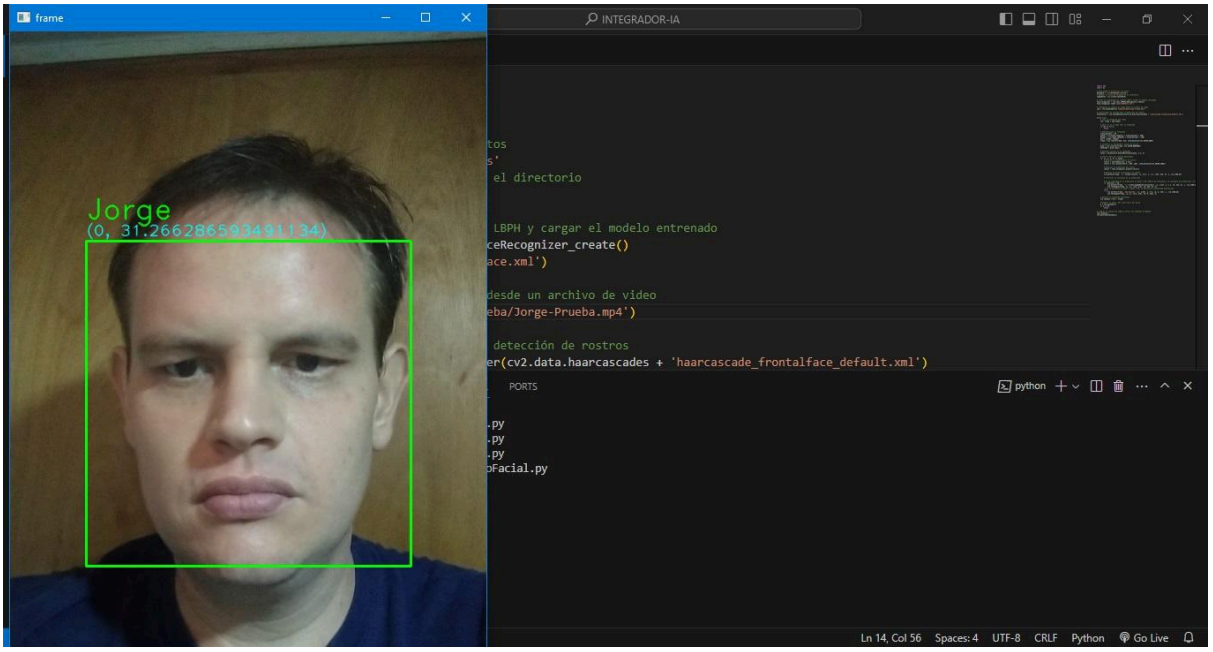
Relación entre LBP y redes neuronales

La relación entre Local Binary Patterns Histograms (LBP) y redes neuronales se centra en cómo pueden utilizarse juntos para tareas de reconocimiento de patrones. Esto incluye:

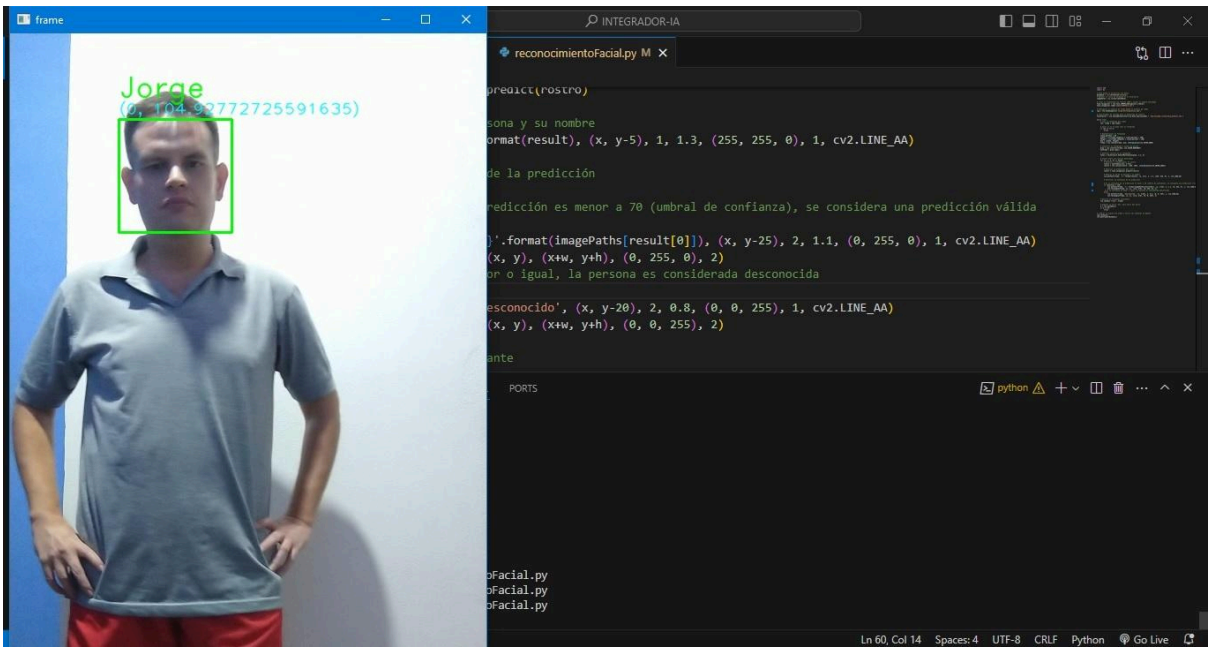
- Extracción de características: LBP puede ser utilizado para extraer características de las imágenes antes de alimentarlas a una red neuronal, mejorando así la representación de los datos de entrada.
- Preprocesamiento: Los histogramas LBP pueden ser útiles para normalizar o mejorar la representación de las imágenes antes de ser procesadas por una red neuronal, lo que puede mejorar su capacidad de aprender patrones relevantes.
- Complementariedad: Los histogramas LBP pueden capturar características únicas que complementan las características aprendidas por la red neuronal, lo que puede mejorar el rendimiento del sistema de reconocimiento o clasificación.
- Inicialización de pesos: Los histogramas LBP pueden ser utilizados para inicializar los pesos de las capas iniciales de una red neuronal, acelerando así el proceso de entrenamiento.
- Arquitecturas híbridas: Es posible diseñar arquitecturas de red neuronal que incluyan capas específicamente diseñadas para procesar características extraídas por LBP, lo que puede mejorar la representación de características de alto nivel.
- Aprendizaje conjunto: Se pueden desarrollar sistemas que aprendan simultáneamente las características y la tarea de clasificación utilizando una combinación de métodos basados en LBP y redes neuronales, lo que puede conducir a una representación de características más robusta y adaptativa.

Pruebas realizadas:

Con videos “medio cuerpo” con confianza de predicción con umbral 70, el algoritmo reconocía los rostros.



Luego se probó con videos de cuerpo entero, y el umbral superaba el valor de 100, las detecciones eran erróneas, se ocupó otra luz, otro fondo.



Nota: para el óptimo funcionamiento del reconocimiento con LBPH, se debe capturar y entrenar rostros con imágenes que no sean de cuerpo entero, en el ambiente donde se desea realizar el reconocimiento y con la luminocidad adecuada.

Bibliografía

- Reconocimiento de objetos a través de la metodología Haar Cascades. UTN - Facultad Regional Córdoba. Recuperado de https://confedi.org.ar/reconocimiento-de-objetos-a-traves-de-la-metodologia-haar-cascades/ [06/02/2024].
- Consultas a ChatGPT. Recuperado de: https://openai.com/blog/chatgpt [07/02/2024].