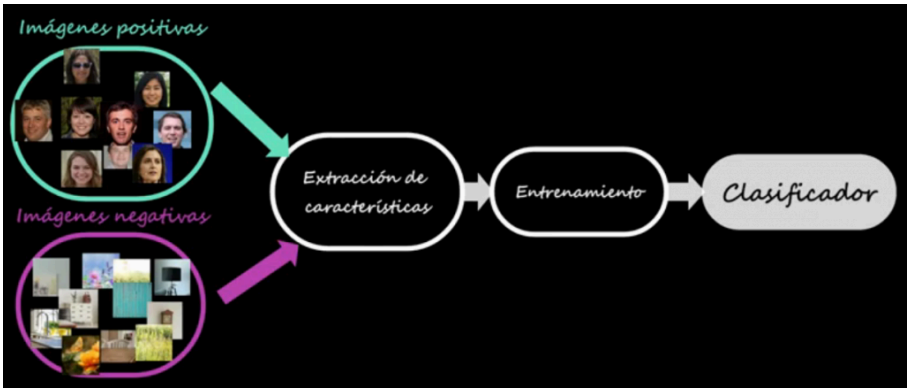
DETECCIÓN DE ROSTROS con Haar Cascades Python

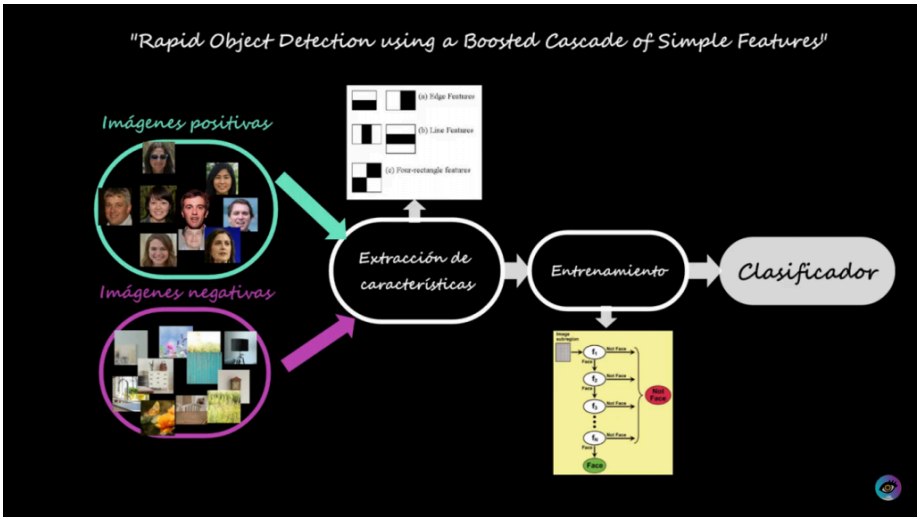
La detección de rostros es una técnica que identifica rostros o caras en una imagen, excluyendo el fondo y otros objetos. Se utiliza comúnmente en aplicaciones diarias como Facebook, donde se sugiere etiquetar a las personas en las imágenes, y en filtros de Instagram que requieren la detección de rostros para aplicarse. También se encuentra en aplicaciones bancarias y en la función de fotografía de smartphones, donde destaca los rostros en la imagen.

¿Cómo se detectan los rostros en una imagen?

Detectar un rostro en una imagen es una tarea desafiante para una computadora, por lo que se utiliza el aprendizaje automático o machine learning. Para entrenar un clasificador, se necesitan muchas imágenes, tanto positivas (con rostros) como negativas (sin rostros). Las características de todas las imágenes se extraen y se utiliza el enfoque de machine learning para el entrenamiento. Al final, se obtiene un clasificador capaz de discernir entre la presencia y ausencia de un rostro en nuevas imágenes.



Proceso para crear un clasificador de rostros empleando Haar-Cascade



En el proceso de detección de rostros, se emplean imágenes positivas y negativas. La extracción de características se lleva a cabo mediante características de Haar. Para el entrenamiento del clasificador, se utiliza una cascada de clasificadores que descarta áreas no consideradas como rostros y continúa analizando posibles candidatos a ser un rostro. Este enfoque permite obtener un detector de rostros que puede identificar eficientemente las caras en nuevas imágenes.

Haar Cascades Python – OpenCV

El uso de Haar Cascades en Python con OpenCV ofrece clasificadores preentrenados para diversos objetos, como rostros, ojos, sonrisas, e incluso caras de gatos. Estos clasificadores están disponibles en el repositorio de OpenCV en GitHub: https://github.com/opencv/opencv/tree/master/data/haarcascades.

La función clave para la detección de rostros es detectMultiScale, la cual utiliza un clasificador para encontrar objetos en una imagen. Esta función devuelve un rectángulo delimitador que rodea el objeto detectado. Algunos de los argumentos importantes para esta función son:

- Image: La imagen en la que se realizará la detección de rostros.
- ScaleFactor: Este parámetro indica cuánto se reducirá la imagen en cada iteración. Un valor como 1.1 significa una reducción del 10%, y un valor como 1.3 significa una reducción del 30%. Valores demasiado altos pueden perder detecciones, mientras que valores muy bajos aumentan el tiempo de procesamiento y pueden generar falsos positivos.
- minNeighbors: indica cuántos rectángulos vecinos deben estar presentes para que un candidato se retenga como una detección válida. Cuando se buscan rostros en una imagen, este parámetro ayuda a evitar la multiplicidad de rectángulos para un mismo rostro. En otras palabras, establece el número mínimo de rectángulos cercanos que debe tener un rostro para ser considerado como una detección confirmada, reduciendo así posibles falsos positivos y mejorando la precisión del algoritmo de detección de rostros.
- MinSize: Este parámetro indica el tamaño mínimo posible del objeto. Ignora objetos más pequeños.
- MaxSize: Este parámetro indica el tamaño máximo posible del objeto. Ignora objetos más grandes.