



Tema de la
presentacion:



Libreria: OpenCV



¿Que es OpenCV?

OpenCV (Open Source Computer Vision Library) es una libreria de visión por computadora y procesamiento de imágenes que ofrece más de 2500 algoritmos optimizados, que incluyen una amplia gama de tareas como el procesamiento de imágenes, el reconocimiento facial, la detección de objetos entre otros más.



¿Porque OpenCV es popular?

- Facilidad de uso: Python es un lenguaje conocido por su sintaxis sencilla y legible, lo que facilita el aprendizaje y la implementación de algoritmos de visión por computadora.
- Integración con otras bibliotecas: Se integra fácilmente con otras bibliotecas de Python como NumPy, Matplotlib y scikit-learn, lo que permite realizar análisis de datos más complejos.



Conceptos basicos:

- Imágenes: Son representadas como matrices de píxeles. En OpenCV, las imágenes suelen ser leídas en formato BGR (azul, verde, rojo).
- Detección de Bordes (Canny): Un algoritmo que identifica los bordes más relevantes de una imagen, utilizando gradientes de intensidad.
- Seguimiento de objetos: Rastrear el movimiento de objetos a lo largo de un video.



Formatos:

OpenCV soporta varios formatos de imagen como JPEG, PNG, BMP, TIFF y WebP, y formatos de video como AVI, MP4, MOV y MKV. También es compatible con códecs como H.264, MJPEG y XVID. Puedes usar `cv2.imread` para cargar imágenes y `cv2.VideoCapture` para cargar videos, mientras que `cv2.imwrite` y `cv2.VideoWriter` permiten guardar imágenes y videos en diferentes formatos. Cada formato ofrece distintas ventajas según la calidad, compresión y uso.



Características

- Procesamiento de imágenes: Manipulación de imágenes (ajuste de tamaño, filtros, etc.).
- Detección de características: Detección de bordes, contornos, y puntos de interés en imágenes.
- Visión por computadora: Detección de objetos, reconocimiento facial y clasificación de imágenes.
- Procesamiento de video: Captura y análisis de videos en tiempo real.
- Reconocimiento facial y de objetos: Detección y reconocimiento de caras y objetos.



```
from '@material-ui/core'; 63.32 kB (gzip: 20.3
from '../graphql/client';

e, darkTheme } from '../utils/theme';
page from '../hooks/useLocalStorage';

m '../components/NavBar';

component, pageProps }: AppProps) {
tTheme, setCurrentTheme] = useLocalStorage(key
lient = useApollo(pageProps.initialApolloState

ffect: () => {
...useSelector(selectors: '#iss
```




Casos de Usos

- Detección de rostros: Utilizando clasificadores en cascada, OpenCV puede detectar rostros en imágenes y videos en tiempo real
- Reconocimiento de objetos: Con técnicas como la detección de contornos y el uso de modelos preentrenados, puedes identificar y clasificar objetos en imágenes
- Procesamiento de imágenes: OpenCV permite realizar operaciones como el ajuste de tamaño, la rotación, el filtrado y la detección de bordes en imágenes.



- Seguimiento de objetos: Puedes rastrear el movimiento de objetos en videos, lo cual es útil en aplicaciones de vigilancia y deportes.
- Realidad aumentada: Integrando OpenCV con otras bibliotecas, puedes superponer información digital sobre el mundo real, creando experiencias de realidad aumentada.

Pasos para la instalacion

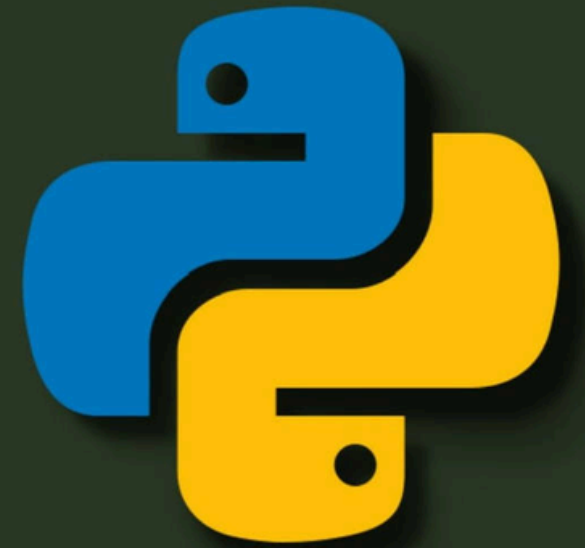
Para instalar OpenCV, podemos abrir una terminal y colocar lo siguiente:

- OpenCV y Matplotlib se pueden instalar con pip en la terminal

```
pip install opencv-python matplotlib
```

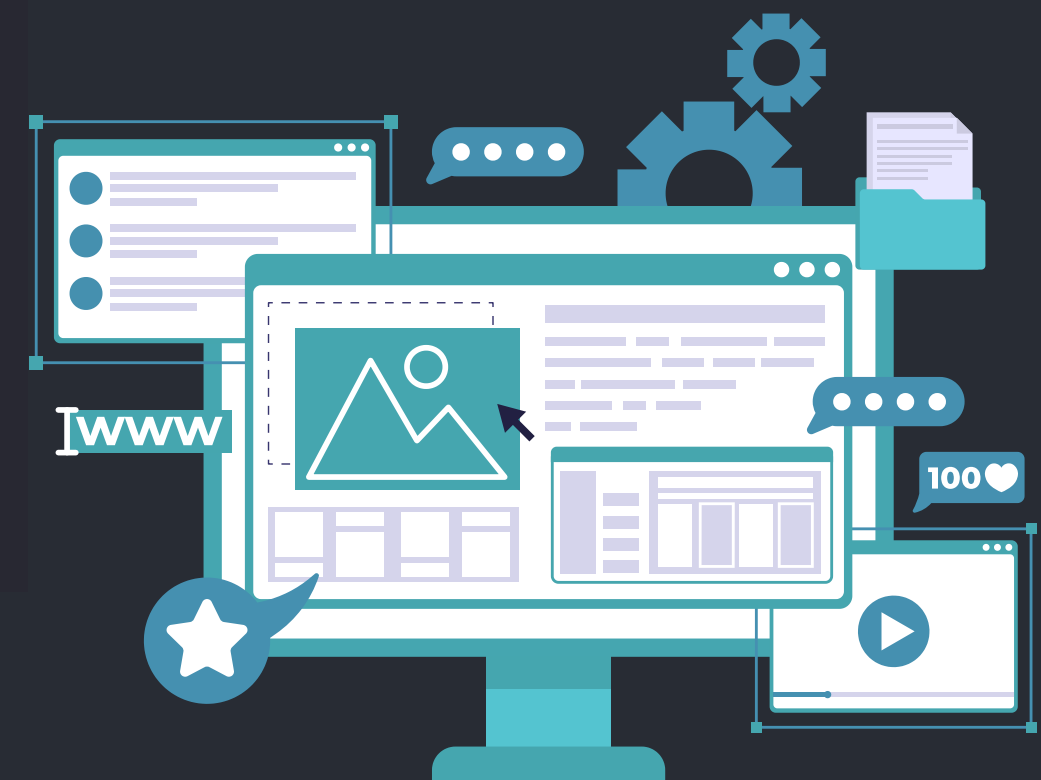
Verificamos la instalación: Para verificar que todo esté instalado correctamente, podemos ejecutar el siguiente código:

```
import cv2  
print(cv2.__version__)
```



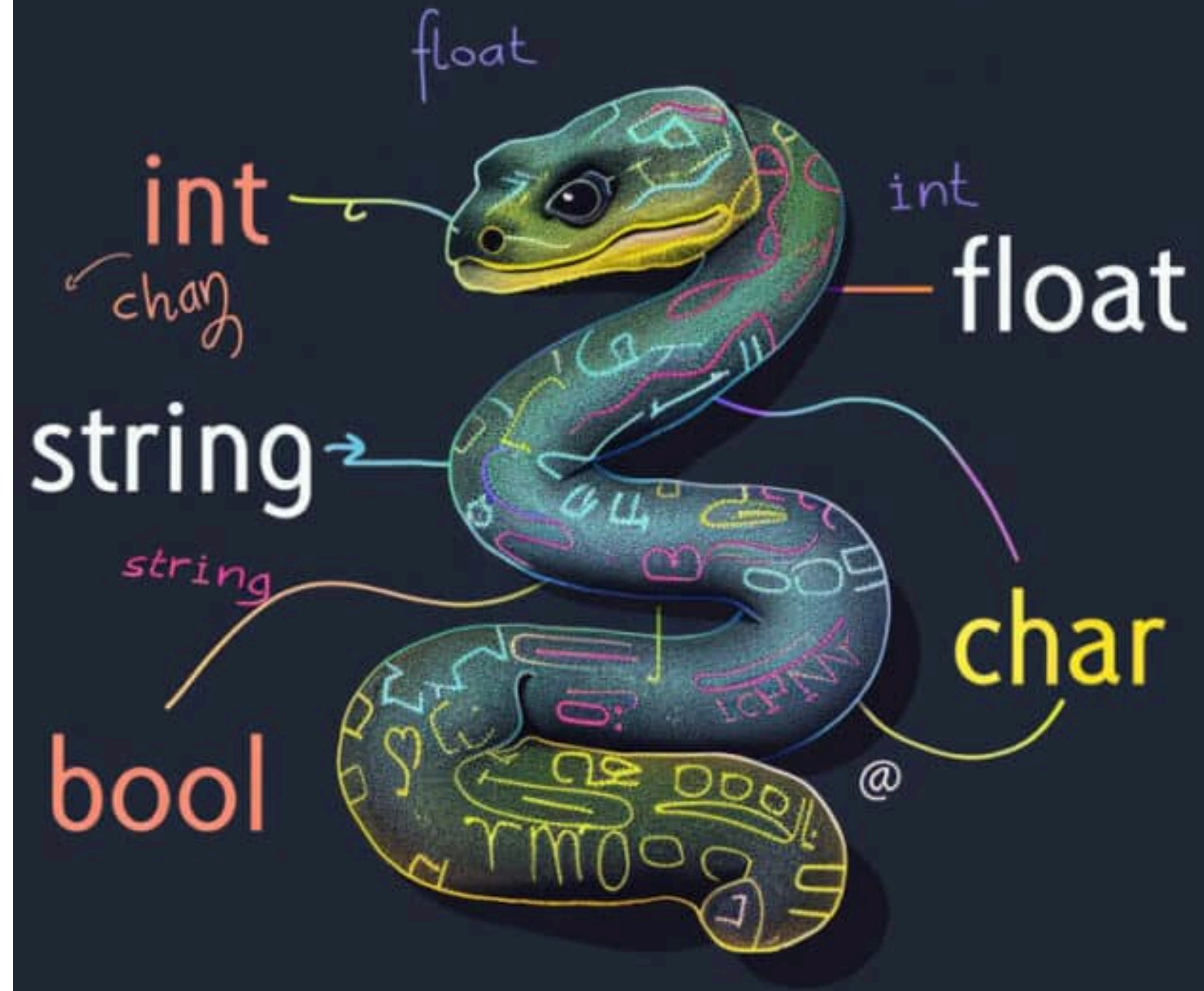
Ejemplo Practico

```
1  import cv2
2  # Iniciar captura de video (webcam)
3  video = cv2.VideoCapture(0) # Usa '0' para webcam
4  # Cargar el clasificador preentrenado de rostros
5  face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')
6  while True:
7      ret, frame = video.read() # Leer un frame del video
8
9      if not ret:
10         print("No se pudo capturar el frame.")
11         break
12     # Convertir a escala de grises
13     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
14     # Detectar rostros en la imagen
15     cara = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
16     # Dibujar rectángulos alrededor de los rostros detectados
17     for (x, y, w, h) in cara:
18         cv2.rectangle(frame, (x, y), (x + w, y + h), (10, 255, 10), 2)
19
20     # Mostrar el video con los rectángulos dibujados
21     cv2.imshow('Detección de Rostros', frame)
22     # Cerrar ventana con la tecla 'x'
23     if cv2.waitKey(1) & 0xFF == ord('x'):
24         break
25 # Liberar recursos
26 video.release()
27 cv2.destroyAllWindows()
```



Conclusion

En conclusión, OpenCV es una herramienta poderosa para cualquier tarea relacionada con procesamiento de imágenes y visión por computadora. Es fácil de integrar con otras bibliotecas y lenguajes, y su amplia gama de funcionalidades la convierte en una opción popular tanto en proyectos académicos como industriales.



Recursos:

<https://imaginaformacion.com/tutoriales/opencv-en-python>

<https://es.stackoverflow.com/questions/349957/convertir-una-imagen-a-escala-de-grises-pero-conservando-los-canales-python>

<https://www.youtube.com/watch?v=kUMjVo25kX0&t=320s>