Introduccion y funcionamiento de OpenCV

TIC

Introducción a OpenCV

OpenCV

OpenCV, que significa Open Source Computer Vision Library, es una biblioteca de visión por computadora de código abierto y gratuita que proporciona un amplio conjunto de herramientas y algoritmos para el procesamiento de imágenes y la visión artificial. Desarrollada originalmente por Intel en 1999, OpenCV se ha convertido en una de las bibliotecas más utilizadas en el campo de la visión por computadora debido a su eficiencia, versatilidad y amplia gama de características.

Ver la documentcion oficial en: https://opencv.org/

Características principales de OpenCV

Ofrece una amplia gama de funciones y algoritmos para el procesamiento de imágenes, incluidos operaciones básicas como carga, visualización y guardado de imágenes, así como operaciones más avanzadas como transformaciones geométricas, filtros, segmentación, detección de bordes, y extracción de características.

Está implementado en C/C++, lo que le confiere una alta eficiencia y rendimiento. Además, la biblioteca aprovecha la capacidad de procesamiento paralelo de los procesadores modernos y puede ser optimizada para su ejecución en GPU, lo que acelera significativamente el procesamiento de imágenes en aplicaciones de alto rendimiento.

Es compatible con varios sistemas operativos, incluyendo Windows, Linux, macOS, Android e iOS, lo que permite su uso en una amplia variedad de plataformas y dispositivos.

Aunque OpenCV está escrita principalmente en C/C++, también proporciona interfaces para varios otros lenguajes de programación, incluyendo Python, Java y MATLAB, lo que facilita su integración en aplicaciones desarrolladas en estos lenguajes.

Incluye algoritmos de aprendizaje automático para tareas de reconocimiento de objetos, seguimiento de objetos, detección de rostros, reconocimiento de gestos, entre otros. Estos algoritmos permiten desarrollar aplicaciones avanzadas de visión por computadora sin la necesidad de implementar algoritmos complejos desde cero.

La librería proporciona funciones para acceder a cámaras web, cámaras de video, archivos de video y otros dispositivos de entrada, lo que facilita el desarrollo de aplicaciones d procesamiento de imágenes en tiempo real, como sistemas de vigilancia, reconocimiento de gestos y realidad aumentada.

Cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen con el desarrollo y la mejora continua de la biblioteca.
Además, existen numerosos recursos educativos, tutoriales y documentación en línea que facilitan el aprendizaje y la utilización de OpenCV en proyectos de visión por computadora.

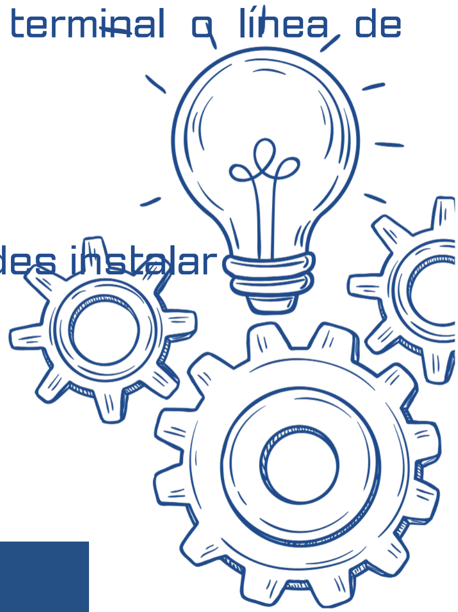
TIC

OpenCV es una biblioteca poderosa y versátil que proporciona una amplia gama de herramientas y algoritmos para el procesamiento de imágenes y la visión artificial. Su eficiencia, compatibilidad multiplataforma y amplio conjunto de características la convierten en una opción popular para desarrolladores y científicos en el campo de la visión por computadora.

Instalación y configuración de OpenCV en entornos de desarrollo

Para comenzar a trabajar con OpenCV en Python, primero necesitas instalar la biblioteca en tu entorno de desarrollo. Para la instalación e inicio con OpenCV en Python con **pip**, el gestor de paquetes de Python, debes abrir una terminal o línea de comandos y ejecutar el siguiente comando:

```
pip install opencv-python
```



Si estás utilizando el entorno de **conda**, puedes instalar OpenCV con el siguiente comando:

```
conda install -c conda-forge opencv
```

Inicio con OpenCV en

Operaciones básicas de manipulación de imágenes

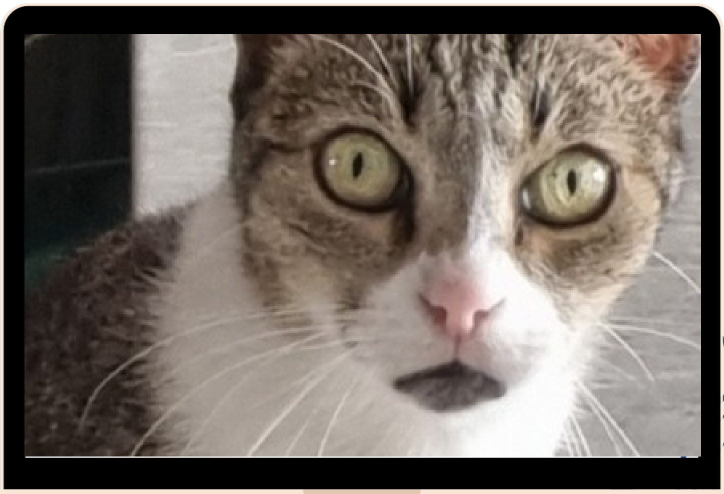
Carg **Visualizaci** **Guarda**

Una vez que OpenCV esté instalado en tu entorno de desarrollo, puedes

comenzar a utilizarlo en tus proyectos de Python.

- ## **Ejemplo básico para cargar y mostrar una imagen utilizando OpenCV:**

```python
import cv2

# Cargar una imagen desde el archivo
image = cv2.imread('imagen.jpg')

# Mostrar la imagen en una ventana
cv2.imshow('Imagen', image)

# Esperar a que el usuario presione una tecla para cerrar la ventana
cv2.waitKey(0)

# Cerrar todas las ventanas abiertas
cv2.destroyAllWindows()
```



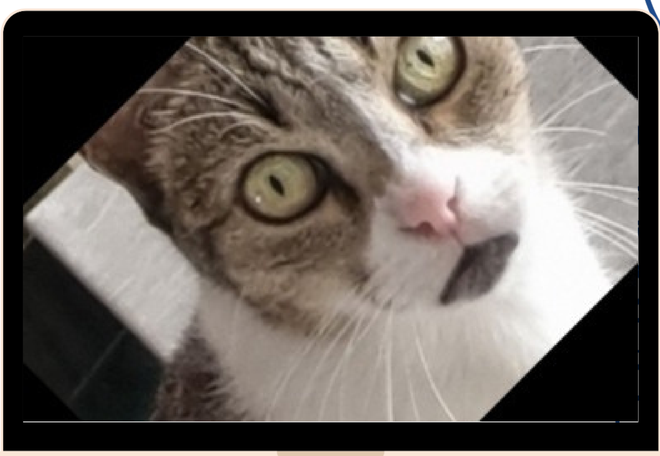
Este código carga una imagen llamada 'imagen.jpg' desde el directorio actual, la muestra en una ventana utilizando OpenCV y espera a que el usuario presione una tecla para cerrar la ventana.

TIC

Transformaciones geométricas

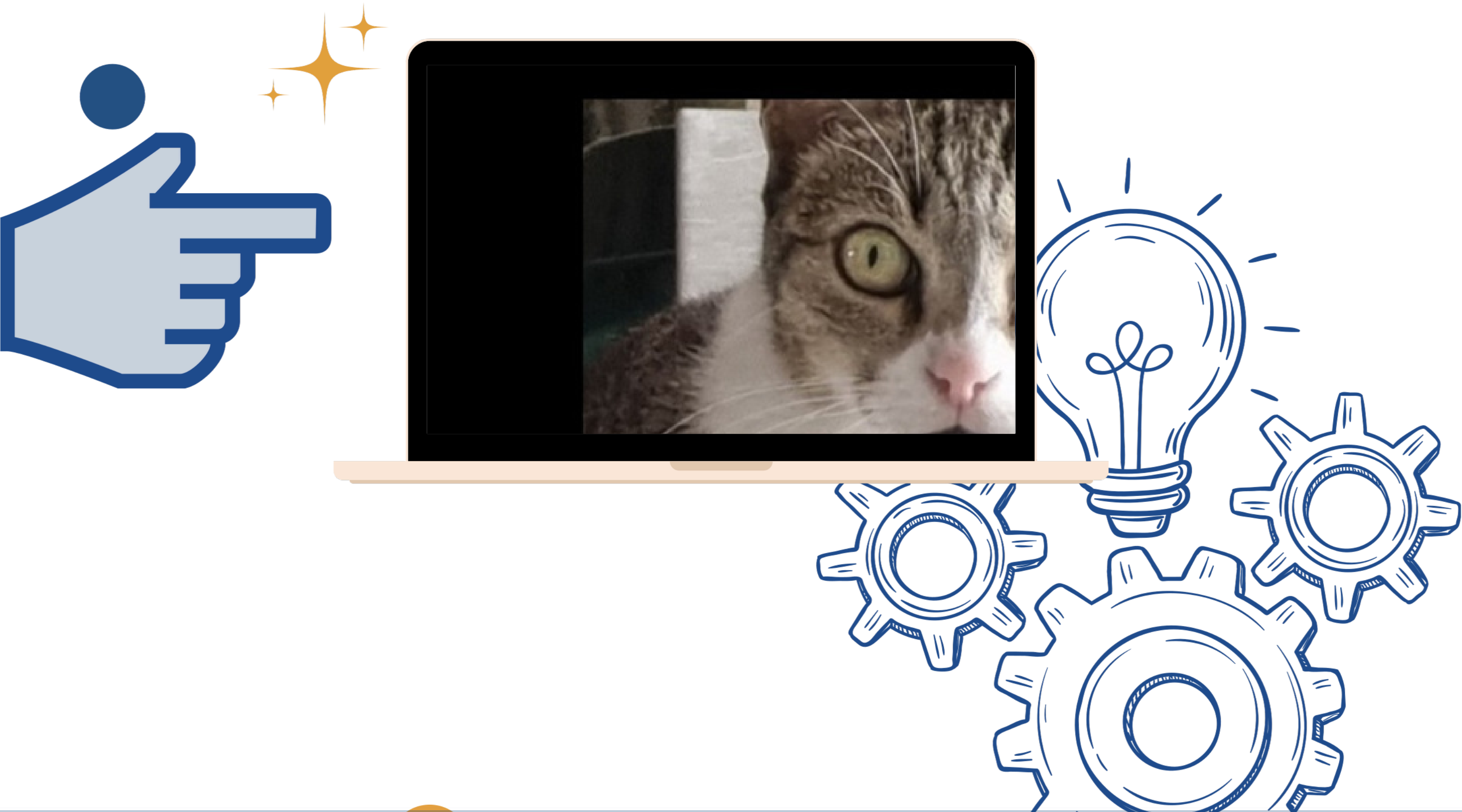
- **Rotación**

```python
import cv2
import numpy as np

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Obtener dimensiones de la imagen
height, width = image.shape[:2]

# Calcular el centro de la imagen
center = (width / 2, height / 2)

# Definir la matriz de rotación
angle = 45
M = cv2.getRotationMatrix2D(center, angle, 1.0)

# Aplicar la rotación a la imagen
rotated = cv2.warpAffine(image, M, (width, height))

# Mostrar la imagen rotada
cv2.imshow('Rotada', rotated)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Traslación

```python
import cv2
import numpy as np

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Definir la matriz de traslación
tx, ty = 100, 50
M = np.float32([[1, 0, tx], [0, 1, ty]])

# Aplicar la traslación a la imagen
translated = cv2.warpAffine(image, M, (image.shape[1], image.shape[0]))

# Mostrar la imagen trasladada
cv2.imshow('Traslación', translated)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

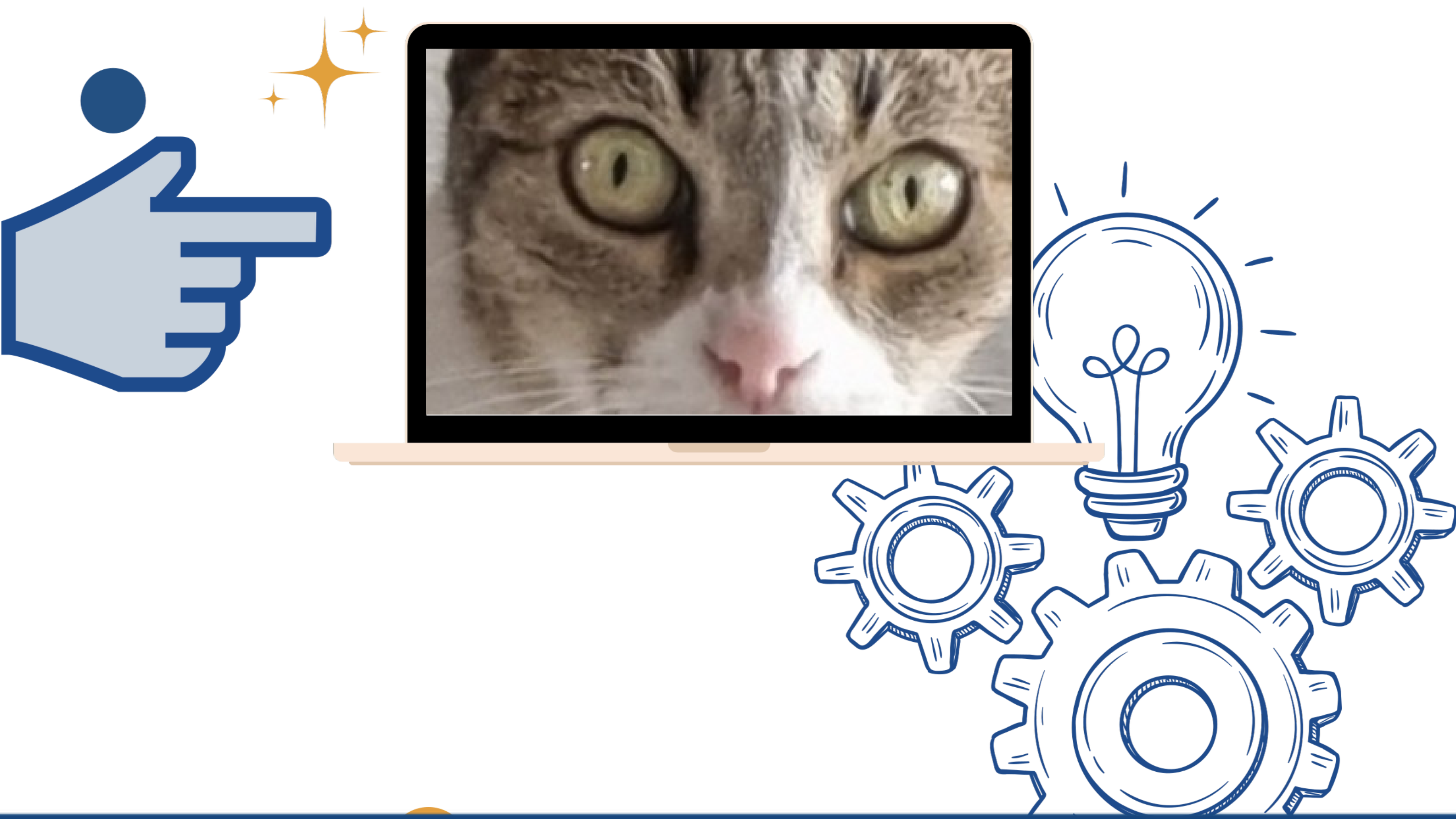


- **Escala**

```python
import cv2

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Definir la nueva anchura y altura
new_width, new_height = 300, 200

# Aplicar la escala a la imagen
scaled = cv2.resize(image, (new_width, new_height))

# Mostrar la imagen escalada
cv2.imshow('Escalada', scaled)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- ## Recorte

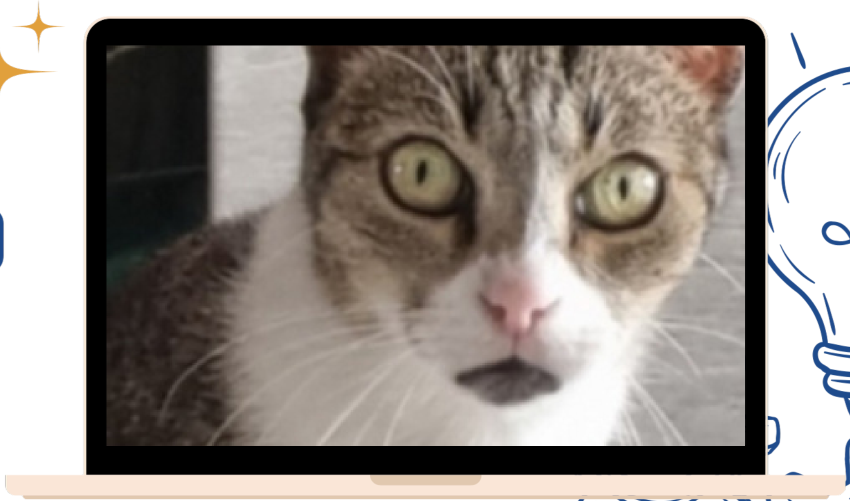
```python
import cv2

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Definir las coordenadas del área de interés (ROI)
x, y, w, h = 100, 100, 200, 150

# Recortar la región de interés (ROI)
cropped = image[y:y+h, x:x+w]

# Mostrar la imagen recortada
cv2.imshow('Recorte', cropped)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Filtrado de imágenes

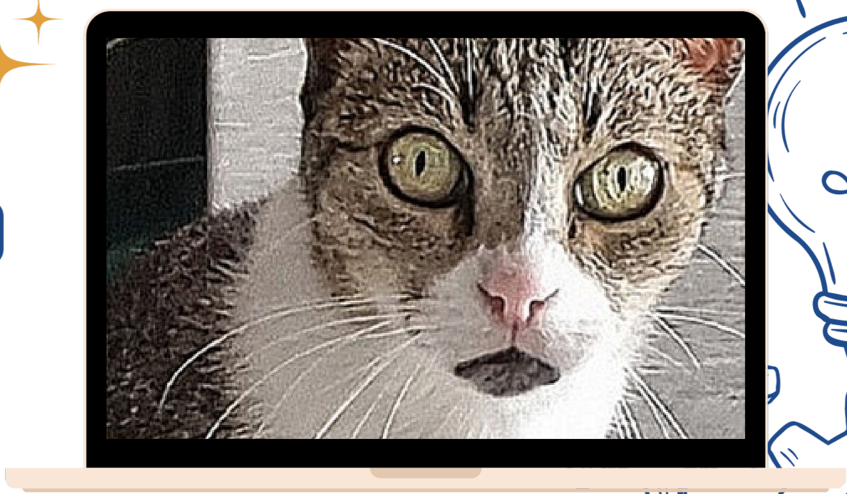
- ## Suavizado

```
import cv2

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Aplicar el filtro Gaussiano para suavizar la imagen
smoothed = cv2.GaussianBlur(image, (5, 5), 0)

# Mostrar la imagen suavizada
cv2.imshow('Suavizado', smoothed)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



- ## Realce

```python
import cv2
import numpy as np

# Cargar la imagen
image = cv2.imread('imagen.jpg')

# Definir el kernel para el filtro de afilado
kernel = np.array([[-1, -1, -1],
                   [-1,  9, -1],
                   [-1, -1, -1]])

# Aplicar el filtro de afilado para realzar los detalles
sharpened = cv2.filter2D(image, -1, kernel)

# Mostrar la imagen realzada
cv2.imshow('Realce', sharpened)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



- ## Detección de bordes

```python
import cv2

# Cargar la imagen en escala de grises
image = cv2.imread('imagen.jpg', cv2.IMREAD_GRAYSCALE)

# Aplicar el operador Sobel para detectar bordes
sobelx = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=5)
sobely = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=5)

# Combinar las respuestas en magnitud
edges = cv2.magnitude(sobelx, sobely)

# Normalizar los valores para mostrar la imagen correctamente
edges = cv2.normalize(edges, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

# Mostrar la imagen con bordes detectados
cv2.imshow('Detección de Bordes', edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Ejercicios relacionados con

1
Ejercicio de Carga y Visualización de Imágenes

- Carga una imagen en color y en escala de grises utilizando
- OpenCV. Muestra ambas imágenes en ventanas separadas.

2
Ejercicio de Transformación Geométrica

- Carga una imagen y rota la imagen 90 grados en sentido horario. Escala la
- imagen al doble de su tamaño original. Muestra ambas imágenes resultantes.
-

3
Ejercicio de Filtrado

- Carga una imagen en escala de
- grises. Aplica un filtro Gaussiano para suavizar la imagen. Aplica un
- filtro de realce para mejorar los detalles de la imagen. Muestra
- ambas imágenes resultantes.

4
Ejercicio de Detección de Bordes

- Carga una imagen en escala de
- grises. Utiliza el operador Sobel para detectar bordes en la imagen.
- Muestra la imagen original y la imagen con los bordes detectados.