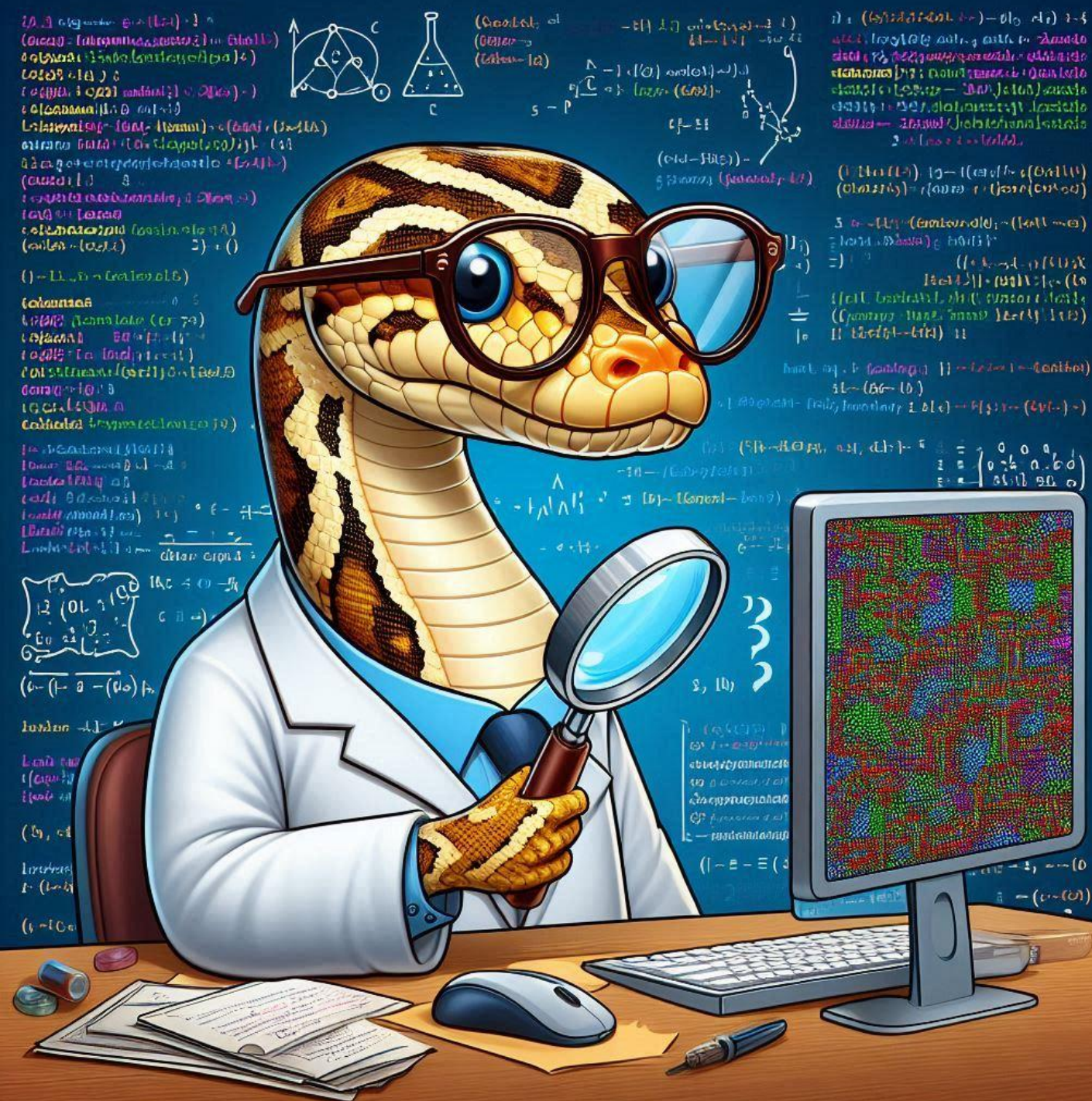


Transformações Visuais

O Poder do OpenCV



Lucas Fonseca

01

INTRODUÇÃO

Visão Computacional

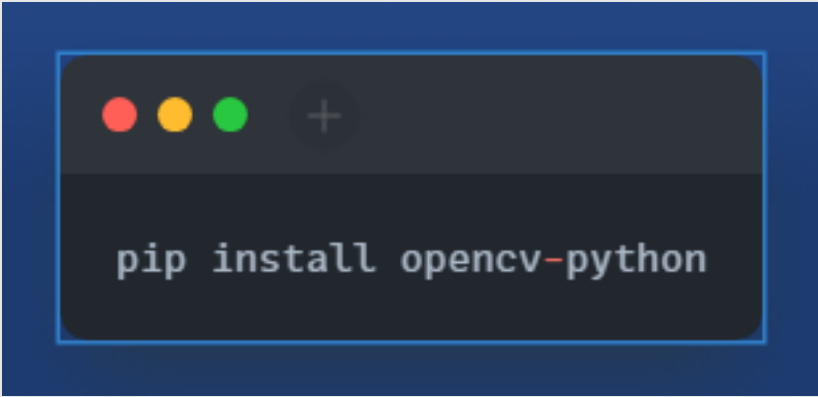
A visão computacional é um campo da inteligência artificial que capacita as máquinas a "verem" e entenderem o mundo visual, assim como nós fazemos. Imagine ensinar um robô a reconhecer uma maçã entre várias frutas. É isso que a visão computacional faz: permite que computadores interpretem e façam sentido de imagens e vídeos.

OpenCV

Agora, vamos falar sobre OpenCV, a biblioteca queridinha para quem está começando com visão computacional. OpenCV (Open Source Computer Vision Library) é uma biblioteca poderosa e open-source que facilita o trabalho com imagens e vídeos.

Instalação

Primeiro, você precisa instalar o OpenCV. Se você estiver usando Python, basta rodar:

A screenshot of a terminal window with a dark background and a blue border. The window has three colored circles (red, yellow, green) and a plus sign in the top-left corner. The text `pip install opencv-python` is displayed in the terminal.

```
pip install opencv-python
```



02

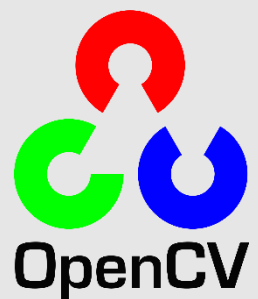
**Importar
Imagens**

Carregando e Exibindo Imagens

Vamos começar com algo simples: carregar e exibir uma imagem. Pense na imagem como uma matriz gigante de pixels. Cada pixel tem um valor que representa sua cor. Aqui está como fazer isso com OpenCV:

```
# Converte para escala de cinza
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Exibe a imagem em escala de cinza
cv2.imshow('Imagem em Escala de Cinza', img_gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



03

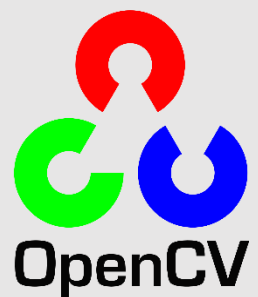
**Trabalhando
com
Imagens**

Convertendo para Escala de Cinza

Agora, vamos dar um toque artístico. Que tal converter essa imagem para escala de cinza? É como ver o mundo em preto e branco, tipo aqueles filmes antigos.

```
# Converte para escala de cinza
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Exibe a imagem em escala de cinza
cv2.imshow('Imagem em Escala de Cinza', img_gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



04

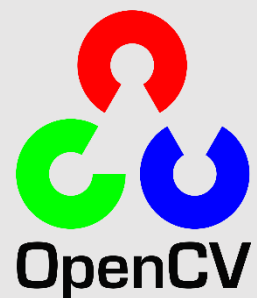
Detecção

Detecção de Bordas

Vamos avançar um pouco mais e fazer uma detecção de bordas, algo muito útil para identificar contornos de objetos em uma imagem. Pense nisso como desenhar as linhas principais de um desenho.

```
# Aplica o detector de bordas Canny
edges = cv2.Canny(img_gray, 100, 200)

# Exibe a imagem com bordas
cv2.imshow('Detecção de Bordas', edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Fim