

Introdução

A detecção de objetos em imagens é uma área de pesquisa da tecnologia mais promissora da atualidade. É utilizada em diversos setores, como indústria e saúde, possibilitando automatizar processos industriais até a assistência médica. A qualidade dos dados de entrada é fundamental para o bom funcionamento desses sistemas. Sendo assim, o pré-processamento de imagens é uma etapa crucial para garantir a qualidade dos dados de entrada, permitindo que os algoritmos de detecção consigam extrair informações precisas em um menor tempo de processamento.

Quais são as principais bibliotecas e frameworks para pré-processamento de imagens?

A utilização de bibliotecas e frameworks especializadas em processamento de imagem é uma etapa fundamental, pois essas ferramentas possuem classes e funções pré-estabelecidas que promovem agilidade e qualidade no código. Dentre elas destacam-se a **OpenCV**, **TensorFlow** e **PyTorch**.

OpenCV: A biblioteca de visão computacional mais utilizada, oferecendo um amplo conjunto de ferramentas para diversas tarefas, como leitura, escrita, transformação e análise de imagens.

TensorFlow e PyTorch: Frameworks de deep learning que incluem módulos para pré-processamento de imagens, como normalização e aumento de dados.

Aplicação

A detecção de objetos em imagens, é um método frequentemente utilizado em diversas áreas, exige um pré-processamento aprimorado das imagens para otimizar o desempenho dos algoritmos de detecção. O pré-processamento serve para melhorar a imagem, para que as próximas etapas sejam bem-sucedidas.

O pré-processamento de imagens é uma etapa crucial em diversas tarefas, como reconhecimento facial, segmentação e classificação de objetos. Ele abrange a preparação das imagens para que sejam adequadas para o modelo de aprendizado de máquina. O Python, com suas bibliotecas como **OpenCV**, **TensorFlow** e **PyTorch**, oferece diversos recursos para realizar essas tarefas.

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

img = cv2.imread('panda.png')

img_suavizada = cv2.GaussianBlur(img, (5, 5), 0)

cv2_imshow(img)
cv2_imshow(img_suavizada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Para a redução de ruído foi utilizada a biblioteca OpenCV e NumPy, Após realizar a importação desses recursos é necessário dispor de funções para realizar o pré-processamento da imagem. A função **cv2.imread()** lê a imagem que é armazenada na variável **img**. Já a função **cv2.GaussianBlur()** recorre ao filtro Gaussiano, aplicando na imagem para reduzir o ruído. Para visualizar os dados apresentados foi utilizada a função **cv2.imshow()**.