

YOLOv8

# Introdução à Detecção de Objetos com YOLO

A detecção de objetos é uma tarefa essencial em visão computacional, que envolve identificar e localizar objetos dentro de uma imagem. Um dos modelos mais populares para essa tarefa é o YOLO (You Only Look Once). Neste capítulo, vamos explorar como o YOLO realiza a análise de imagens, acompanhada de exemplos de código práticos.

## Como o YOLO Funciona?

O YOLO divide a imagem em uma grade e, para cada célula da grade, ele prevê um conjunto de caixas delimitadoras e as probabilidades de classe para esses boxes. Em vez de escanear a imagem várias vezes, o YOLO faz tudo em uma única passagem, tornando-o extremamente rápido e eficiente.

# Passo a Passo da Análise de Imagem com YOLO

## 1. Divisão da Imagem em Grades

- A imagem é dividida em uma grade de  $S \times S$ .

## 2. Predição das Caixas Delimitadoras

- Para cada célula da grade, o modelo prevê  $B$  caixas delimitadoras e uma pontuação de confiança para cada caixa.

## 3. Predição das Classes

- Para cada caixa delimitadora, o modelo prevê as probabilidades de pertencer a cada uma das classes.

# Exemplo de Código: Detectando Objetos com YOLO

Configurando o ambiente e carregando uma imagem para detecção

```
# Importando as bibliotecas necessárias
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Carregando a imagem
image = cv2.imread('caminho/para/sua/imagem.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Exibindo a imagem
plt.imshow(image)
plt.axis('off')
plt.show()
```

# Exemplo de Código: Detectando Objetos com YOLO

## Carregando o Modelo YOLO

```
# Carregando o modelo YOLO pré-treinado e as classes
net = cv2.dnn.readNet('caminho/para/yolov3.weights', 'caminho/para/yolov3.cfg')
with open('caminho/para/coco.names', 'r') as f:
    classes = [line.strip() for line in f.readlines()]

# Obter as camadas de saída do modelo YOLO
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
```

# Exemplo de Código: Detectando Objetos com YOLO

## Preparando a Imagem para a Detecção

```
# Redimensionar a imagem para o tamanho esperado pelo modelo YOLO
blob = cv2.dnn.blobFromImage(image, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
```

# Exemplo de Código: Detectando Objetos com YOLO

## Realizando a Detecção

```
# Realizar a detecção de objetos
outs = net.forward(output_layers)

# Processar as saídas
class_ids = []
confidences = []
boxes = []
height, width = image.shape[:2]
```

```
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)
```

# Exemplo de Código: Detectando Objetos com YOLO

## Desenhando as Caixas Delimitadoras na Imagem

```
# Aplicar Non-Maximum Suppression para eliminar caixas redundantes
indices = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Desenhar as caixas delimitadoras na imagem
for i in indices:
    i = i[0]
    box = boxes[i]
    x, y, w, h = box[0], box[1], box[2], box[3]
    label = str(classes[class_ids[i]])
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    cv2.putText(image, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

# Exibir a imagem com as detecções
plt.imshow(image)
plt.axis('off')
plt.show()
```



# Conclusão

Neste capítulo, abordamos os fundamentos da detecção de objetos utilizando YOLO, explicando o processo passo a passo e fornecendo exemplos de código em Python. Com isso, você está pronto para aplicar o YOLO em suas próprias imagens e projetos, explorando o poder da visão computacional em diversas aplicações.

Curtiu esse conteúdo? Ele foi gerado por inteligência artificial, mas foi revisado por alguém 100% Humano, e se quiser se conectar comigo, me siga no LinkedIn

Ferramentas de produção:

- Editor de imagem: Apresentações Google
- Conteúdo gerado por: ChatGPT
- Revisões Humanas: Duda Duarte