

RODRIGO ARAÚJO VIANNINI

POSTECH

IA PARA DEVS

COMPUTER VISION

AULA 01

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	7
SAIBA MAIS.....	15
O QUE VOCÊ VIU NESTA AULA?	18
REFERÊNCIAS.....	19

EMSE

O QUE VEM POR AÍ?

Fala, Devs! Neste módulo aprenderemos os conceitos básicos de visão computacional, campo da inteligência artificial que capacita os computadores a interpretarem e compreenderem o mundo visual, tal qual os seres humanos. Este campo envolve a aquisição, processamento e análise de imagens e vídeos digitais para extrair informações significativas.

Vejamos algumas aplicações:

- **Segurança:** sistemas de vigilância e reconhecimento facial.
- **Automação:** robôs e veículos autônomos.
- **Medicina:** diagnósticos assistidos por imagem e análise de exames médicos.
- **Entretenimento:** realidade aumentada e jogos.

História da visão computacional

- **1963:** Larry Roberts, considerado um dos fundadores da visão computacional, escreveu sua tese de doutorado no MIT sobre o uso de informações bidimensionais em uma cena tridimensional, o que foi uma das primeiras tentativas de usar computadores para interpretar imagens.
- **1966:** o “Projeto de Verão de Visão do MIT”, liderado por Seymour Papert, foi uma das primeiras iniciativas de pesquisa significativa na área, buscando desenvolver algoritmos que pudessem compreender cenas visuais.
- **1971:** David Marr, um neurocientista britânico, propôs um novo modelo de visão, sugerindo que o processamento visual deve ser entendido como um sistema de estágios. Seus trabalhos nos anos subsequentes foram fundamentais para a teoria da visão computacional.
- **1973:** o desenvolvimento do algoritmo de detecção de bordas de John Canny, conhecido como “Detector de Bordas de Canny”, foi um avanço importante no reconhecimento de padrões em imagens.

- **1981:** David Marr publicou seu livro "Vision", onde apresentou uma teoria completa do processamento visual, que incluía os conceitos de representações de imagem primária, secundária e terciária.
- **1988:** a introdução do conceito de redes neurais artificiais por Yann LeCun e outros, que mais tarde evoluiria para redes neurais convolucionais (CNNs), abriu novas possibilidades para a visão computacional.
- **1990:** a introdução do modelo "Active Shape" por Cootes, Taylor e outros permitiu o reconhecimento de formas baseadas em modelos flexíveis.
- **1993:** a invenção da "Transformada de Hough Generalizada" permitiu a detecção de formas mais complexas nas imagens, aprimorando a precisão dos algoritmos de visão computacional.
- **2001:** Viola-Jones introduziu um método eficiente para detecção de objetos em tempo real, especialmente rostos, utilizando características de Haar e uma estrutura de cascata.
- **2006:** Geoffrey Hinton e seus colaboradores revitalizaram o interesse em redes neurais profundas, introduzindo o conceito de redes neurais profundas (Deep Learning).
- **2012:** a vitória do AlexNet no ImageNet Large Scale Visual Recognition Challenge (ILSVRC) demonstrou o poder das redes neurais convolucionais profundas, revolucionando a visão computacional.
- **2014:** a introdução das Redes Gerativas Adversárias (GANs) por Ian Goodfellow permitiu a criação de imagens realistas a partir de descrições textuais ou outras imagens.
- **2015:** o lançamento do YOLO (You Only Look Once) por Joseph Redmon e outros, permitiu a detecção de objetos em tempo real com alta precisão.
- **2020:** o desenvolvimento de modelos como o "Vision Transformer" (ViT), demonstrou que técnicas baseadas em transformers, anteriormente utilizadas principalmente em processamento de linguagem natural, poderiam ser eficazes em tarefas de visão computacional.

- **2023:** avanços contínuos em IA explicável (Explainable AI - XAI) começaram a tornar as decisões de modelos de visão computacional mais transparentes e compreensíveis, facilitando a adoção em áreas críticas como medicina e segurança.

O que podemos esperar do futuro...

- **Visão 3D e reconstrução de cena:** melhoria na precisão e eficiência de técnicas de reconstrução 3D a partir de imagens 2D.
- **Visão computacional e robótica:** integração mais profunda entre visão computacional e robótica para navegação autônoma e interação com ambientes complexos.
- **Aplicações em saúde:** uso aprimorado de técnicas de visão computacional em diagnósticos médicos, análise de imagens radiológicas e monitoramento de pacientes.

Evolução da tecnologia

A evolução tecnológica em termos de hardware, como GPUs e softwares (como bibliotecas de aprendizado profundo), permitiu avanços rápidos na precisão e velocidade dos sistemas de visão computacional.

Componentes de um sistema de visão computacional

- **Aquisição de imagem:** envolve a captura de imagens ou vídeos utilizando dispositivos como câmeras digitais, scanners ou sensores.
- **Processamento de imagem:** refere-se às técnicas aplicadas para melhorar a qualidade das imagens e extrair informações importantes. Isso pode incluir operações como filtragem, transformação e segmentação.
- **Interpretação de imagem:** é o passo final, onde os dados processados são analisados para identificar e classificar objetos, reconhecer padrões e tomar decisões baseadas na análise visual.

Fundamentos técnicos

- Conceitos básicos de imagem digital:
 - **Pixels:** as menores unidades de uma imagem digital.
 - **Resolução:** o número de pixels em uma imagem é o que determina a nitidez.
 - **Cores:** imagens podem ser representadas em diferentes espaços de cor, como RGB (Red, Green, Blue) ou grayscale (tons de cinza).
- Representação de imagens
 - **Matrizes:** imagens são representadas como matrizes de pixels.
 - **Canais RGB:** imagens coloridas são compostas por três canais (vermelho, verde e azul).
 - **Grayscale:** imagens em tons de cinza têm apenas um canal.
- Ferramentas e bibliotecas comuns
 - **OpenCV:** é uma biblioteca de visão computacional open-source amplamente utilizada para processamento de imagens e vídeos.
 - **PIL/Pillow:** é uma biblioteca Python para abrir, manipular e salvar diferentes formatos de imagem.
 - **scikit-image:** é uma coleção de algoritmos para processamento de imagem em Python, baseada no scikit-learn.
- Tópicos adicionais
 - **Deep learning em visão computacional:** é a exploração do uso de redes neurais profundas, como CNNs, para tarefas avançadas de visão computacional, incluindo reconhecimento de objetos e segmentação semântica.
 - **Realidade aumentada:** são aplicações de visão computacional em realidade aumentada, onde informações digitais são sobrepostas no mundo real, utilizando algoritmos de rastreamento e detecção de objetos.

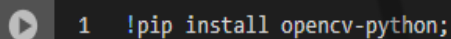
HANDS ON

Agora focaremos na implementação prática dos conceitos básicos de visão computacional discutidos anteriormente. A seguir, detalharemos como configurar o ambiente e os passos necessários para realizar operações básicas de processamento de imagens (carregamento, exibição, conversão para grayscale, redimensionamento, aplicação de filtros, detecção de bordas e desenho de formas) utilizando a biblioteca OpenCV. Essas operações são fundamentais para o desenvolvimento de aplicações de visão computacional mais complexas.

- Configuração do ambiente

Instalação das bibliotecas a serem utilizadas

Para começar, vamos instalar a biblioteca OpenCV. Abra o terminal ou prompt de comando e execute o seguinte comando:

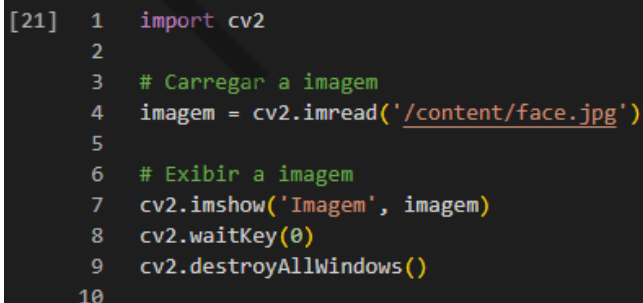


```
1 !pip install opencv-python;
```

Figura 1 – Prompt de comando para instalação da biblioteca OpenCV
Fonte: Elaborado pelo autor (2024)

- Carregamento e exibição de imagens

Vamos iniciar carregando uma imagem do disco e exibindo-a na tela usando IDEs (PyCharm ou VsCode, por exemplo):



```
[21] 1 import cv2
      2
      3 # Carregar a imagem
      4 imagem = cv2.imread('/content/face.jpg')
      5
      6 # Exibir a imagem
      7 cv2.imshow('Imagem', imagem)
      8 cv2.waitKey(0)
      9 cv2.destroyAllWindows()
     10
```

Figura 2 – Carregamento e exibição de imagens usando IDEs
Fonte: Elaborado pelo autor (2024)

Usando o Google Colab

```

1  import cv2
2  import matplotlib.pyplot as plt
3
4  # Carregar a imagem
5  imagem = cv2.imread('/content/face.jpg')
6
7  # Converter a imagem de BGR para RGB
8  imagem_rgb = cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)
9
10 # Exibir a imagem
11 plt.imshow(imagem_rgb)
12 # plt.axis('off') # Desabilita os eixos
13 plt.show()
14

```

Figura 3 – Carregamento e exibição de imagens usando Google Colab
Fonte: Elaborado pelo autor (2024)

Output:

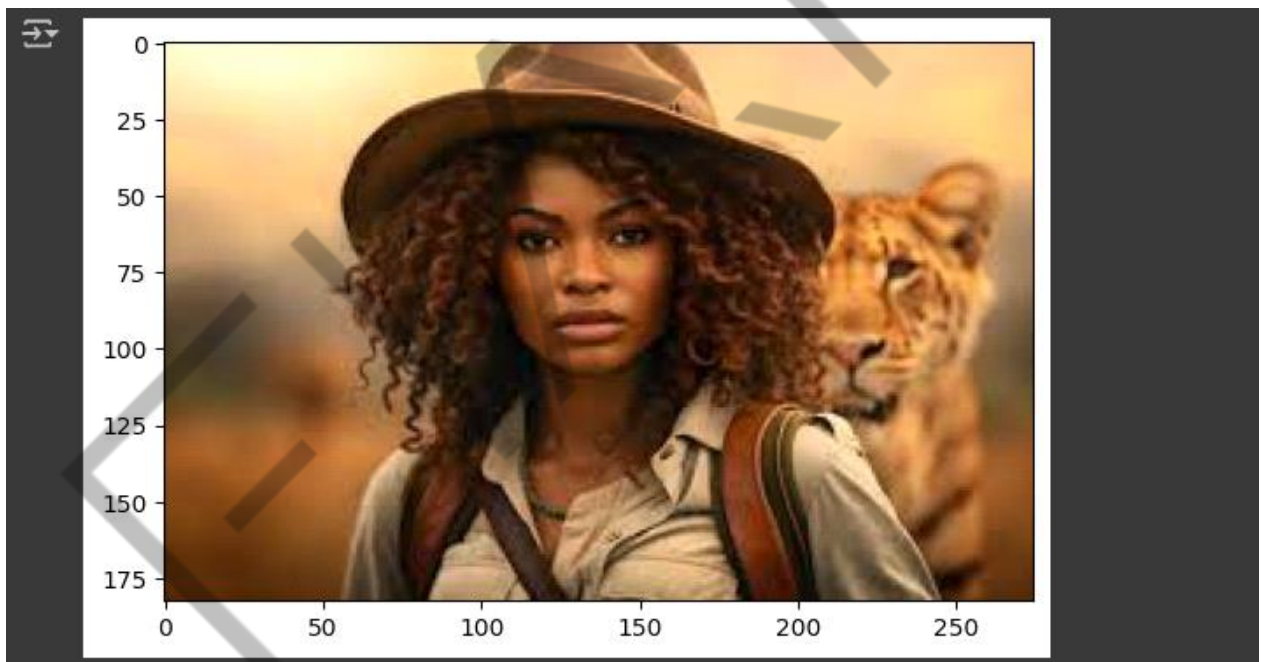


Figura 3 – Output
Fonte: Elaborado pelo autor (2024)

- Conversão de imagens para grayscale

Agora, vamos converter a imagem colorida para escala de cinza. Esta é uma operação comum em visão computacional. Isso é feito facilmente usando OpenCV.


```

1  import cv2
2  import matplotlib.pyplot as plt
3
4  # Carregar a imagem
5  imagem = cv2.imread('/content/face.jpg')
6
7  # Converter a imagem para grayscale
8  imagem_gray = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
9
10 # Exibir a imagem grayscale
11 plt.imshow(imagem_gray, cmap='gray')
12 plt.axis('off') # Desabilita os eixos
13 plt.show()
14

```

Figura 4 – Conversão da imagem usando OpenCV
Fonte: Elaborado pelo autor (2024)

Output:



Figura 5 – Output (2)
Fonte: Elaborado pelo autor (2024)

- Redimensionamento de imagem

Vamos redimensionar uma imagem para quando for necessário padronizar o tamanho das imagens de entrada.

Output:



Figura 6 – Output (3)
Fonte: Elaborado pelo autor (2024)

- Aplicação de filtros para suavização de imagens

Vamos aplicar a suavização de uma imagem, o que ajuda a reduzir o ruído e na melhora da detecção de bordas.

```

1  import cv2
2  import matplotlib.pyplot as plt
3
4  # Carregar a imagem
5  imagem = cv2.imread('/content/face.jpg')
6
7  # Aplicar suavização com filtro Gaussiano
8  imagem_suavizada = cv2.GaussianBlur(imagem, (15, 15), 0)
9
10 # Converter a imagem de BGR para RGB
11 imagem_suavizada_rgb = cv2.cvtColor(imagem_suavizada, cv2.COLOR_BGR2RGB)
12
13 # Exibir a imagem suavizada
14 plt.imshow(imagem_suavizada_rgb)
15 plt.axis('off') # Desabilita os eixos
16 plt.show()
17

```

Figura 7 – Aplicando filtros para suavização de imagens
Fonte: Elaborado pelo autor (2024)

Output:

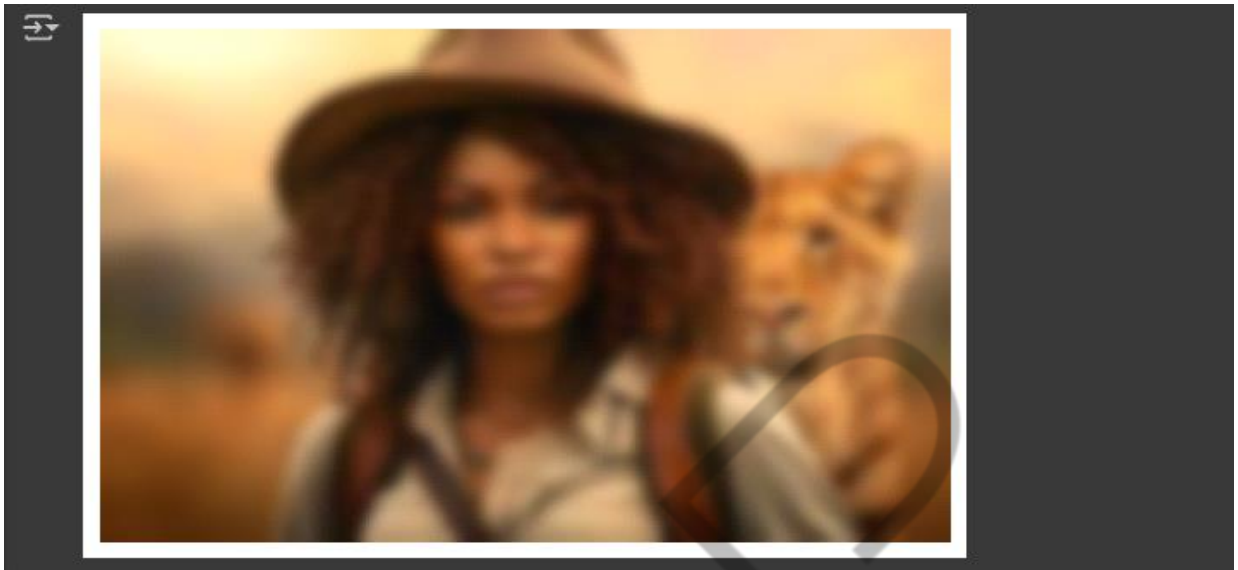


Figura 8 – Output (4)
Fonte: Elaborado pelo autor (2024)

Deteção de bordas com canny

Agora, vamos iniciar a detecção de bordas. Este é um passo crucial em muitos algoritmos de visão computacional, como é o caso do algoritmo de detecção de objetos.

```
1 import cv2
2 import matplotlib.pyplot as plt
3
4 # Carregar a imagem
5 imagem = cv2.imread('/content/face.jpg')
6
7 # Converter a imagem para grayscale
8 imagem_gray = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
9
10 # Detectar bordas na imagem
11 bordas = cv2.Canny(imagem_gray, 100, 200)
12
13 # Exibir a imagem com bordas detectadas
14 plt.imshow(bordas, cmap='gray')
15 plt.axis('off') # Desabilita os eixos
16 plt.show()
17
```

Figura 9 – Deteção de bordas
Fonte: Elaborado pelo autor (2024)

Output:



Figura 10 – Output (5)
Fonte: Elaborado pelo autor (2024)

Desenhar formas na imagem

Podemos desenhar formas como retângulos para destacar áreas de interesse na imagem.

```

1  import cv2
2  import matplotlib.pyplot as plt
3
4  imagem = cv2.imread('/content/face.jpg')
5
6  # Definir as coordenadas e a cor do retângulo
7  inicio = (50, 5)
8  fim = (220, 160)
9  cor = (255, 0, 0) # Azul em BGR
10  espessura = 2
11
12  # Desenhar o retângulo na cópia da imagem
13  imagem_com_retangulo = cv2.rectangle(imagem.copy(),
14  | | | | | | | | | inicio, fim, cor, espessura)
15
16  # Converter a imagem para RGB
17  imagem_com_retangulo_rgb = cv2.cvtColor(imagem_com_retangulo,
18  | | | | | | | | | cv2.COLOR_BGR2RGB)
19
20  # Exibir a imagem com retângulo
21  plt.imshow(imagem_com_retangulo_rgb)
22  plt.axis('off') # Desabilita os eixos
23  plt.show()

```

Figura 11 – Desenhando formas na imagem
Fonte: Elaborado pelo autor (2024)

Output:



Figura 12 – Output (6)
Fonte: Elaborado pelo autor (2024)

Salvando a imagem processada

Finalmente podemos salvar a imagem processada no disco:

```
1 import cv2
2
3 # Salvar a imagem processada
4 caminho_salvar = '/content/imagem_processada.jpg'
5 cv2.imwrite(caminho_salvar, imagem_com_retangulo)
6
```

Figura 13 – Salvando a imagem processada
Fonte: Elaborado pelo autor (2024)

Output:

```
True
```

Figura 14 – Output (7)
Fonte: Elaborado pelo autor (2024)

Output:

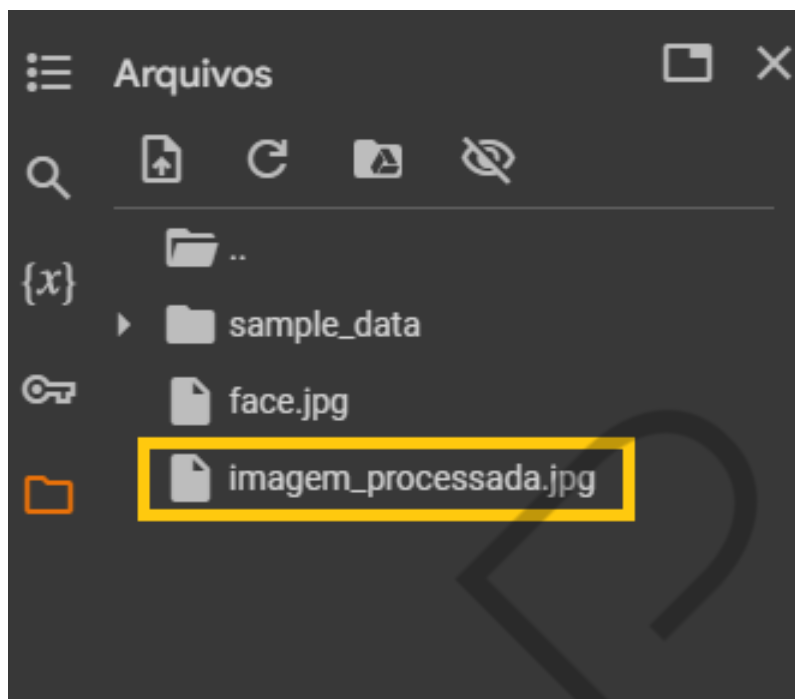


Figura 15 – Output (8)
Fonte: Elaborado pelo autor (2024)

SAIBA MAIS

À medida que nos aprofundamos na introdução à visão computacional, é importante entender que esse campo abrange uma variedade de técnicas e aplicações que vão muito além do processamento de imagens básico. Aqui estão algumas áreas de interesse adicionais que você pode explorar:

- **Reconhecimento de padrões:** o reconhecimento de padrões é uma área essencial da visão computacional, que se concentra na identificação de padrões complexos em dados visuais. Ele é fundamental para aplicações como reconhecimento facial, classificação de objetos e identificação de padrões em imagens médicas.
- **Segmentação de imagem:** a segmentação de imagem envolve a divisão de uma imagem em regiões distintas com base em características específicas, como cor, textura ou intensidade. Essa técnica é utilizada em aplicações como detecção de bordas, segmentação de órgãos em imagens médicas e segmentação de objetos em cenários de vigilância.
- **Visão estéreo e reconstrução 3D:** a visão estéreo envolve a utilização de duas ou mais imagens para estimar a estrutura tridimensional de uma cena. Isso é fundamental para aplicações como reconstrução 3D, mapeamento de ambientes e navegação autônoma.
- **Aprendizado profundo em visão computacional:** o aprendizado profundo, particularmente as redes neurais convolucionais (CNNs), revolucionou a visão computacional, permitindo avanços significativos em áreas como reconhecimento de imagem, detecção de objetos e segmentação semântica.
- **Realidade aumentada:** a realidade aumentada combina elementos do mundo real com elementos virtuais, geralmente por meio de dispositivos como smartphones, óculos inteligentes ou headsets de realidade aumentada. A visão computacional desempenha um papel fundamental na detecção e rastreamento de objetos do mundo real para sobrepor informações virtuais de forma precisa.

- **Aplicações em medicina e saúde:** na área da saúde, a visão computacional é usada para uma variedade de aplicações, incluindo diagnósticos médicos assistidos por computadores, cirurgias assistidas por computadores, monitoramento de pacientes e análise de imagens médicas.
- **Visão computacional embarcada:** com o avanço da miniaturização de dispositivos e o aumento da capacidade de processamento, a visão computacional está se tornando cada vez mais comum em dispositivos embarcados, como drones, robôs autônomos, câmeras de segurança e sistemas de assistência ao motorista.
- **Reconhecimento de texto em imagens (OCR):** o OCR é uma técnica que permite extrair texto de imagens ou documentos digitalizados. É amplamente utilizado em sistemas de digitalização de documentos, reconhecimento de placas de veículos, tradução de texto em tempo real e muito mais.
- **Análise de movimento e pose:** esta área se concentra em entender e interpretar o movimento humano em imagens ou vídeos. É útil em aplicações como controle de gestos, análise de movimento esportivo, vigilância baseada em comportamento e detecção de quedas para assistência aos idosos.
- **Segmentação semântica:** a segmentação semântica atribui rótulos a cada pixel em uma imagem, identificando diferentes objetos e regiões semânticas. Isso é essencial para aplicações como direção autônoma, mapeamento de ambientes internos e realidade virtual.
- **Visão computacional para agricultura:** na agricultura de precisão, a visão computacional é usada para monitorar o crescimento das culturas, detectar doenças nas plantas, realizar contagem de frutas, gerenciar a irrigação e muito mais, ajudando a aumentar a eficiência e a produtividade agrícola.
- **Identificação de emoções em faces:** esta área visa reconhecer e interpretar as emoções humanas a partir de expressões faciais em imagens ou vídeos. É utilizada em aplicações de interação humano-computador, análise de sentimentos em mídias sociais, detecção de emoções para personalização de experiências e muito mais.

- **Fusão de dados sensoriais:** a fusão de dados sensoriais combina informações de diferentes sensores, como câmeras, radar e ultrassom, para obter uma compreensão mais completa do ambiente ao redor. Isso é fundamental para aplicações como veículos autônomos, robótica móvel e sistemas de vigilância inteligente.
- **Deteção de anomalias e eventos raros:** esta área visa identificar padrões e eventos incomuns em dados visuais. É crucial em aplicações de segurança, tais como detecção de intrusos, reconhecimento de comportamentos suspeitos e monitoramento de eventos em tempo real.
- **Simulação e realidade virtual:** a visão computacional desempenha um papel fundamental na criação de ambientes virtuais realistas e interativos, utilizados em treinamento militar, simulação de voo, design de produtos, entretenimento digital e muito mais.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, conhecemos os fundamentos da Visão Computacional, explorando desde a sua definição até a sua evolução ao longo das décadas. A visão computacional é uma disciplina da inteligência artificial que permite aos computadores interpretar e compreenderem o mundo visual, de maneira muito semelhante dos seres humanos. Suas aplicações são vastas, indo desde sistemas de segurança e automação até diagnósticos médicos e entretenimento.

Além disso, exploramos a implementação prática desses conceitos, desde a configuração do ambiente de desenvolvimento até operações básicas de processamento de imagens, utilizando bibliotecas como OpenCV e scikit-image. Essas operações são fundamentais para o desenvolvimento de aplicações mais complexas neste campo que está em constante evolução.

Ao final, destacamos áreas adicionais de interesse, como reconhecimento de padrões, segmentação de imagem e realidade aumentada, e que demonstram o vasto potencial e as oportunidades de pesquisa e desenvolvimento na Visão Computacional.

REFERÊNCIAS

BRUNELLI, R. **Template Matching Techniques in Computer Vision: Theory and Practice**. Nova Jersey: Wiley, 2009.

NIXON, M. S.; AGUADO, A. S. **Feature Extraction & Image Processing for Computer Vision**. Massachusetts: Academic Press, 2012.

SHAPIRO, L. G.; STOCKMAN, G. C. **Computer Vision**. Nova Jersey: Prentice Hall, 2001.

EMAP

PALAVRAS-CHAVE

Visão Computacional. Algoritmos. Reconhecimento de Padrões. Processamento de Imagem.

ENAMP



POSTECH