



## Aula 2

# Modelos de Deep Learning para Visão Computacional



# Conteúdo da Aula

1. Introdução a Conceitos Básicos
2. Tipos de Modelos de Visão Computacional
3. Tarefas Comuns em Visão Computacional
4. Demonstração Prática
5. Definição da Tarefa Individual

## 💡 Introdução a Conceitos Básicos

- **Visão Computacional:** Campo da inteligência artificial que ensina computadores a ver e compreender imagens. Utilizando imagens digitais e modelos de deep learning, máquinas podem identificar, classificar e responder a objetos com alta precisão.

# 💡 Introdução a Conceitos Básicos

- **Visão Computacional:** Campo da inteligência artificial que ensina computadores a ver e compreender imagens. Utilizando imagens digitais e modelos de deep learning, máquinas podem identificar, classificar e responder a objetos com alta precisão.
- **Deep Learning:** Tipo de aprendizado de máquina que utiliza redes neurais artificiais com muitas camadas. Em visão computacional, permite que modelos aprendam padrões diretamente dos pixels da imagem, evoluindo de formas simples até objetos complexos.

# 💡 Introdução a Conceitos Básicos

- **Visão Computacional:** Campo da inteligência artificial que ensina computadores a ver e compreender imagens. Utilizando imagens digitais e modelos de deep learning, máquinas podem **identificar, classificar e responder a objetos com alta precisão.**
- **Deep Learning:** Tipo de aprendizado de máquina que utiliza redes neurais artificiais com muitas camadas. Em visão computacional, permite que modelos aprendam padrões diretamente dos pixels da imagem, **evoluindo de formas simples até objetos complexos.**
- **Redes Neurais Convolucionais (CNNs):** Principal método de deep learning para análise de imagens. CNNs utilizam camadas específicas para **aprender automaticamente padrões visuais em múltiplos níveis**, de bordas básicas até estruturas detalhadas.

## Tipos de Modelos de Visão Computacional

- ResNet: Usa conexões residuais para permitir o treinamento de redes muito profundas.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoatenção.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoatenção.
- **Redes Siamesas:** Redes gêmeas que aprendem métricas de similaridade entre pares de entrada.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoattenção.
- **Redes Siamesas:** Redes gêmeas que aprendem métricas de similaridade entre pares de entrada.
- **Autoencoders:** Arquiteturas encoder-decoder para representações comprimidas e tarefas generativas.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoattenção.
- **Redes Siamesas:** Redes gêmeas que aprendem métricas de similaridade entre pares de entrada.
- **Autoencoders:** Arquiteturas encoder-decoder para representações comprimidas e tarefas generativas.
- **U-Net:** Encoder-decoder com conexões de atalho, eficaz para **segmentação semântica e imagens médicas**.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoattenção.
- **Redes Siamesas:** Redes gêmeas que aprendem métricas de similaridade entre pares de entrada.
- **Autoencoders:** Arquiteturas encoder-decoder para representações comprimidas e tarefas generativas.
- **U-Net:** Encoder-decoder com conexões de atalho, eficaz para **segmentação semântica e imagens médicas**.
- **GANs:** Duas redes (gerador e discriminador) que competem, produzindo imagens cada vez mais realistas.

## Tipos de Modelos de Visão Computacional

- **ResNet:** Usa conexões residuais para permitir o treinamento de redes muito profundas.
- **Vision Transformer (ViT):** Processa imagens como sequências de patches usando blocos transformer e autoattenção.
- **Redes Siamesas:** Redes gêmeas que aprendem métricas de similaridade entre pares de entrada.
- **Autoencoders:** Arquiteturas encoder-decoder para representações comprimidas e tarefas generativas.
- **U-Net:** Encoder-decoder com conexões de atalho, eficaz para **segmentação semântica e imagens médicas**.
- **GANs:** Duas redes (gerador e discriminador) que competem, produzindo imagens cada vez mais realistas.
- **Modelos de Difusão:** Modelos gerativos que criam imagens refinando iterativamente um ruído inicial.

## ■ Tarefas Comuns em Visão Computacional - Classificação de Imagens

### Classificação de Imagens

- Atribui um único rótulo para a imagem inteira.
- Pergunta: O que há nesta imagem?



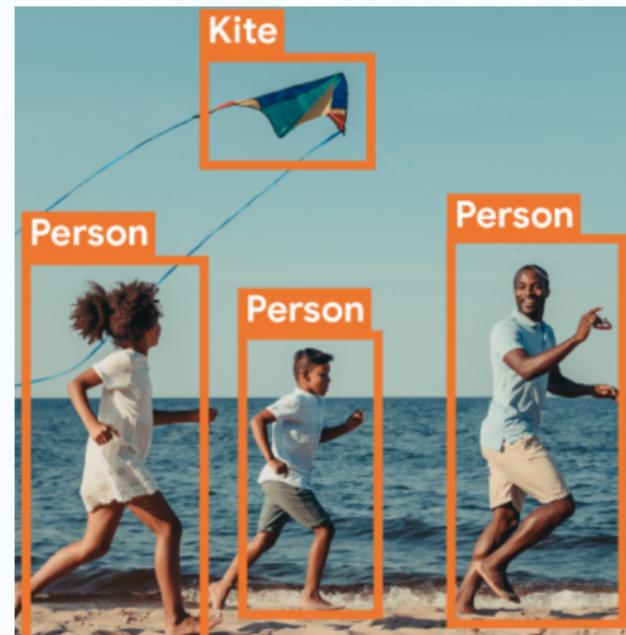
Classification

Cat

# Tarefas Comuns em Visão Computacional - Detecção de Objetos

## Detecção de Objetos

- Identifica e localiza objetos em uma imagem desenhando **caixas delimitadoras**.
- **Pergunta:** Quais objetos estão na imagem e onde estão?



# ■ Tarefas Comuns em Visão Computacional - Segmentação Semântica

## Segmentação Semântica

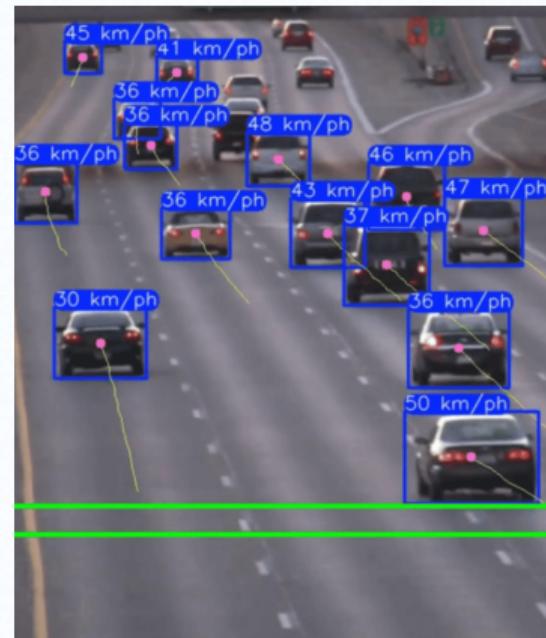
- Classifica cada pixel em uma categoria, sem distinguir instâncias diferentes da mesma classe.
- **Pergunta:** A qual categoria pertence cada pixel (carro, estrada, céu, etc.)?



## ■ Tarefas Comuns em Visão Computacional - Rastreamento de Objetos

### Rastreamento de Objetos

- Acompanha o movimento de objetos em múltiplos quadros de vídeo.
- **Pergunta:** Onde está este objeto no próximo quadro?



# 💬 Tarefas Comuns em Visão Computacional - Geração de Legendas

## Captioning de Imagens

- Gera descrições em linguagem natural do conteúdo de uma imagem.
- Combina visão computacional com NLP.
- **Pergunta:** O que está acontecendo nesta imagem?

'The image is a close-up of a Kodak VR35 digital camera. The camera is black in color and has the Kodak logo on the top left corner. The body of the camera is made of wood and has a textured grip for easy handling. The lens is in the center of the body and is surrounded by a gold-colored ring. On the top right corner, there is a small LCD screen and a flash. The background is blurred, but it appears to be a wooded area with trees and greenery.'



# 为人 Tarefas Comuns em Visão Computacional - Estimativa de Esqueleto

## Estimativa de Esqueleto

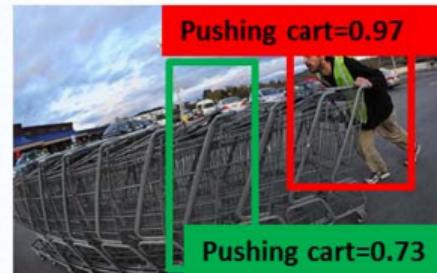
- Detecta e estima a pose humana e a posição das articulações.
- Mapeia pontos anatômicos chave no corpo humano.
- **Pergunta:** Onde estão as articulações e como o corpo está posicionado?



# 🏃 Tarefas Comuns em Visão Computacional - Detecção de Ações

## Detecção de Ações

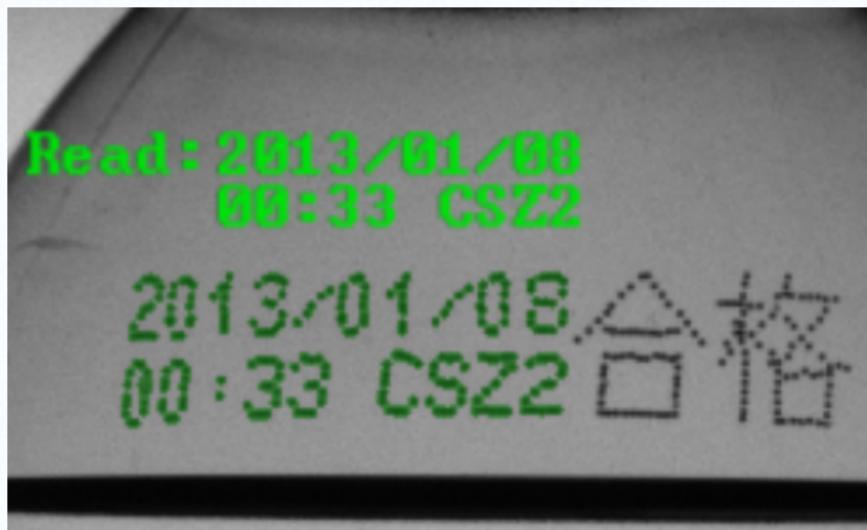
- Identifica e localiza ações humanas em vídeos.
- Combina informações espaciais e temporais.
- Pergunta: Que ação está sendo realizada e onde?



# Tarefas Comuns em Visão Computacional - OCR

## Reconhecimento Óptico de Caracteres (OCR)

- Converte imagens de texto em formato legível por máquina.
- **Extrai textos de documentos, placas e imagens.**
- **Pergunta:** Qual texto está escrito nesta imagem?



## ▶ Demonstração Prática

**Objetivo:** Usar o YOLO (Ultralytics) para detectar objetos em uma foto.

■ **Passo 1: Instalação:** Instalar o pacote Ultralytics YOLO com `pip install ultralytics`.

## ▶ Demonstração Prática

**Objetivo:** Usar o YOLO (Ultralytics) para detectar objetos em uma foto.

- **Passo 1: Instalação:** Instalar o pacote Ultralytics YOLO com `pip install ultralytics`.
- **Passo 2: Carregar Modelo:** Carregar um modelo pré-treinado YOLOv8:
  - ▶ Importar YOLO de ultralytics
  - ▶ Carregar modelo: `model = YOLO('yolov8n.pt')`
  - ▶ Escolher tamanho (n, s, m, l, x) conforme **velocidade vs precisão**

## ▶ Demonstração Prática

**Objetivo:** Usar o YOLO (Ultralytics) para detectar objetos em uma foto.

- **Passo 1: Instalação:** Instalar o pacote Ultralytics YOLO com `pip install ultralytics`.
- **Passo 2: Carregar Modelo:** Carregar um modelo pré-treinado YOLOv8:
  - ▶ Importar YOLO de ultralytics
  - ▶ Carregar modelo: `model = YOLO('yolov8n.pt')`
  - ▶ Escolher tamanho (n, s, m, l, x) conforme **velocidade vs precisão**
- **Passo 3: Detecção de Objetos:** Rodar inferência em uma imagem usando `model.predict()`.

## ▶ Demonstração Prática

**Objetivo:** Usar o YOLO (Ultralytics) para detectar objetos em uma foto.

- **Passo 1: Instalação:** Instalar o pacote Ultralytics YOLO com `pip install ultralytics`.
- **Passo 2: Carregar Modelo:** Carregar um modelo pré-treinado YOLOv8:
  - ▶ Importar YOLO de ultralytics
  - ▶ Carregar modelo: `model = YOLO('yolov8n.pt')`
  - ▶ Escolher tamanho (n, s, m, l, x) conforme **velocidade vs precisão**
- **Passo 3: Detecção de Objetos:** Rodar inferência em uma imagem usando `model.predict()`.
- **Passo 4: Análise dos Resultados:** Visualizar objetos detectados com caixas, scores de confiança e rótulos.

## 👤 Definição da Tarefa Individual

- **Segmentação Semântica com SAM 2.1:** Implementar um pipeline de segmentação usando o **Segment Anything Model 2.1**. Aplicar o modelo para segmentar objetos em imagens ou vídeos, gerando máscaras em nível de pixel. Testar diferentes formas de prompting (pontos, caixas ou geração automática) e documentar **qualidade, tempo de processamento e uso de memória** em comparação a abordagens tradicionais.