[slide\_01] [long\_pause]

Aula 3: Uso de Modelos de Visão Computacional em Pipelines Reais.

Hoje será abordado o uso de modelos de visão computacional em pipelines do mundo real. Vamos caminhar desde os conceitos fundamentais até a implementação prática de um sistema completo, apresentando também cuidados importantes que precisam ser tomados para colocar uma solução em produção.

A ideia é que, ao final, vocês tenham não só o entendimento teórico, mas também uma visão clara dos desafios e das decisões que envolvem cada etapa de um pipeline de visão computacional.

[slide\_02] [short\_pause]

O conteúdo está organizado em cinco partes.

Primeiro, teremos uma introdução aos conceitos básicos, onde revisaremos rapidamente a evolução da visão computacional e o papel das redes neurais.

[slide\_03] [short\_pause]

Em seguida veremos como utilizar um modelo de visão computacional, desde a definição do problema até o pós-processamento dos resultados.

[slide\_04] [short\_pause]

Depois, vamos discutir os principais pontos de atenção para colocar um modelo em produção, como desempenho, processamento dos dados e limitações de hardware.

[slide\_05] [short\_pause]

Após isso, teremos também um fluxo prático, em que vou detalhar um exemplo real de contagem e classificação de veículos.

[slide\_06]

E, por fim, será proposta uma tarefa individual para que vocês possam aplicar o que aprenderam.

[slide\_07] [short\_pause]

Introdução a Conceitos Básicos

Quando falamos em modelos de visão computacional, pensamos em sistemas que tentam imitar a nossa capacidade de interpretar o que vemos. Esses modelos vão desde redes convolucionais clássicas até arquiteturas mais recentes, como transformadores visuais.

[slide\_08] [short\_pause]

A inferência é o momento em que o modelo, já treinado, é executado com o objetivo de analisar dados que ele nunca viu antes, algo fundamental quando queremos trabalhar em ambientes dinâmicos.

[slide\_09] [long\_pause]

O pipeline é toda a sequência de módulos que conecta a captura das imagens, o pré-processamento, a inferência e as ações que dependem do resultado. No mundo real, é essa integração que garante que uma solução não seja apenas um experimento acadêmico, mas um sistema que funciona de forma contínua e confiável.

[slide\_10] [short\_pause]

Como Utilizar um Modelo de Visão Computacional

Para que um modelo de visão realmente funcione em produção, precisamos pensar em um fluxo de trabalho estruturado. Tudo começa com a formulação do problema: é classificação de imagens? Detecção de objetos? Segmentação? Essa definição orienta a escolha do modelo.

[slide\_11] [short\_pause]

Em seguida, tratamos das imagens. Normalizar, redimensionar, garantir que o formato e a qualidade sejam consistentes é essencial para evitar resultados instáveis.

[slide\_12] [short\_pause]

Depois vem a inferência, em que os dados pré-processados passam pelo modelo e geram saídas brutas, como probabilidades ou coordenadas.

[slide\_13] [long\_pause]

Mas não para por aí: o pós-processamento transforma essas saídas em informações ÚTEIS, aplicando técnicas como supressão não máxima para eliminar detecções duplicadas ou de baixa confiança. Cada uma dessas etapas tem suas próprias armadilhas, que precisam ser monitoradas quando o sistema já estiver rodando.

[slide\_14] [short\_pause]

Requisitos Importantes para a Implantação da Solução

Na prática, colocar um modelo no ar exige equilíbrio entre desempenho e precisão. Aplicações em tempo real, como monitoramento de trânsito ou sistemas de segurança, exigem baixa latência. Isso PODE SIGNIFICAR QUE O SISTEMA DEVE abrir mão de um pouco de acurácia para ganhar velocidade.

[slide\_15] [short\_pause]

Outro ponto é a deriva de dados: o ambiente muda, a iluminação muda, novos objetos aparecem. Sem reavaliações periódicas e re-treinamentos, o modelo perde qualidade.

[slide\_16] [short\_pause]

Além disso, a escolha do hardware é crítica. Processar vídeo em tempo real com CPU costuma ser inviável, por isso recorremos a GPUs, TPUs ou dispositivos dedicados.

[slide\_17] [long\_pause]

E não podemos esquecer os casos extremos: condições climáticas adversas, objetos parcialmente ocultos ou movimento intenso. Testes em cenários variados ajudam a descobrir limitações antes que o sistema seja usado em situações críticas.

[slide\_18] [short\_pause]

Demonstração Prática

Agora vamos ver tudo isso funcionando em um exemplo. A proposta é criar um script em Python para contar e classificar veículos em tempo real. Usaremos a biblioteca OpenCV para lidar com os vídeos das câmeras de tráfego.

[slide\_19] [short\_pause]

Para a detecção, iremos aproveitar um modelo YOLOv8 pré-treinado, que já oferece ótimo desempenho em tarefas de detecção de objetos.

[slide\_20] [short\_pause]

Iremos integrar um algoritmo de rastreamento, como DeepSORT ou ByteTrack, para acompanhar cada veículo ao longo dos quadros e garantir que a contagem seja de fato de veículos únicos, não de detecções repetidas.

[slide\_21]

Por fim, iremos apresentar estatísticas em tempo real, como velocidade média estimada e número de veículos por categoria, tudo anotado diretamente no vídeo.

[slide\_22]

Definição da Tarefa Individual

Para fixar os conceitos, propomos que cada participante desenvolva um contador de PESSOAS em tempo real. Vocês vão construir um pipeline que detecta a classe ‘person’ em uma transmissão ao vivo e exibe a contagem atual de pessoas em cada quadro. Esse desafio vai fazer vocês lidarem com problemas práticos, como ajustar limiares de confiança, gerenciar recursos de hardware e tratar casos de oclusão. Mesmo sendo um projeto relativamente simples, ele reúne todos os elementos centrais de um pipeline de visão computacional. A ideia é que, ao final, vocês possam adaptar o mesmo raciocínio para problemas mais complexos, como vigilância de ambientes, monitoramento de filas ou análise de comportamento em tempo real.