

CENÁRIO INDUSTRIAL FICTÍCIO – Edge AI (Visão Computacional Embarcada)

Contexto Geral

A empresa MetalForming Brasil S.A., líder nacional no setor de manufatura de peças automotivas, está modernizando sua linha de inspeção de segurança em um setor crítico: a área de prensas industriais.

Essas prensas operam centenas de ciclos por hora e são classificadas como máquinas de alto risco, com histórico recente de incidentes envolvendo:

- operadores entrando na zona de risco durante ciclos ativos,
- ferramentas esquecidas dentro da área da prensa,
- objetos indevidos próximos ao equipamento,
- comportamentos inseguros não detectados a tempo.

Para prevenir acidentes graves e cumprir normas de segurança industrial, a empresa está implantando um sistema Edge AI embarcado, capaz de analisar imagens em tempo real diretamente de uma Raspberry Pi instalada ao lado de cada prensa.

O sistema será responsável por identificar, com base em modelos pré-treinados, instâncias de objetos presentes nas imagens capturadas pela câmera da máquina — sem necessidade de treinar novos modelos ou classes customizadas.

A sua tarefa é desenvolver o módulo embarcado de visão computacional que será executado dentro da Raspberry Pi.

OBJETIVO DO TESTE

Desenvolver um serviço de inferência de visão computacional embarcado, rodando obrigatoriamente dentro de um container na Raspberry Pi, capaz de:

1. Receber uma URL apontando para uma imagem capturada pela câmera da prensa.
2. Rodar inferência utilizando modelo pré-treinado.
3. Detectar objetos de classes já existentes no modelo pré-treinado, tais como: person, scissors, knife, bottle, cell phone, backpack, etc.
4. Atender a dois endpoints específicos, definidos a seguir.

REQUISITOS FUNCIONAIS

Sua aplicação deve expor dois endpoints HTTP, ambos executados dentro da Raspberry Pi via um container.

1. Endpoint /detect/json

O serviço deve:

- ☐ Executar a inferência,
- ☐ Filtrar predições com confiança mínima fornecida,
- ☐ Retornar, em JSON:
 - ☐ lista completa das detecções (classe, confiança, bbox),
 - ☐ classes detectadas,
 - ☐ somatório de instâncias por classe,
 - ☐ metadados (modelo, tempo de inferência, imagem recebida).

2. Endpoint /detect/image

O serviço deve:

- ☐ Realizar a inferência,
- ☐ Filtrar predições com confiança mínima fornecida,
- ☐ Gerar uma imagem anotada com bounding boxes e labels.
- ☐ Retornar o resultado como PNG,

REQUISITOS DE CONTAINERIZAÇÃO

A aplicação deve rodar exclusivamente dentro de um container.

O container deve:

- Ser iniciado automaticamente no boot da Raspberry Pi
- Ser configurado com política de restart (ex.: restart: always)
- Expor a porta HTTP corretamente

DICA (opcional)

- fastapi – para criação da API
- uvicorn – para servir a aplicação
- ultralytics – para inferência com YOLO pré-treinado
- opencv-python-headless – para manipulação de imagens e anotação
- numpy – suporte matemático básico