

Flask Application with YOLOv8 Object Detection

Este aplicativo integra Flask e Socket.IO para detecção de objetos em tempo real utilizando o modelo YOLOv8n, processa quadros de vídeo recebidos via WebSocket, aplica detecção de objetos e retorna os quadros anotados para o cliente.

Módulos Utilizados

- Flask: Framework web para Python.
- Flask-SocketIO: Comunicação em tempo real utilizando WebSockets.
- OpenCV (cv2): Processamento de imagens e visão computacional.
- NumPy: Manipulação de arrays.
- YOLO: Modelo de detecção de objetos (YOLOv8n).
- dotenv: Gerenciamento de variáveis de ambiente.

Arquivos Necessários

1. Modelo YOLOv8 (`yolov8n.pt`) localizado no diretório `./model/`.
2. Templates HTML: `index.html`, `deteccao.html` e `sobre.html`.
3. Arquivo `.env` contendo a variável `FLASK_SECRET_KEY`.

Rotas Disponíveis

- `/`: Renderiza a página inicial.
- `/deteccao.html`: Renderiza a página de detecção.
- `/sobre.html`: Renderiza a página "Sobre".
- `/detections`: Retorna os dados mais recentes das detecções em formato JSON.

Funções

Função: handle_frame

Descrição: Processa quadros de vídeo recebidos via WebSocket, aplica detecção de objetos usando o YOLOv8 e retorna o quadro processado ao cliente.

Argumentos:

- `data` (str): String codificada em base64 contendo o quadro de vídeo.

Emite:

- `processed_frame` (str): Quadro processado codificado em base64.

Função: get_detections

Descrição: Endpoint para buscar as detecções mais recentes.

Retorno:

- JSON: Dados das detecções mais recentes.

Função: get_deteccao

Descrição: Renderiza a página HTML de detecção.

Retorno:

- HTML: Página de detecção.

Função: get_sobre

Descrição: Renderiza a página HTML "Sobre".

Retorno:

- HTML: Página "Sobre".

Função: index

Descrição: Renderiza a página inicial do aplicativo.

Retorno:

- HTML: Página inicial.

Configuração

- Certifique-se de ter as dependências instaladas:

```
pip install flask flask-socketio opencv-python numpy ultralytics python-dotenv
```

- Insira a variável `FLASK_SECRET_KEY` no arquivo `.env`.

Execução

- Execute o aplicativo:

```
python app.py
```

- O servidor estará disponível em `http://0.0.0.0:5000`.

Documentação da Página HTML para Detecção em Tempo Real com YOLOv8n

Esta página HTML é a interface do usuário para o sistema de detecção em tempo real utilizando YOLOv8n. São combinadas funcionalidades de captura de vídeo da webcam, envio de frames para um servidor de processamento, e exibição dos frames processados com detecções anotadas.

Estrutura Geral

1. Cabeçalho (Head):

- Define o título da página como "YOLOv8 Live Detection".
- Inclui um link para o arquivo de estilo CSS (`style-deteccao.css`) localizado na pasta estática.
- Carrega o script da biblioteca Socket.IO para comunicação em tempo real.

2. Corpo (Body):

- Contém uma barra de navegação com links para as páginas "Detecção" e "Sobre".
- Inclui uma área principal com um título e um link externo para o catálogo da Baumer.
- Possui uma seção de visualização de vídeo e um canvas para exibição dos frames processados.

3. Script:

- Captura frames da webcam.
- Envia os frames para o servidor por meio de Socket.IO.
- Recebe e exibe os frames processados.

Componentes Detalhados

Cabeçalho (Head)

- Meta Tags:
 - `charset="UTF-8"`: Define o conjunto de caracteres como UTF-8.
 - `viewport`: Torna a página responsiva para dispositivos móveis.
- Título: "YOLOv8 Live Detection".
- Estilos:
 - Linka o arquivo `style-deteccao.css` para estilização.
- Scripts:
 - Carrega a biblioteca Socket.IO versão 4.5.4.

Barra de Navegação (Navbar)

- Logo: Um link para a página inicial, rotulado como "YOLOv8".
- Links de Navegação:
 - "Detecção" (`deteccao.html`).
 - "Sobre" (`sobre.html`).
- Menu Responsivo:
 - Um botão hamburger para alternar a exibição do menu em dispositivos móveis.

Seção Principal

- Título: Exibe "Link para o catálogo Baumer".
- Link Externo:
 - Um ícone de link externo redireciona para `https://www.baumer.com.br/catalogo`.

Área de Detecção

- Vídeo:
 - Elemento `` para capturar e exibir a transmissão da webcam.
- Canvas:
 - Elemento `` para desenhar frames anotados com as detecções.

Scripts e Funcionalidades

1. Botão Hamburger

- Alterna a exibição dos itens da barra de navegação ao clicar.

2. Inicialização da Webcam

- Solicita permissão para acessar a webcam do usuário.
- Exibe a transmissão ao vivo no elemento ``.

3. Processamento de Frames

- Captura frames da webcam usando o elemento ``.
- Converte os frames para uma string base64 e os envia ao servidor via Socket.IO.
- Exibe uma mensagem de erro caso o frame capturado esteja vazio.

4. Recepção de Frames Processados

- Recebe frames anotados com detecções do servidor.
- Desenha os frames recebidos no `` para exibição.

Fluxo de Execução

1. A página solicita permissão para acessar a webcam e exibe o vídeo.
2. Os frames capturados são processados a cada 900 ms e enviados ao servidor.
3. Frames anotados são recebidos e exibidos em tempo real.

Estilização

A página utiliza o arquivo de estilo externo `style-deteccao.css` para:

- Customização da barra de navegação.
- Estilização do título e link externo.
- Responsividade do layout.

Dependências

- Socket.IO: Comunicação em tempo real entre cliente e servidor.
- Navegador Compatível: Suporte para `getUserMedia` (para acesso à webcam) e Canvas API.

Notas

- O servidor WebSocket deve estar configurado para processar e responder aos frames recebidos.
- Garantir que a câmera e as permissões de vídeo estejam habilitadas no navegador.