#### 一、项目概述

本系统实现了基于浏览器端TensorFlow.js框架的实时目标检测解决方案，核心特点包括：(1) 支持通过目标物品图像定义检测目标，(2) 融合YOLOv11目标检测与MobileNet特征匹配的双阶段检测架构，(3) 浏览器端模型缓存与优化推理，(4) 动态自适应阈值机制。系统采用模块化设计，涵盖模型加载、图像处理、特征工程、相似度计算、实时检测等关键技术模块。

#### 二、技术架构分析

1. 模型架构

• YOLOv11检测模型: 基于TensorFlow.js GraphModel格式，输入分辨率640×640，输出层包含边界框坐标和分类得分，采用非极大值抑制（NMS）进行候选框筛选。

• MobileNet特征提取: 使用MobileNetV2（α=1.0）作为特征提取器，输出1280维特征向量，并通过多尺度特征融合增强鲁棒性。

• 模型缓存机制: 利用IndexedDB实现模型持久化存储，只需第一次使用时从云端下载，后续便支持离线使用，本地推理。

2. 实时检测流程

A [摄像头初始化] --> B [帧捕获] --> C [YOLO目标检测，获取候选框] --> D [ROI特征提取] --> E [多尺度特征融合] --> F [混合相似度计算] --> G [动态阈值筛选] --> H [最优结果输出]

#### 三、关键技术实现

本系统基于浏览器端深度学习框架，集成YOLOv11目标检测与MobileNetV2特征提取双模型，实现实时精准识别。通过四向旋转和缩放处理提取目标多尺度特征，结合曼哈顿距离、欧氏距离和余弦相似度三元度量算法，构建混合相似度评分体系。创新动态阈值机制，依据目标尺寸占比非线性调整判定阈值，平衡大小目标检测灵敏度。

采用TensorFlow.js内存自动回收与显式释放策略，结合画布内存重置技术，控制内存峰值低于300MB。模型加载通过IndexedDB缓存与CDN双轨并行，提升40%加载效率。检测流程引入10fps帧率控制 (可自行调整)、ROI预筛及类别分组过滤，减少无效计算。支持WebGPU加速与安全上下文检测，适配Chrome/Safari等主流浏览器。

#### 四、核心优化策略

本系统通过三重优化策略实现高效实时检测：模型加载层面采用YOLO与MobileNet异步并行加载技术，结合`tf.io.listModels()`缓存验证机制，实现40%的加载效率提升；检测性能方面，通过RAF帧率控制（10fps可调）与ROI预筛技术过滤大量干扰目标，结合WebWorker将特征计算任务分流至后台线程，降低主线程计算延迟；浏览器兼容性层面，采用标准化媒体设备访问接口`navigator.mediaDevices.getUserMedia()`实现跨平台摄像头适配，严格的安全上下文检测（HTTPS/localhost）保障隐私合规，并集成WebGPU实验性加速后端，在Chrome等支持浏览器中实现30%的推理速度提升，形成覆盖模型加载、计算优化、跨平台适配的全链路技术方案。