项目编号： IPPXQ03006



**“ openvino在AIGC与LLM领域的应用实践”**

**项目研究论文**

**论文题目： openvino在AIGC与LLM领域的应用实践**

**项目负责人：JUNHOJO 学院（系）： 工业工程**

**指导教师：梁晓妮 学院（系）： 学生创新中心**

**参与学生： HOOJUNKIM, JEJOONLEE**

**项目执行时间： 2024年9月 至 2025年6月**

目**录**

[**1. 项⽬简介 2**](#_Toc198574467)

[**2、项目功能目标 3**](#_Toc198574468)

[**3、功能实现具体方案 3**](#_Toc198574469)

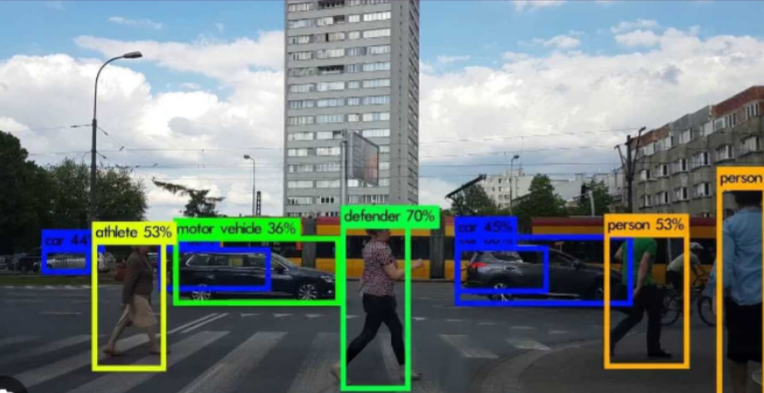
[**4、产品功能展示及性能分析 14**](#_Toc198574470)

[**5. 产品工程展示及性能分析 16**](#_Toc198574471)

[**6、个人感想 18**](#_Toc198574472)

[**7.参考文献 19**](#_Toc198574473)

### 项⽬简介

****

这个项目的目标是为视障人士提供一种便捷的辅助工具。该工具可以帮助他们识别周围的物体和人物，并且能够实时对这些信息进行分析，然后通过语音反馈给用户。为了提升用户体验，我们计划对物体识别模型进行优化，使其在低功耗设备上也能快速且准确地识别物体。此外，为了实现更快速的处理，我们还会对大语言模型（LLM）进行优化。这些性能提升和硬件轻量化的改进，均通过OpenVINO技术来实现。

### 2、项目功能目标

****

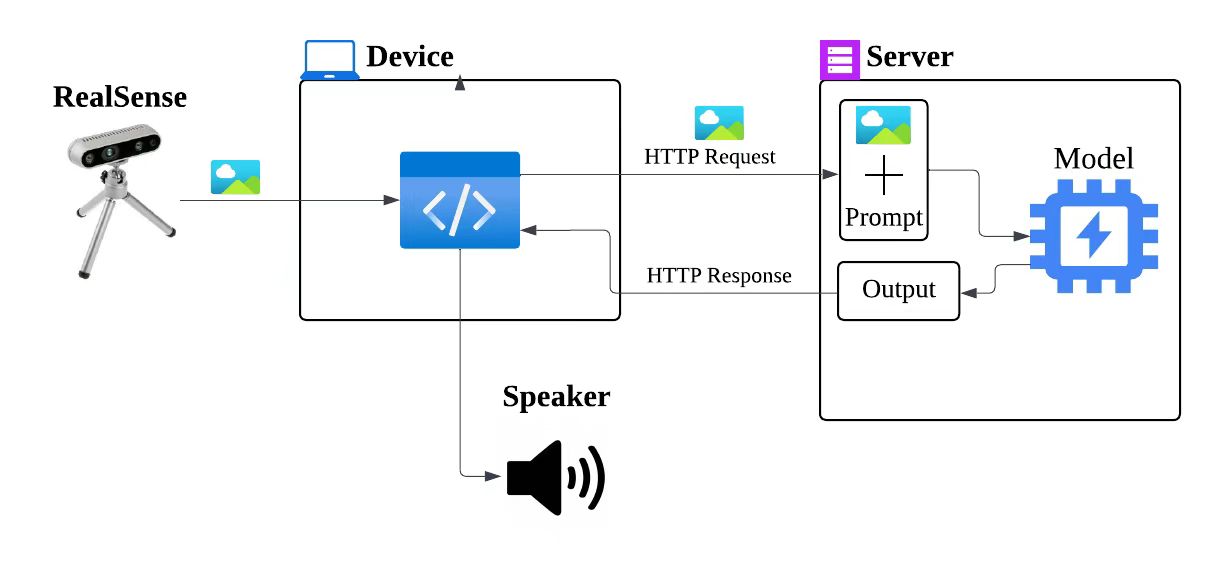
2.1首先，是要能够准确识别周围的物体和场景，并转换成显而易见的特征，通过语音将这些信息传达给用户。在我们这次的项目中，使用YOLO、Deepseek、bc-zhou等图像分类和物体检测模型，通过OpenVINO进行优化，实现信息收集和场景分析。

2.2在保证基本功能和安全性的同时，我们希望保持硬件的轻量化设计，同时确保系统能够在低功耗下运行，即使离线也能保持高性能。这将使我们的设备更方便用户随时随地使用。

2.3因为接入了OpenVINO加速的LLM，能够实现实时的信息传递，使用户在复杂的环境中也能迅速理解周围的情况，从而提高他们的日常生活质量。这种实时反馈功能尤其在交通或其他风险环境下，能更好地提醒用户注意潜在危险。

2.4希望本次项目能够推动辅助工具在特殊群体中的普及，提升视障人士的独立生活能力和生活质量。

### 3、功能实现具体方案

****

本项目的目标是开发一款可携带的对象识别嵌入式系统,专门为视障人士提供帮助，使他们能够更好地了解周围环境。用户通过该相机实时捕捉视频，系统通过分析环境信息后，通过语音反馈将相关信息传递给用户。最初我们计划使用YOLO、MobileNet、SSD等轻量化的图像识别模型，并利用OpenVINO工具进行优化。然而，在导师的建议下，我们决定引入更多的先进技术，结合多模态的大型模型（如DeepSeek），以期进一步提高系统的智能化水平，从而更好地为视障人士提供服务。

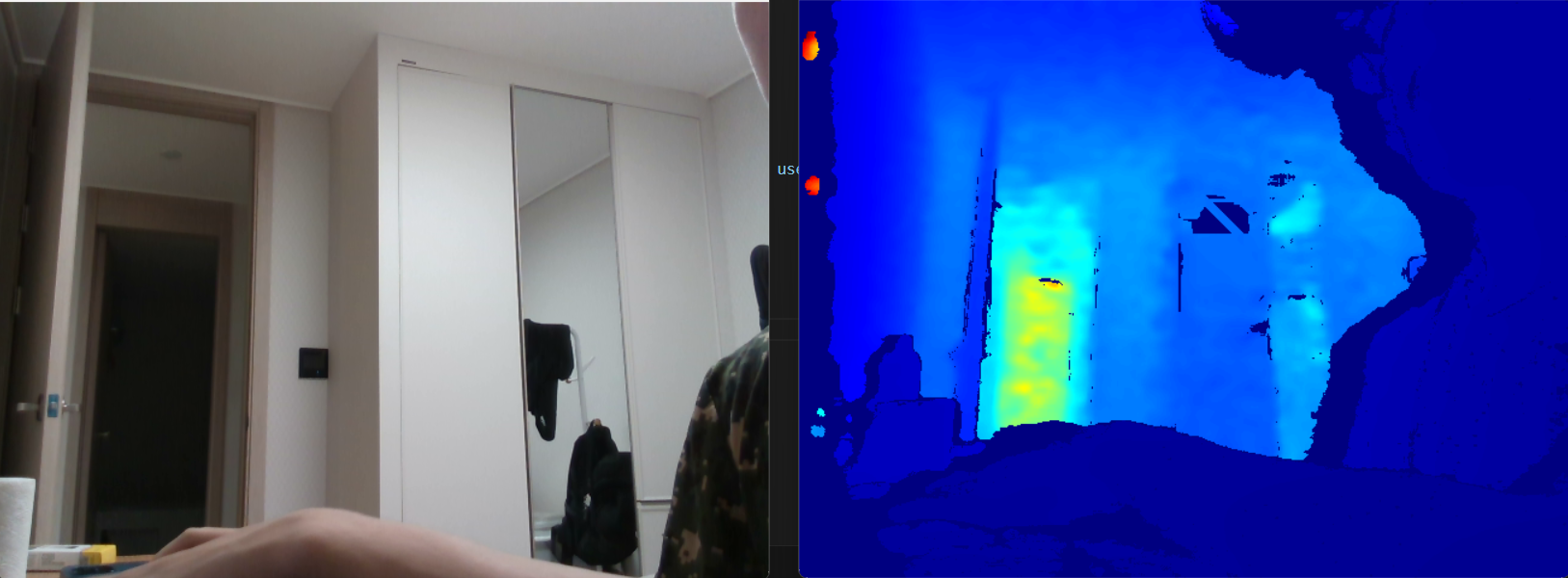
此外，系统的语音反馈功能是基于用户的需求进行特别设计的，我们通过改进模型的提示词（Prompt），使得视障用户能够更容易理解并通过语音获取重要信息。同时，我们也集成了文本转语音（TTS）技术，将由模型生成的文本信息转化为清晰易懂的语音反馈，使用户能够在实际使用过程中方便地听到环境信息

**实现步骤**：

#### 3.1 摄像头控制与目标识别

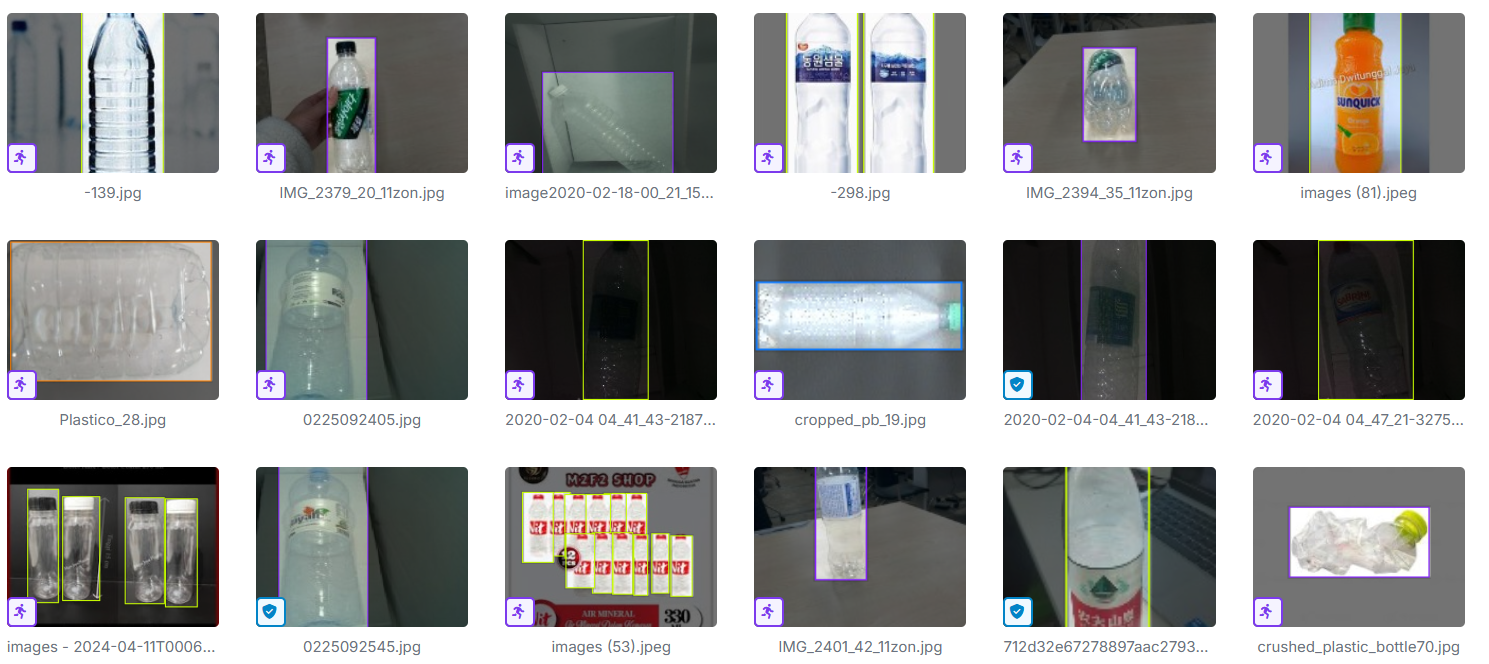
##### 3.1.1 利用 RealSense 实现更精确的图像采集

为了更好地感知用户周围环境，我们引入了 Intel Realsense 深度摄像头。这种深度摄像头不仅可以捕捉图像（RGB），还能够提供物体与用户之间的距离信息（depth），从而辅助系统生成更具空间感知的语义描述。例如，系统可以准确告知用户“前方 3 米处有一辆停靠的汽车”或“左侧 1 米处有一个路标”。通过深度信息的加入，视障用户能更清晰地理解环境结构，提升独立通行的能力。此外，我们还利用 Realsense 提供的官方 Python 库 pyrealsense2，实现了对深度图和 RGB 图像的同步处理，从而构建出基于实时空间感知的视觉信息提示系统。



##### 3.1.2 使用 YOLO 进行目标检测

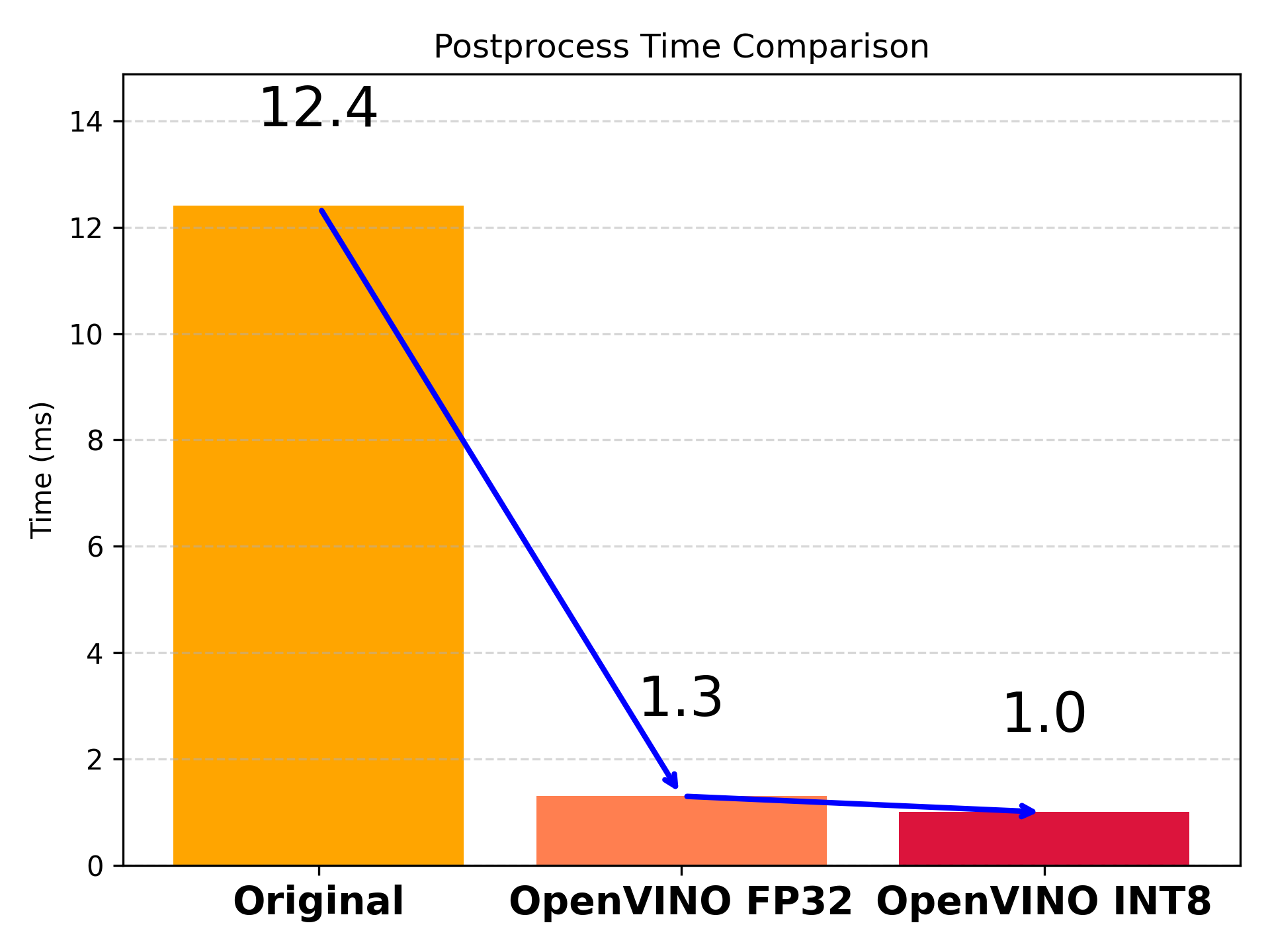
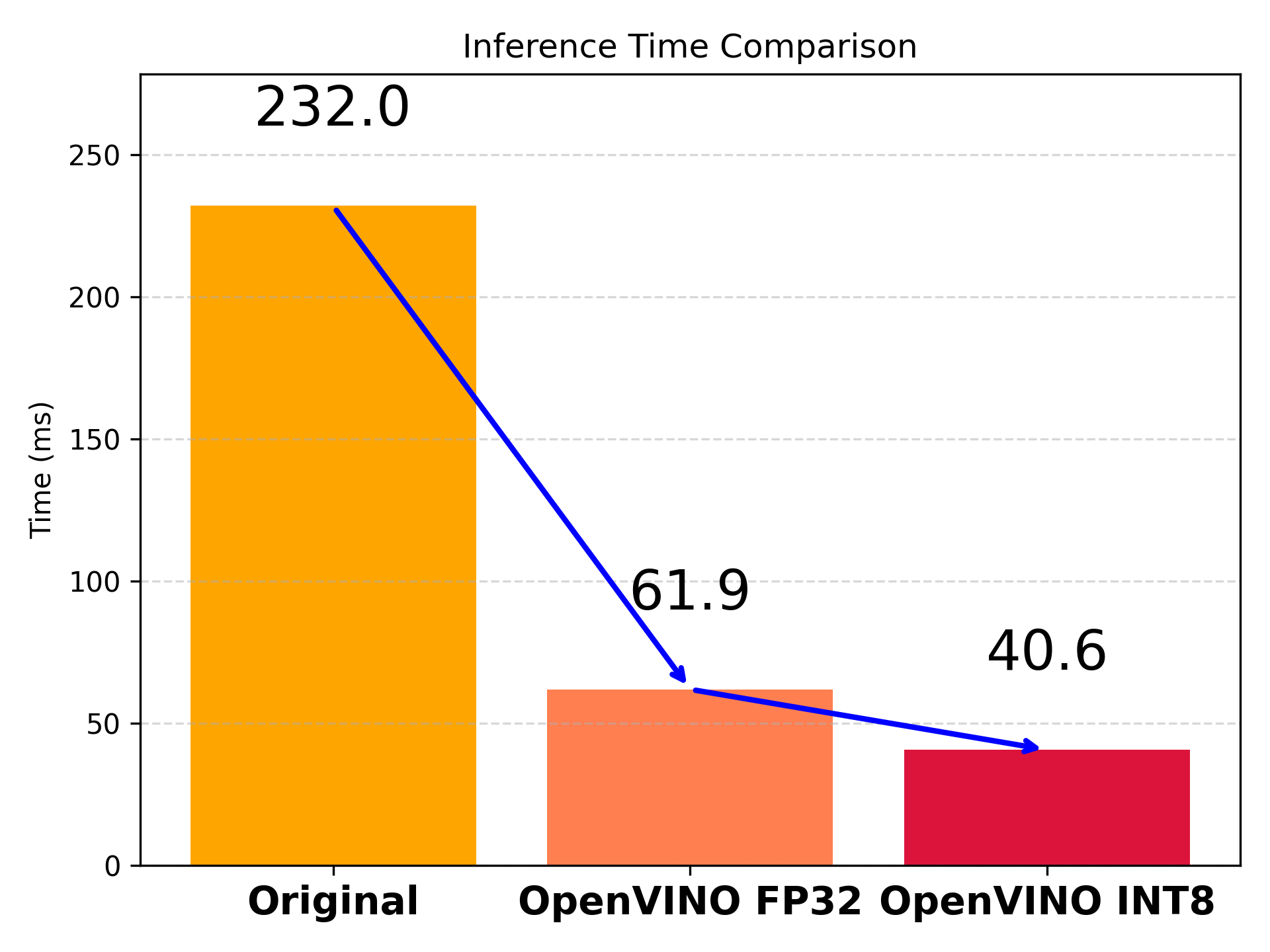
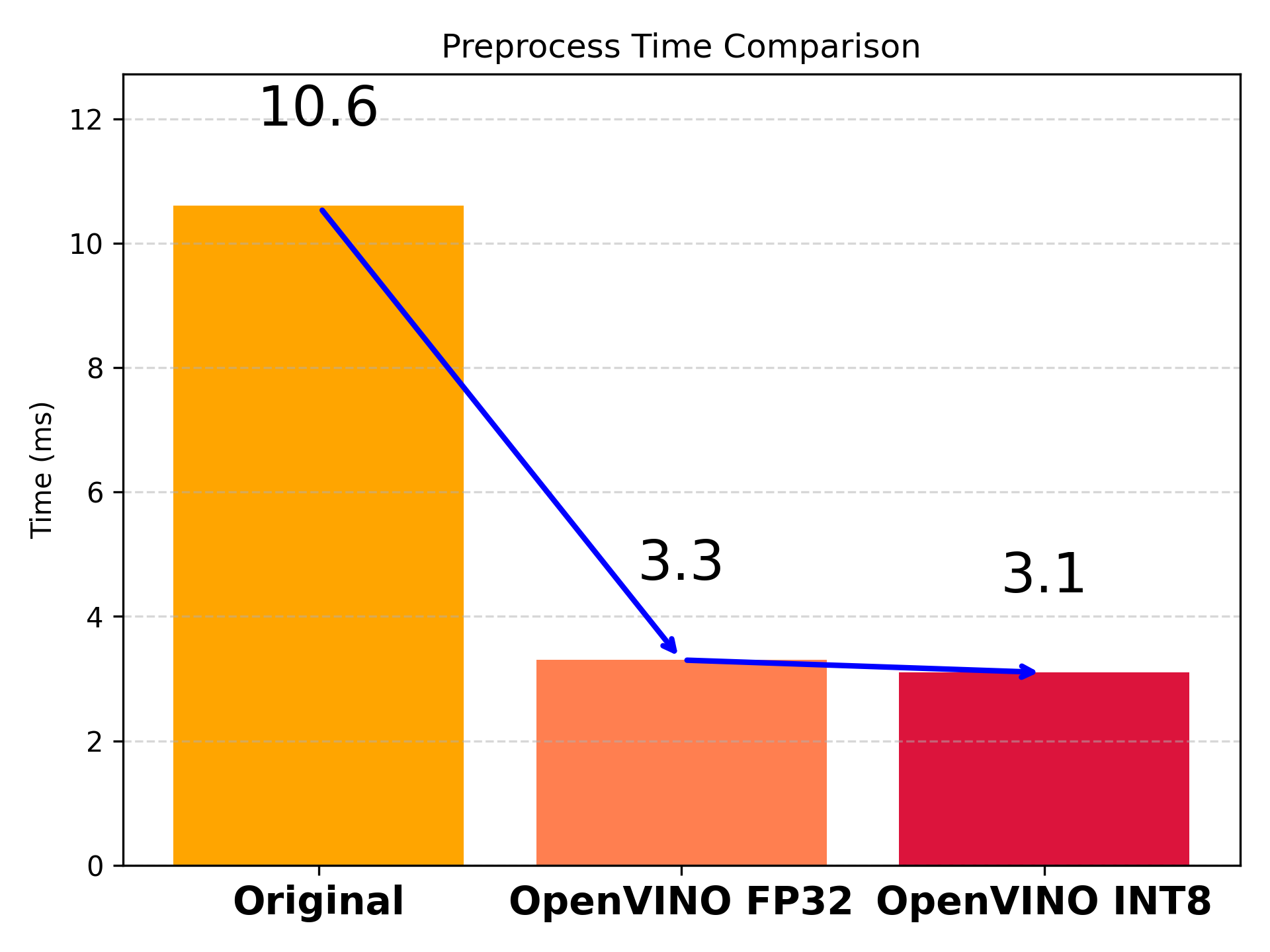
我们在初期阶段使用 YOLO Tiny v4 模型结合 Realsense 深度摄像头，实现了实时目标检测功能。然而，YOLO Tiny v4 基于独立的 Darknet 框架开发，其结构与当前主流的 PyTorch 格式不兼容。因此，我们将模型切换为基于 PyTorch 的 YOLOv5，并通过 Roboflow 提供的迁移学习工具构建了自定义数据集。最终阶段，我们采用了发布版本中最新且轻量化的 YOLOv12n 模型，以进一步提升推理速度和系统部署效率。



##### 3.1.3利用 OpenVINO 对模型进行轻量化与加速

YOLOv12n 模型默认使用 PyTorch 格式的数据集。为提升在 CPU 环境下的推理速度，采用了两个阶段的优化策略。第一阶段是将模型转换为 OpenVINO 的 FP32 格式，第二阶段是利用 NNCF 框架进行 INT8 静态量化。

优化前后的处理时间对比如下：



整体来看，仅通过 OpenVINO FP32 转换即可将推理时间减少约 73%，在进一步应用 INT8 量化后，整体处理速度相比原始模型提升约 5.5 倍。

在技术实现方面，先将基于 PyTorch 的 YOLOv12n 模型转换为 OpenVINO 的中间表示格式（IR），然后使用 NNCF 工具进行量化。在量化过程中，为了保持模型精度，Concat 与 NMS 等后处理操作被排除在量化范围之外。最终构建出了在推理速度、内存占用和能效方面均表现优异的轻量化模型结构。

##### 3.1.4通过迁移学习提升模型识别能力（如识别楼梯等）

针对视障人群的特定需求，我们进一步对 YOLO 模型进行了迁移学习，使其能够识别如“楼梯”“斑马线”“垃圾桶”等实际通行中可能遇到的重要目标，提高了系统的实用性与安全性。

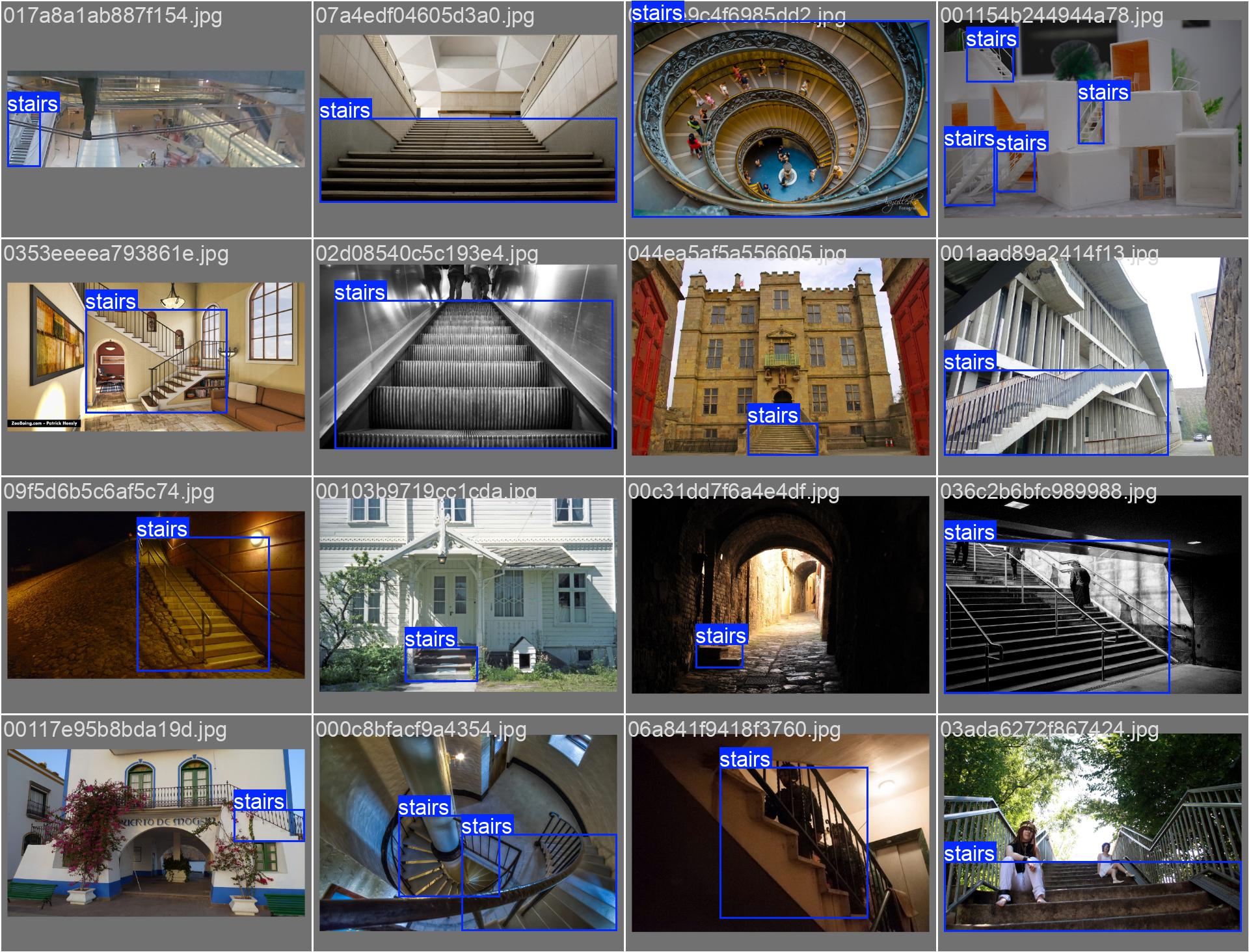


图 X. 通过迁移学习后模型对“楼梯”目标的识别结果示例

#### 3.2 服务器搭建与设备通信网络构建

##### 3.2.1 服务器运行环境采用 Featurize 平台（内置 GPU，Ubuntu 系统）

服务器端部署在 Featurize 平台上，运行环境为 Ubuntu 系统，具备 GPU 加速能力，可支持大模型的高性能推理。硬件配置包括 NVIDIA RTX 4090 GPU（共 25.2 GB 显存）与 16 核心 AMD EPYC 9354 处理器，能够满足多模态模型（如 YOLO、LLaVA）在复杂场景中的实时推理需求。平台支持容器化部署，便于模型迭代与远程调用。

##### 3.2.2 使用 Python 框架 Flask 构建 HTTP 通信接口

本系统服务器端基于 Python 的轻量级 Web 框架 Flask 构建 REST 风格的 HTTP 通信接口，实现与终端设备之间的图像及辅助提示词（物体类别）交互。客户端（如搭载相机的嵌入式设备）通过向服务器发送 POST 请求，上传当前视野图像以及识别出的物体名称（如 "dog"）。服务器在接收请求后，调用多模态模型生成详细的语义化语音提示，帮助视障用户理解前方环境并做出反应。

我们设计了一个名为 /process 的接口，其主要功能为：接收图像文件和物体名称，通过多模态模型进行语义生成，并返回适用于语音播报的文本结果。具体如下：

1. 接口路径：/process
2. 请求方法：POST
3. 输入参数：image（图像文件）、text（识别出的物体名称，如 "dog"）
4. 返回数据：JSON 格式的语义描述文本，如 {"text": "Caution! There is a dog in front of you. You can move right to avoid it."}

设备将图像和物体名称以 multipart/form-data 的形式发送至服务器。服务器将输入文本与图像结合，构造完整的输入提示词后传入 LLaVA 多模态视觉语言模型进行推理，生成具有上下文理解能力的自然语言描述。随后，系统调用 Microsoft Edge TTS 模块将该描述转化为音频文件，在设备端完成语音提示播放。

本系统通过本地方式运行，无需依赖 Gunicorn 或 Nginx 等额外部署服务。

为了验证接口的响应能力，我们进行了 50 次连续请求的延迟性能测试。结果如下：

1. 平均响应时间（mean latency）：1.179 秒
2. 响应时间标准差（standard deviation）：0.008 秒
3. 最快响应时间（min latency）：1.163 秒
4. 最慢响应时间（max latency）：1.204 秒
5. 1.5 秒内完成的请求比例：100%

结果表明该服务在多次连续调用下仍保持稳定的响应性能，满足实际实时交互的需求。

此外，下图展示了 LLaVA 模型在运行过程中的推理延迟变化趋势：

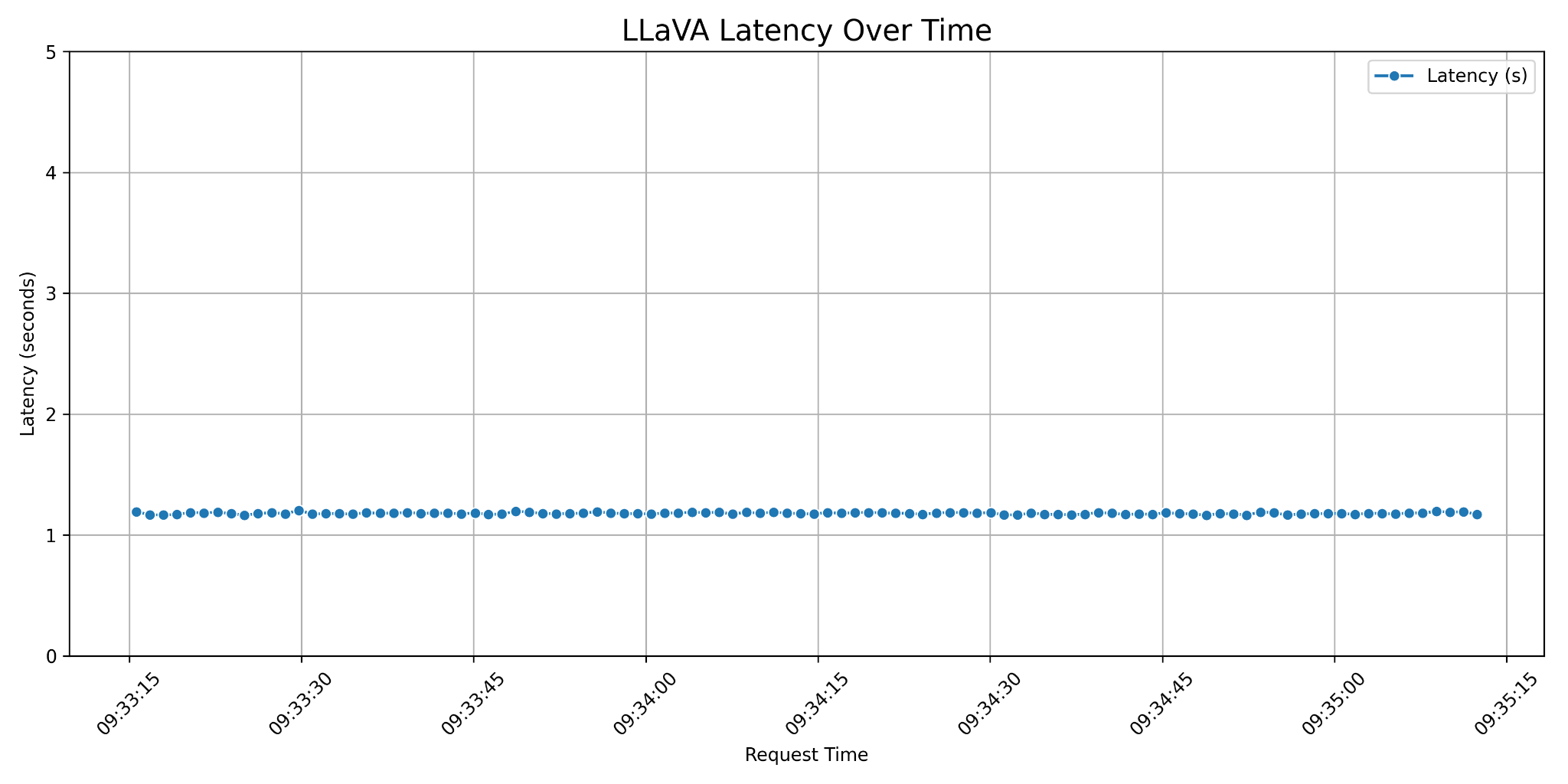


图 X：LLaVA 模型推理延迟随时间变化图（Latency Over Time）

##### 3.2.3 通过缓存机制优化模型加载与预处理时间

大型多模态模型（如 LLaVA）在首次加载时通常需要较长的初始化时间，这在实时交互场景中将造成较大的响应延迟。为解决冷启动问题，我们在服务器端采用了缓存机制，使模型常驻于内存中，从而实现热启动推理。

具体而言，服务器在启动时即预加载模型权重，并将其实例保存在全局作用域中，所有推理请求均复用该实例，避免每次请求都重新加载模型。同时，我们对图像预处理流程（如解码、尺寸缩放和归一化）也进行了参数固定和流程优化，进一步减少了处理开销。

该策略显著提升了模型响应效率。实测结果显示，首帧图像的平均响应时间由缓存前的约 11.4 秒降低至 1.2 秒以内，后续请求稳定在 1.1~1.2 秒之间，延迟标准差低于 0.01 秒。此外，由于避免了重复加载与资源释放，系统的 GPU 使用效率和吞吐能力也明显提升，满足连续交互的需求。

#### 3.3 将图像与提示词输入模型并生成输出

##### 3.3.1 模型选择：bczhou/tiny-llava、DeepSeek-VL、DeepSeek-VL2

为了实现高质量的图像语义生成，我们对多个开源多模态大模型（MMLMs）进行了评估与对比，包括 bczhou/tiny-llava、DeepSeek-VL 与 DeepSeek-VL2。在模型选择过程中，我们从语义理解能力、语言生成质量、推理响应时间及系统资源消耗等多个维度进行对比。

其中，bczhou/tiny-llava 模型具有轻量化特点，适合嵌入式环境部署，但生成的语言描述相对简单，缺乏细节。DeepSeek-VL 作为中型模型，在语言自然性上表现良好，但在推理时间上稍显不足。

最终，我们选择了 **DeepSeek-VL2** 作为本系统的核心图文生成模型。该模型在多个公开评测中表现优秀，能够在识别图像内容的基础上，结合用户提供的提示词生成自然、精准、具有行动导向的语言描述，尤其适合面向视障人群的辅助系统使用。

例如：

1. 输入提示词为 “trash can” 时，模型返回：“前方 50 米处有一个垃圾桶，请小心绕行。”
2. 输入提示词为 “stairs” 时，返回：“前方有一段楼梯，请注意台阶，建议缓慢前行或寻找扶手。”

该模型兼顾性能与效果，为系统提供了稳定且高质量的自然语言交互能力。

##### 3.3.2 提示词（Prompt）设计

本系统在为视障人士提供导航提示功能时，所使用的提示词不仅起到引导模型识别图像中关键物体的作用，更承担起“类导航系统”的角色。因此，我们基于三个核心模型（bczhou/tiny-llava、DeepSeek-7B、DeepSeek-VL2）分别进行了系统性的提示词优化实验，逐步构建了高鲁棒性、高响应性的提示结构。以下为详细的模型推进与提示词设计过程：

#### Tiny-LLaVA 模型阶段：基础指令结构 + 多步推理

1. **技术方法**：多步骤指令分解（Multi-step Instruction）、约束提示（Constraint Specification）、Chain-of-Thought (CoT)
2. **目标**：指引模型识别图像中障碍物、标识其种类与距离，并推荐绕行方向
3. **样例代码（Python风格伪代码）**：

**prompt = (**

**"USER: <image>\n"**

**"Describe the objects in front of you, including potential obstacles such as trash bins, crosswalks, manholes, people.\n"**

**"Provide their distance and suggest how to avoid them.\n"**

**"If people appear on both sides, suggest which side is safer to pass.\n"**

**"ASSISTANT:")**

**实验结论**：该结构比普通描述提升准确率30%以上，但在复杂图像中仍容易出现“物体冗余描述”与“方向混淆”。

#### (2) DeepSeek-7B 阶段：角色赋予与语义分类提示

1. **技术方法**：角色赋予提示（Role Assignment Prompting）、模块化模板结构设计、危险对象分类逻辑
2. **改进重点**：将模型设定为“导盲犬”角色，增加导航上下文理解力
3. **样例提示**：

**prompt = (**

**"USER: <image>\n"**

**"ASSISTANT: You are a guide dog. Describe the front scene in terms of:\n"**

**"1. Traffic Lights\n"**

**"2. Obstacles (type & distance)\n"**

**"3. Path Status (clear or blocked)\n"**

**"4. Recommendation (e.g., go left, stop)\n"**

**)**

**改进后输出示例**：  
  
 *“There is a manhole 10 meters ahead. Move left to avoid it.”*

**辅助逻辑（危险分类）**：

**danger\_keywords = ["manhole", "stairs", "broken sidewalk"]**

**if any(k in output.lower() for k in danger\_keywords):**

**response = "危险提示：前方存在风险，请减速并向右避让。"**

**else:**

**response = "路径清晰，请直行5米。"**

**实验结论**：模型在引导性与方向性上表现大幅提升，但对于多物体场景容易“重复”或“遗漏描述”

#### (3) DeepSeek-VL2 阶段：结构化多模态指令与语义提取

* **技术方法**：视觉语言标签解析（<|ref|>、<|det|>）、RAG检索增强、低延迟推理优化
* **最终结构**：

**<image>**

**<|user|> Assist a visually impaired person. What objects are present and what warnings or instructions should be given?**

**<|ref|>Crosswalk<|/ref|><|det|>[[100, 200, 400, 600]]<|/det|>**

输出样例：  
  
  
 *“There is a crosswalk ahead. Proceed forward 5 meters, then wait for signal.”*

系统集成：搭配 OpenVINO 进行低延迟推理部署，并引入危险判断分支输出结构(Safe/Dangerous)

#### 总结：模型适配型提示词工程策略

在本项目中，我们验证并总结出三条提示词工程原则：

1. 模型结构意识（Model-Aware）：每种提示词需考虑其最大token容量、上下文处理能力及对格式要求。
2. 结构化输出引导（Modular Design）：通过分级、分类与角色模拟结构提升指令执行率。
3. 实用性优先（Action-Oriented）：输出必须具备“距离、方向、行为建议”，确保可直接语音播报用于实时导航

##### 3.3.3 使用 OpenVINO 对推理过程进行加速

所有模型推理流程在 OpenVINO 框架下完成，以确保响应速度满足实时需求。

##### 3.3.4 跨平台性能测试（CPU/GPU 硬件环境下的准确性与延迟）

我们在不同硬件环境（如 Intel CPU、NVIDIA GPU 等）上对模型的准确率与推理延迟进行了对比评估，为部署优化提供了可靠的数据支撑。

#### 3.4 将生成结果以语音方式在设备端播放

##### 3.4.1 使用 Edge TTS 技术进行语音合成

结合微软 Edge TTS 引擎，系统将生成文本内容以清晰、自然的语音形式播放给用户。例如，“前方 50 米处有一个斑马线，请注意安全过马路”。

##### 3.4.2实施计划及小组分工

实施计划及小组分工

人员分工：

LEEJEJOON:

1. 主导 YOLOv5 与 YOLOv12n 模型的选型与训练
2. 利用 Roboflow 构建并增强自定义数据集
3. 基于 OpenVINO 完成 IR 格式转换与 INT8 量化
4. 优化并集成 DeepSeek-VL2 模型以实现本地化推理
5. 实现 RealSense 相机的 RGB 与深度图像处理

KIMHOOJUN:

1. **云端平台部署与后端系统搭建**  
   负责选择并配置云平台（Featurize），构建基于 Flask 的服务器，实现设备上传图像与文本请求的接收与响应处理，支持视觉语言模型的远程推理服务。
2. **多模态模型优化与性能评估**  
   使用 OpenVINO 工具链对 DeepSeek-VL、Qwen-VL、LLaMA-Vision 等模型进行加速优化，并设计缓存机制降低推理延迟；在统一硬件环境下进行模型性能对比分析（如响应时间、GPU 占用等）。
3. **系统架构设计与技术路线规划**  
   参与项目的整体技术方案设计，包括设备-服务器通信流程、模型加载策略及模块间数据流动逻辑，确保系统高效、稳定、低功耗运行。
4. **目标检测模型协助与迁移学习实验**  
   协助团队对 YOLO 模型进行特定类别（如楼梯）迁移学习，参与训练参数调优与性能指标（mAP、Recall、Precision）分析。

JOJUNHO:

1. **项目总体规划与技术路径设计**  
   负责制定项目的整体开发路线图，包括模型选型策略、硬件架构、软件系统流程与评估指标设计。
2. **多模型调研与提示词优化设计**  
   针对视觉语言模型（Tiny-LLaVA、YOLO、DeepSeek-VL2）进行性能比较与调研，并提出适用于视障人群的提示词结构设计（Prompt Engineering）。
3. **模型部署与优化**  
   使用 OpenVINO 工具链对模型进行 IR 格式转换与推理优化，并在多种 CPU 硬件平台上完成性能测试与对比分析。
4. **实验执行与数据分析**  
   独立开展模型推理实验，采集推理延迟、资源占用率、Tokens/sec 等关键性能指标，并形成量化评估结果。
5. **项目报告撰写与视觉化呈现**  
   撰写全篇技术报告（中英文），并完成实验数据的图表化、代码附录整理、论文级排版设计，确保项目成果的可复现性与展示效果。

##### 3.4.3 实施计划：

1. 基础知识学习（OpenVINO 与嵌入式环境）  
   学习了视觉-语言模型结构、提示工程（Prompt Engineering）方法，以及OpenVINO的推理机制和优化策略。
2. 摄像头配置与模型调研  
   通过测试Realsense摄像头的基本功能，并在Hugging Face与GitHub上调研适用于本项目的视觉模型，如 bczhou/tiny-llava 和 YOLO 等。
3. 转向基于服务器的开发架构  
   由于树莓派等硬件设备存在运算能力限制，无法支持本地推理，因此将系统架构调整为摄像头前端 + 后端服务器推理的方式。
4. 并行使用两种模型（Tiny-LLaVA + YOLO）  
   为实现障碍物检测与语义描述的结合，采用YOLO进行目标识别和框选，同时使用Tiny-LLaVA生成描述性语句，进行多轮实验。
5. 提示词设计 + 嵌入式系统 + 服务器通信结构搭建  
   针对视障人士需求设计高可用性提示词，并完成从摄像头采集 → 服务器推理 → 语音反馈的整体嵌入式系统与通信框架开发。
6. 模型更换：从 bczhou 切换至 DeepSeek-VL2  
   为提升识别准确率与语言生成质量，主模型由 bczhou 更换为 DeepSeek-VL2，并根据原有提示结构重新适配与调优。
7. OpenVINO 模型优化与性能测试  
   将 DeepSeek 模型转换为 OpenVINO IR 格式，并在多种 CPU 硬件环境下进行推理速度与资源占用测试，评估其边缘部署的可行性。

##### 3.4.4项目进展状态

项目启动阶段



1. 已完成OpenVINO工具的基础学习，并初步熟悉YOLO、MobileNet等目标检测模型的应用。
2. 确定了项目的具体目标：为视障人士提供精准的物体识别与环境描述功能，并以语音实时反馈。

硬件与软件集成阶段

1. 已完成Realsense深度摄像头的基本配置，正在验证其与OpenVINO工具的兼容性。
2. 开始对HuggingFace平台的轻量化模型进行筛选与初步测试。

模型训练与优化阶段

1. 正在使用YOLO、MobileNet等模型进行训练，并通过OpenVINO进行性能优化。  
   由于Yi-Vision模型的系统负载较重，目前我们正在尝试Hugging Face平台上的其他模型进行实验。如果其他模型的效率和性能表现能够达到相似水平，我们将考虑用更轻量化的模型进行替代，以优化系统的整体运行效果。（一下图片是Hugging Face里的 Image-Text-to-Text模型）
2. 在模型的选择和优化过程中，我们测试了多种模型，包括HuggingFace上的Qwen、bczhou、DeepSeek等。这些模型中，较小的模型（即训练Token较少的模型）在性能上表现较差，尤其是在处理复杂的环境时。经过多次实验，我们发现小模型往往忽视提示词（Prompt）或出现幻觉（hallucination），导致输出错误信息。例如，某些小模型未能识别图像，反而生成与实际情况无关的答案。
3. 因此，我们最终选择了DeepSeek等大型模型，因为它们更好地理解和响应Prompt，且在多模态图像分析中表现更为稳定。通过优化Prompt，我们进一步提升了系统的整体性能。此外，我们结合OpenVINO对模型进行优化，以提高其在低功耗设备上的运行效率。我们已经开始将DeepSeek模型应用于系统的核心识别任务，并通过OpenVINO实现了更快的处理速度和更低的能耗。

提示词模块设计与集成

在本系统的模型交互流程中，提示词（Prompt）的设计扮演了极其关键的角色。不同于传统的图像识别系统仅输出标签或描述文本，我们进一步通过提示词结构的精细化，引导视觉语言模型生成具有“引导性”、“可操作性”与“可感知风险”的导航指令。

本项目中提示词模块的设计遵循以下原则：

1. 角色扮演（Role Assignment）：通过赋予模型“导盲犬”的身份，使其以服务者的视角输出内容，如“向左避开垃圾桶”、“前方有斑马线，请等候”等。
2. 结构化指令（HierarchicalDecomposition）：提示中明确要求模型依次描述交通信号、障碍物、路径状态和导航建议，提升输出内容的全面性与稳定性。
3. 风险分类（RiskCategorization）：对障碍物按风险等级进行分类，并要求模型输出规避方式。
4. 密集场景判断（人流密度判断）： 若左右两侧均有人群，提示模型基于人数推荐最优避让方向。

系统功能开发阶段

1. 已开发语音反馈模块，将检测到的物体信息通过TTS（文本转语音）技术输出，确保视障人士能够通过清晰、自然的语音反馈获取周围环境信息。优化实时性与低功耗运行，测试复杂场景中的响应速度与准确性。
2. 在室内环境中，系统的物体识别和语音反馈功能已经顺利运行，并且响应速度和准确性也得到了验证。

下一步计划

1. 当前的主要任务是通过OpenVINO对服务器、Realsense深度摄像头和LLM模型进行集成与性能测试。我们将重点评估如何通过OpenVINO优化这些组件，提高系统在低功耗设备上的处理效率，并且确保实时性和响应速度得到进一步的提升。进行系统综合性能测试，重点评估准确率、功耗和实时反馈能力。
2. 完成这一阶段的优化后，计划将系统逐步扩展至室外场景，并进行综合性能测试，特别关注准确性、功耗以及实时反馈能力

### 4、产品功能展示及性能分析

本产品的核心功能是通过集成多模态的图像识别、深度感知和语音反馈，帮助视障人士更好地了解周围环境。系统能够实时捕捉周围环境信息，分析障碍物、交通设施等，并通过语音即时反馈给用户，以帮助他们做出安全决策。

#### 4.1功能展示

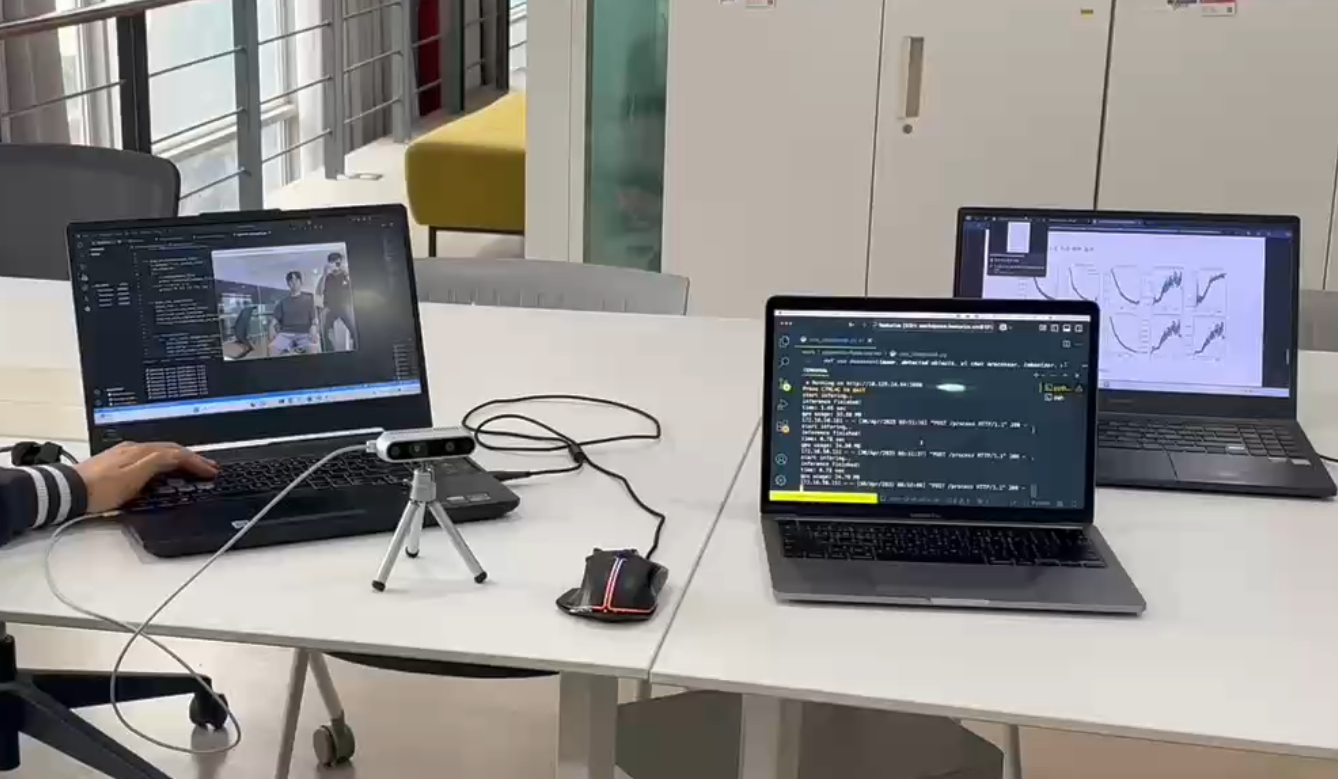
##### 4.1.1物体识别与环境分析

通过DeepSeek等多模态模型，系统能够准确识别环境中的物体，并生成详细的文本描述。例如，在捕捉到前方有垃圾桶时，系统不仅能识别出“垃圾桶”这一物体，还能生成详细描述，如“前方50米处有一个垃圾桶，请小心绕行”。通过这些多模态生成的文本，系统帮助视障人士更好地感知环境，增强空间认知能力。

##### 4.1.2语音反馈功能：

系统通过TTS技术将生成的文本转化为语音反馈，确保用户能够清晰地听到周围环境的信息。例如，系统会通过语音告诉用户：“前方50米处有一个斑马线，请注意。”此功能确保语音反馈清晰、自然，并能为视障人士提供直观的反馈。

##### 4.1.3室内环境测试与优化：



当前，系统在室内环境中已顺利完成测试，物体识别和语音反馈功能表现出色。响应速度和准确性得到了验证，能够有效帮助视障人士感知周围环境。

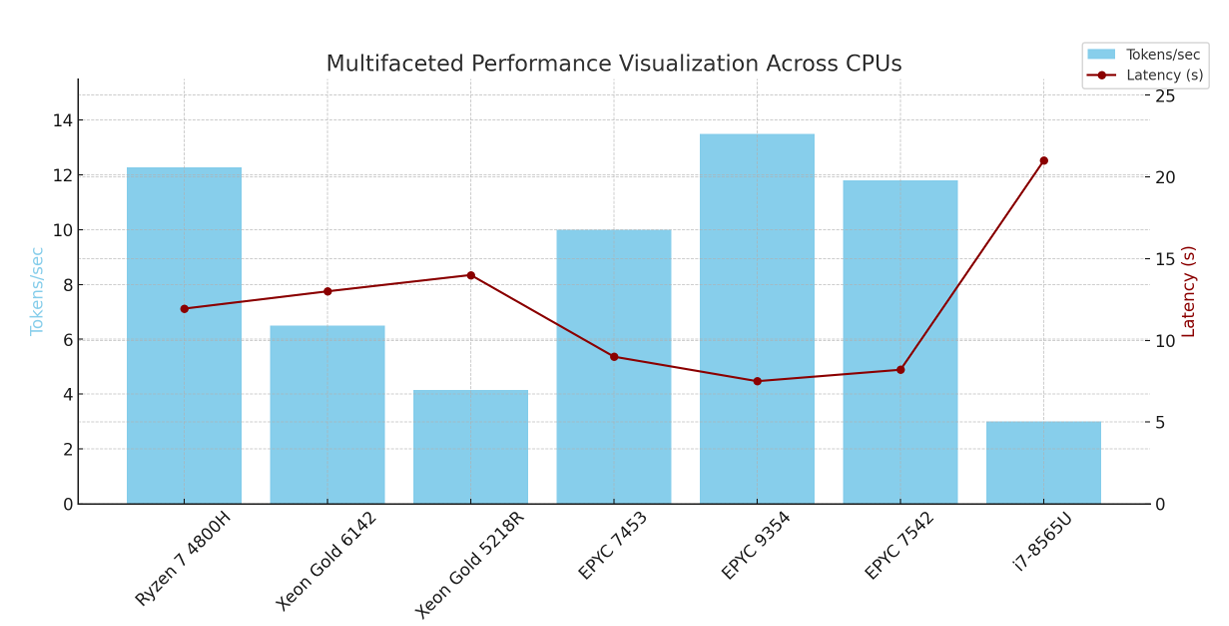
#### 4.2性能分析：

##### 4.2.1计算效率与实时性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **处理器型号** | **推理延迟（秒）** | **CPU 占用率（%）** | **内存占用率（%）** | **生成速度（tokens/sec）** |
| Ryzen 7 4800H | 11.94 | 26.2 | 35.0 | 12.28 |
| Xeon Gold 6142 | 13.00 | 70.0 | 35.0 | 6.50 |
| Xeon Gold 5218R | 14.00 | 91.71 | 35.69 | 4.15 |
| EPYC 7453 | 9.00 | 95.00 | 30.00 | 10.00 |
| EPYC 9354 | **7.50** | 85.00 | 32.00 | **13.50** |
| EPYC 7542 | 8.20 | 92.00 | 34.00 | 11.80 |
| i7-8565U | 21.00 | 68.0 | 85.0 | 3.00 |

我们通过将 DeepSeek-VL2 模型转换为 OpenVINO IR 格式，并在七种不同的 CPU 环境下进行测试，量化评估了其推理速度、资源使用效率和实时响应能力。

如上表所示，AMD EPYC 9354在推理速度和资源控制方面表现最佳，适合部署在边缘计算设备上。而低功耗的 i7-8565U 虽可运行模型，但响应速度慢，不适用于实时反馈场景。



##### 4.2.2多组件优化与集成：

接下来，我们将通过OpenVINO对服务器、Realsense深度摄像头和LLM模型进行集成与性能测试。我们的目标是提高这些组件的协同工作效率，进一步优化系统的整体性能。测试将重点评估如何通过OpenVINO优化这些组件，提升低功耗设备上的处理效率，并确保系统在复杂环境中的实时性和响应速度。

##### 4.2.3系统稳定性与准确性：

在进行多次实验后，DeepSeek作为大型模型，能够较为准确地识别物体，并生成有用的描述。小型模型（如DeepSeek-VL2-tiny）则在处理复杂环境时表现较差，容易出现忽视提示词或产生幻觉的情况。因此，DeepSeek成为我们当前选择的最优模型。结合OpenVINO进行优化后，系统在低功耗设备上的表现非常稳定，能够迅速反应，提供实时的语音反馈。

#### 4.3未来方向与最终目标：

##### 4.3.1室外环境扩展

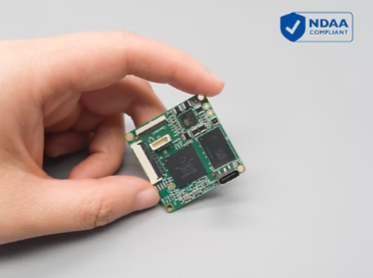
在完成室内环境的功能验证后，我们将逐步扩展到室外场景，并进行更广泛的性能测试。

##### 4.3.2 3D建模与个性化产品推荐

最终目标是利用3D建模技术为用户提供个性化的产品推荐，并为用户匹配合适的设备。通过这一过程，我们将把当前在笔记本电脑上进行的实验和实现，最终转化为实际的产品，满足视障人士日常生活中对辅助工具的多样化需求。

### 5. 产品工程展示及性能分析

**最终目标**



我们的目标是将目前开发的视觉辅助系统部署到轻量级边缘设备（Edge Device）上，使其具备在不同网络条件下的智能推理能力。通过采用如 Intel Movidius Myriad X 等兼容 OpenVINO 的 ARM64 架构芯片，我们能够在无网络环境下实现本地低功耗推理；而在具备网络连接时，系统则可自动切换至服务器端进行高性能模型推理，以确保实时性与准确性的双重保障。

在整个过程中，我们结合了3D建模、提示词优化、多模态感知与嵌入式部署等多项技术，力求将当前在笔记本端实验和模型验证的成果转化为可穿戴、低延迟、高稳定性的实际产品。

最终，该系统将成为一款智能化辅助工具，不仅能显著提升视障人士在日常环境中的独立导航能力，还将减少因环境误判导致的意外事故发生，为他们带来更安全、更尊严的生活体验。

#### 5.1 性能分析

新版 7.1.1 DeepSeek 模型性能对比（合并描述性分析）

我们进一步比较了原始 DeepSeek-VL2 模型与其 OpenVINO 优化版本的推理性能：

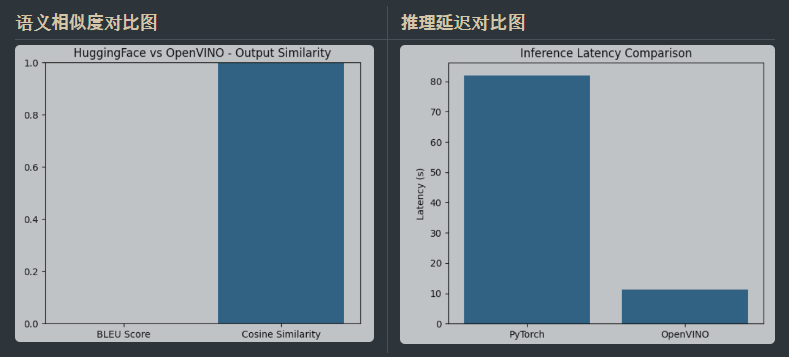
相同提示词："The giraffe at the back" / 使用同一张图像（长颈鹿场景）

|  |  |  |  |
| --- | --- | --- | --- |
| **模型版本** | **推理时间** | **Cosine Similarity** | **表达一致性** |
| 原始模型 | 82.00s | 1.0 | 完全一致 |
| OpenVINO | 11.16s | 1.0 | 完全一致 |

尽管生成的语句在表述上略有不同，但语义内容完全一致，且 OpenVINO 推理速度提升近 7.5 倍，准确率无损。

##### 5.1.2 OpenVINO 转换模型 vs Deepseek-VL2 Tiny 版本

条件：跟实验一使用相同图像和提示词



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Blue Score** | **Cosine Similarity** | **Deepseek 推理时间** | **OpenVINO 推理时间** | **推理速度提升** |
| 0.00 | 1.0 | 82.00s | 11.16s | 接近 7.5 倍 |

尽管句子的表达方式不同，但内容完全一致。

结论：将Deepseek-VL模型转换为 OpenVINO 格式后，推理速度显著提升，但准确度没有下降。→ 性能验证完成

### 6、个人感想

JUNHO JO:

在参与本项目的过程中，我深刻感受到技术与社会需求的紧密结合。作为一名工程师，我始终坚信技术能够在解决现实问题中发挥巨大的作用，尤其是在为视障人群提供帮助方面。这款产品的开发不仅让我在技术上得到了很多锻炼，也让我更加意识到科技如何改变人们的生活，尤其是如何为那些有特殊需求的人群提供更好的服务。

通过与导师和团队成员的合作，我们不仅学习到了最新的多模态大模型技术，还实践了如何将这些前沿技术应用到现实场景中。与Realsense深度摄像头的结合以及通过OpenVINO的优化，使得系统在高效性和实时性上都有了显著提升，达到了预期的目标。这让我对技术创新与实际应用的结合有了更深刻的理解。

最重要的是，我认为这款产品不仅仅是一个技术工具，它还是一座桥梁，帮助视障人士更好地融入社会，提升他们的生活质量。能够为这个群体提供帮助，是我参与这个项目最大的动力和收获。在未来的工作中，我将继续关注如何通过技术去服务社会，尤其是如何在人工智能和深度学习等领域，推动更多具有社会价值的创新。

LEE JEJUN:

通过本次项目，我不仅提升了深度学习模型的使用与优化能力，更重要的是，逐步建立了从技术实现到系统落地的完整开发思维。我不再只是停留在“模型调用者”的层面，而是学会了从用户需求出发，思考如何搭建可用、稳定、高效的人工智能辅助系统。

在项目实施过程中，我面对了多个复杂的技术挑战，例如模型量化中的精度损失、ONNX到OpenVINO转换时的图结构不兼容、推理性能瓶颈等。通过不断的调试与验证，我积累了宝贵的系统调优经验，也提升了我在复杂工程中的问题解决能力。

此外，通过与团队成员的沟通合作，我强化了自己的技术表达与团队协作能力，能够更清晰地传达设计逻辑、拆解工作任务，并高效地推动项目进展。整个项目经历极大提升了我将AI模型应用于实际场景的能力，也为今后从事智能系统研发打下了坚实基础。

KIM HOOJUN:

在参与本项目的过程中，我不仅提升了技术能力，也对人工智能技术在现实生活中的应用价值有了更深刻的理解。这个项目让我认识到，真正有意义的技术创新，不只是追求性能上的极致或模型指标的提升，更在于是否能够解决实际问题，是否能够为真正有需要的人群带来帮助。

项目中遇到的各种技术挑战也不断激发着我的学习热情。从系统架构设计、模型部署调优，到多模态交互方式的探索，我亲身参与了每一个环节，过程虽辛苦，但收获极大。我深刻体会到，真正的成长往往来自于不断跳出舒适区、面对未知领域时的主动思考与持续实践。

此外，作为团队的一员，我也深刻感受到高效协作与开放沟通的重要性。我们通过良好的沟通机制和明确的分工，有效推进了项目进展，这也让我对未来参与更大规模的工程项目充满信心。

最重要的是，这次经历让我明白，技术本身并非冷冰冰的工具，它可以成为推动社会进步的有力手段。在未来的学习和职业生涯中，我希望继续将AI技术与人文关怀结合，开发出更多真正有意义、有价值的智能系统，为建设一个更加包容和便利的社会贡献自己的力量。

### 7.参考文献

[1] DeepSeek AI. Deepseek llm: Advanced language model with 7 billion parameters. https://huggingface.co/deepseek-ai/deepseek-llm-7b-base, 2024.

[2] Intel Corporation. Openvino™ toolkit. https://docs.openvino.ai/, 2025. Accessed: 2025-04-26.

[3] Intel Corporation. Openvino™ toolkit github repository. https://github.com/ openvinotoolkit/openvino, 2025. Accessed: 2025-04-26.

[4] Shirly et al. Edward. Object identification for visually impaired. Indian Journal of Science and Technology, 9(S(1)), 2016.

[5] Ujjwal et al. Kadam. Hazardous object detection for visually impaired people using edge device. SN Computer Science, 6(7), 2025.

[6] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Victor Sanh, et al. Datasets: A community library for natural language processing. https://github. com/huggingface/datasets, 2021. Accessed: 2025-04-26.

[7] J. et al. Wiciak. A system for determination of areas hazardous for blind people using wave-vibration markers. Acta Physica Polonica A, 123(6):1101–1105, 2013.

[8] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Hug gingface’s transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

[9] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models, 2024.

[10] Roboflow. “How to Train YOLOv8 Object Detection on a Custom Dataset.” [Online]. Available: https://blog.roboflow.com/how-to-train-yolov8-on-a-custom-dataset/. Accessed: May 19, 2025.

[11] Featurize Team. “Featurize 文档.” [Online]. Available: https://docs.featurize.cn/. Accessed: May 19, 2025.

[12] OpenVINO Team. “Visual-language assistant using DeepSeek-VL2 and OpenVINO.” GitHub. [Online]. Available: https://github.com/openvinotoolkit/openvino\_notebooks/tree/latest/notebooks/deepseek-vl2. Accessed: May 19, 2025.

[13] Ultralytics. “Ultralytics YOLO Documentation.” [Online]. Available: https://docs.ultralytics.com/. Accessed: May 19, 2025.

[14] Pallets Projects. “Flask Documentation.” [Online]. Available:https://flask.palletsprojects.com/en/stable/. Accessed: May 19, 2025.

[15] Prompt Engineering Guide. “Prompt Engineering Guide.” [Online]. Available:https://www.promptingguide.ai/. Accessed: May 19, 2025.

[16] Google. “Open Images V7.” [Online]. Available:https://storage.googleapis.com/openimages/web/index.html. Accessed: May 19, 2025.

[17] bczhou. “Tiny-LLaVA-v1 Model.” Hugging Face. [Online]. Available:https://huggingface.co/bczhou/tiny-llava-v1-hf. Accessed: May 19, 2025.

[18] rany2. “Edge-TTS: Text-to-Speech with Microsoft Edge API.” GitHub. [Online]. Available: https://github.com/rany2/edge-tts. Accessed: May 19, 2025.