Project Proposal: Vision-Aware Smart Assistant

\*\*Student Name:\*\* Eric Christian

\*\*Course:\*\* E-104 Advanced Deep Learning

\*\*Topic:\*\* Home Assistant / Smart Home Integration with Deep Learning

\*\*Proposed Title:\*\* “Vision-Aware Smart Assistant: Real-Time Object Detection and Natural Language Voice Queries”

# Overview

This project proposes to develop a real-time computer vision system integrated into a smart home environment. Using a webcam and a deep learning model deployed on a local PC, the system will continuously detect and log objects in the home. It will summarize the detected objects periodically and expose the summary to a smart assistant platform (e.g., Google Assistant or Alexa) via Home Assistant.  
The system will allow users to ask in natural language, using a voice assistant like Google Nest Mini or Amazon Echo, what objects have been seen recently. The assistant will respond with a spoken summary derived from the model’s predictions, creating an interactive, vision-enabled smart assistant.

# Objectives

* Build a vision model pipeline using PyTorch and a pre-trained model (YOLOv5 or similar).
* Log detection results over time and generate human-readable summaries.
* Expose the summary to Home Assistant as a sensor value via REST API.
* Integrate with a voice assistant (Google or Alexa) to respond to natural language queries.
* Demonstrate a working prototype with video and voice interaction.

# Deep Learning Component

Use YOLOv5s or YOLOv5n (lightweight but accurate object detection models) implemented in PyTorch. Handle real-time inference from a webcam stream.  
Optionally, if required, I am prepared to fine-tune the model using a small, custom-labeled dataset to adapt to my home environment. This can include freezing the backbone and training only the head to reduce compute requirements while providing hands-on experience with transfer learning in PyTorch.

# Architecture Overview

1. Webcam Feed (PC with NVIDIA GPU)  
2. YOLOv5 Inference  
3. Label Logging (JSON or SQLite)  
4. Hourly Summary Generator  
5. Home Assistant Sensor Update via REST API  
6. Smart Assistant Query via Intent Script  
7. Spoken Response to User

# Technologies Used

* Python, PyTorch, OpenCV
* ultralytics YOLOv5
* pyttsx3 for TTS fallback
* Home Assistant + REST API
* Google Assistant (via Home Assistant intent scripts)
* Optional: Flask microservice for Alexa Skill endpoint

# Hardware Available

* PC with NVIDIA RTX GPU and webcam (primary inference device)
* Amazon Alexa Echo
* Google Nest Mini
* Google AIY Vision Kit
* iPhone/iPad (for testing interface)
* Linux/Windows laptops
* Optional IP cameras (Google Nest, Eufy)

# Deliverables

* Real-time detection demo (Jupyter or Python GUI)
* Integration with Home Assistant (sensor update + voice query)
* 1-page summary, MS Word report, working code, and test dataset
* PowerPoint slides with demo screenshots
* 2-minute and 15-minute YouTube videos

# Why It’s a Fit for E-104

This project demonstrates practical use of PyTorch and pretrained DL models, computer vision applied to smart home use cases, and integration of DL inference, automation, and voice-based interaction. It is a compelling real-world example of how advanced deep learning models can be deployed at the edge and consumed in a user-friendly way.

# Stretch Goals (Time Permitting)

* Run inference on AIY Vision Kit or Raspberry Pi with TFLite
* Track object frequency over days/weeks
* Add visual dashboard in Home Assistant
* Support multiple camera streams

# Summary

This project provides a meaningful and technically challenging opportunity to showcase how deep learning and computer vision can interact with consumer-grade smart home assistants. It emphasizes both the deployment of AI models and human-AI interaction via natural language.  
I respectfully submit this proposal for your approval.