Artificial Intelligence and Data Engineering

VISION-CHAT

INDUSTRIAL APPLICATION PROJECT

Andrea Bochicchio - Ivan Brillo - Filippo Gambelli

INTERACTION CAPABILITIES

Reactive Querying

Users can ask natural language questions like "**What is in front of you**?" or "**What happened in 1939?**". The system must provide answers based also on the current camera frame. Bilingual Support: **EN / IT**

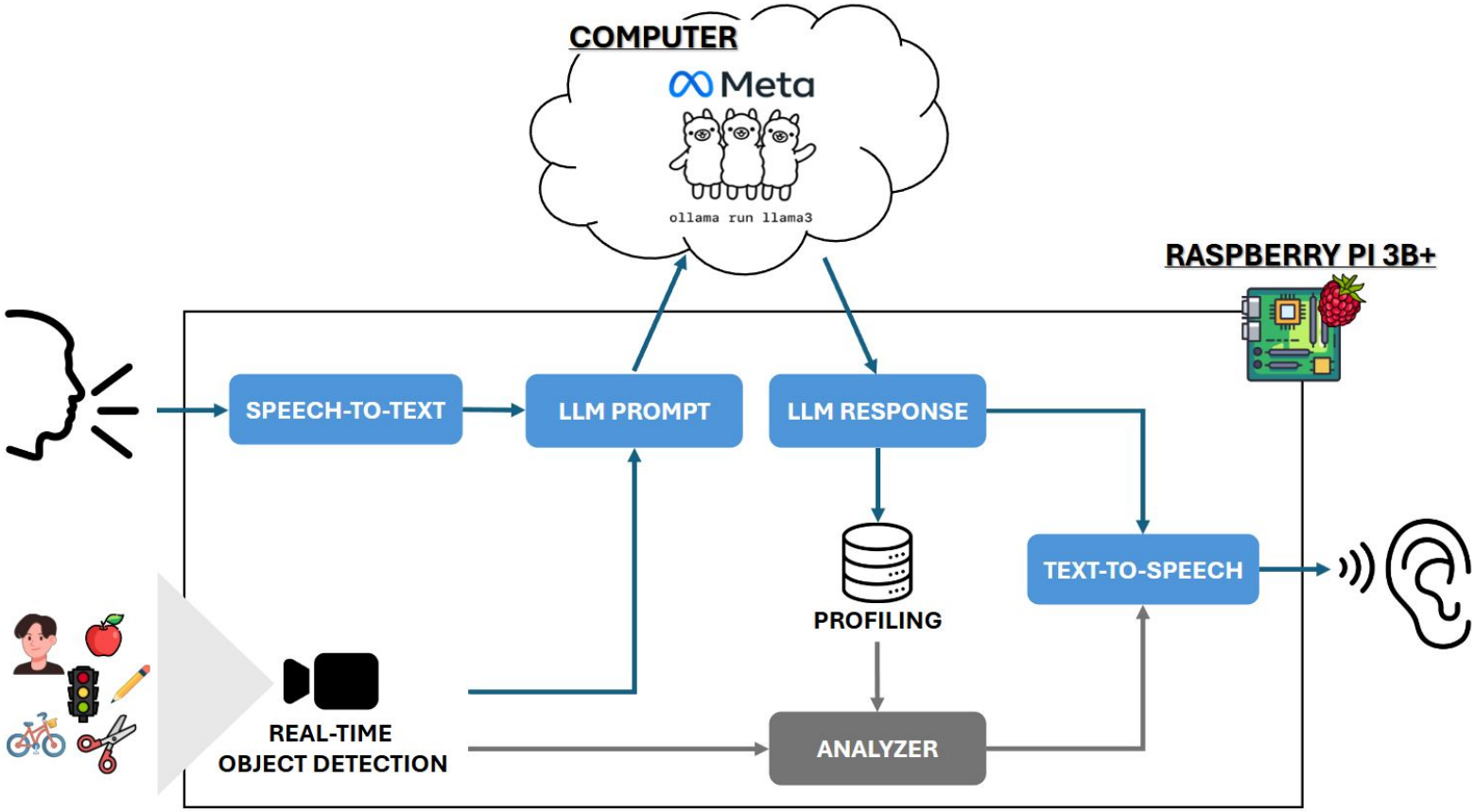
Proactive Profiling

The system builds an interaction profile dynamically. Users can say "**Notify me when you see a bottle**" or "**Notify me when you see a movement**" and the system stores this preference for future reference.

Automated Alerts

An integrated Analyzer module continuously monitors the video stream. If a detected object matches a user's preference, a spoken alert is triggered automatically. The same happen if a motion is detected on video.

SYSTEM ARCHITECTURE



SYSTEM IMPLEMENTATION

Main Process (P1)

- **Dedicated Process:** Owns the Flask web server and Voice Assistant instance.
- **Tasks:** Starts the separate vision process and connects them via multiprocessing queues.

Vision Subprocess (P2)

- **Dedicated Process:** Runs detection models without blocking the UI.
- **Tasks:** Captures Picamera2 frames, runs DNN object/motion detection, and encodes JPEGs.

Background Threads of P1

- **Alert Listener:** Daemon thread that continuously consumes the detection_queue.
- **Audio Callback:** PyAudio thread that buffers audio for the Vosk model.

THE VISION MODULE

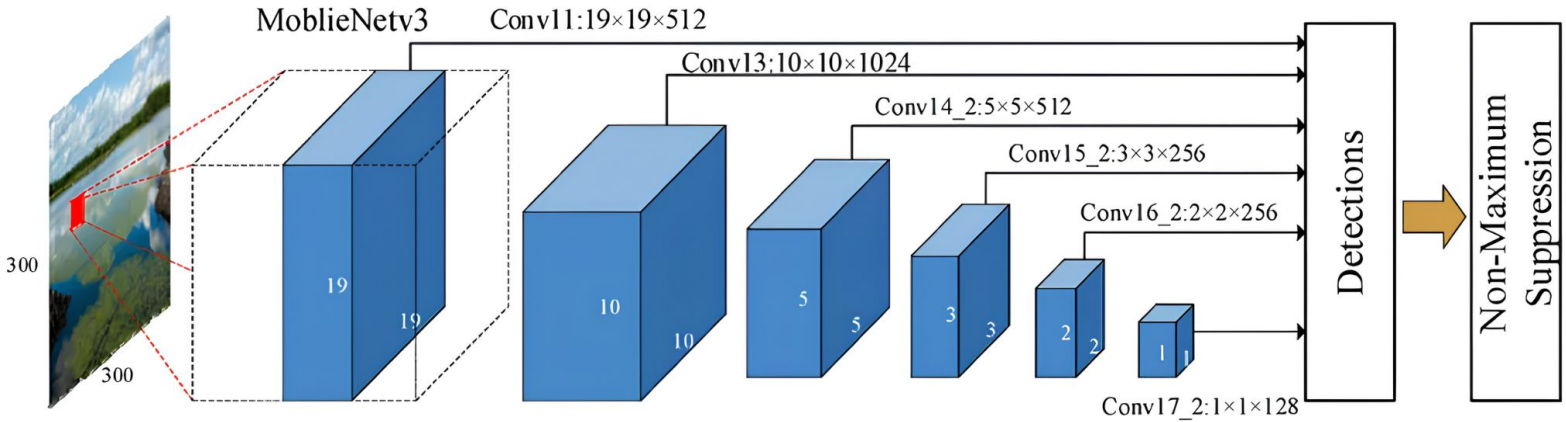
Model: SSD MobileNet V3 Large

We selected SSD MobileNet over **YOLO**-based architectures due to hardware constraints.

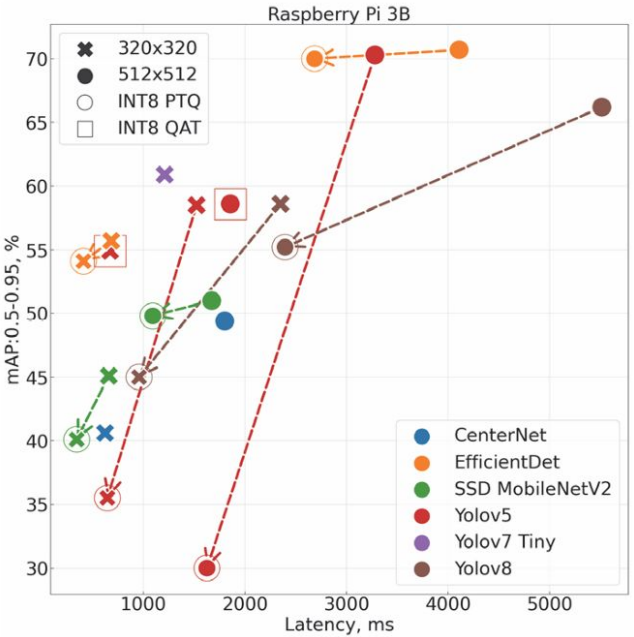
- While YOLO offers better precision, it reduces significantly the system's FPS
- We classify a frame every 3 to obtain a smoother video interaction

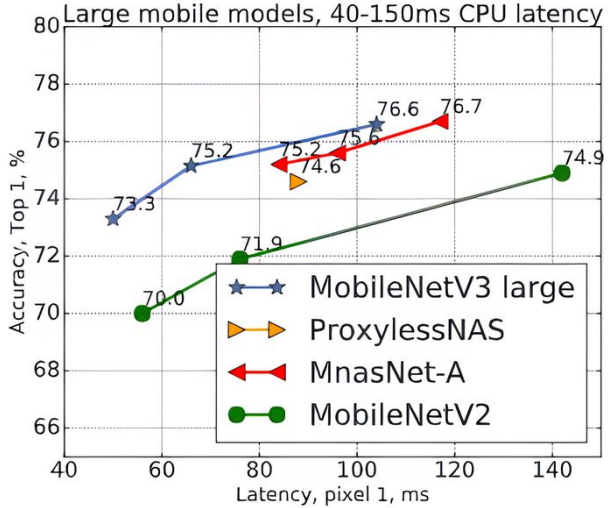
Current Performance:

~4.4 FPS (Average)



CNN COMPARISON





Source:

1. Zagitov, A., et al. "Comparative analysis of neural network models performance on low-power devices for a real-time object detection task." Computer Optics 48.2 (2024): 242-252.
2. Howard, Andrew, et al. "Searching for MobileNetV3". Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1314-1324.

THE LARGE LANGUAGE MODEL

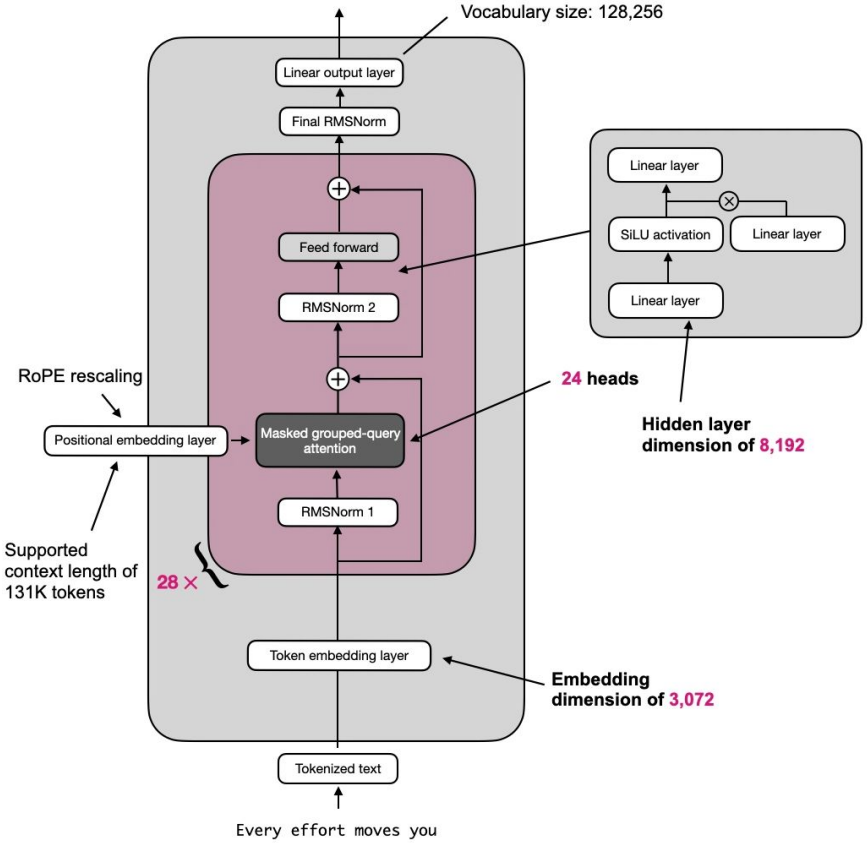
Model: Llama 3.2 3B

It is a lightweight, open-source language model optimized for efficient on-device performance.

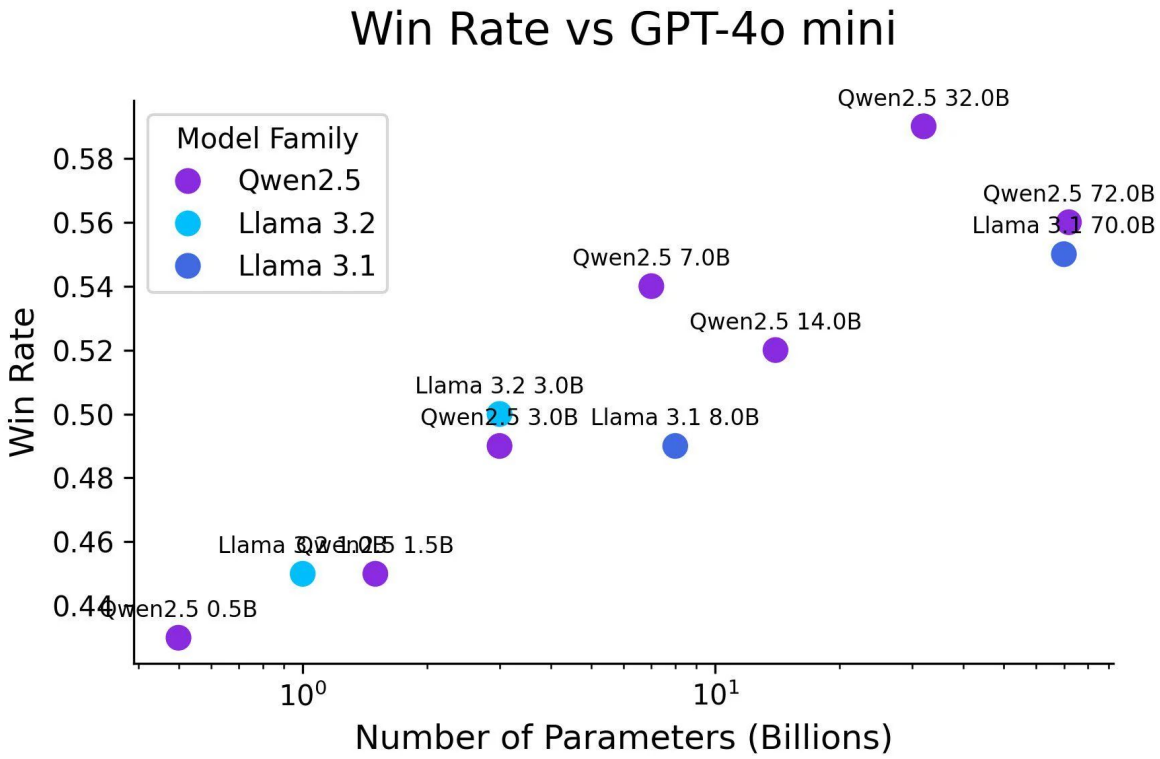
- Runs locally with low hardware requirements (no GPU)
- Fast and private inference
- supports multiple core languages out of the box

We also tried different other models, but they couldn't keep up with the velocity and precision of this one:

1. Llama 3.1 8B
2. Qwen2.5 3B
3. DeepSeek-R1 1.5B



TINY LLM COMPARISON



Source: Win Rate vs GPT-4o mini: Llama 3.2 vs Qwen 2.5. Retrieved from X @ArtificialAnlys

PROMPT ENGINEERING: CLASSIFIER

This strict structure enables prevents the LLM from chatting when it should set a preference, in particular:

1. Understand if the user is asking to be notified when a particular object appear
2. Or be notified if movement is detected

The prompt are translated in Italian if the language is set to IT

```
USER MESSAGE: {<user_text>}

TASK: Classify whether the user is asking for a future notification when objects appear or when there is movement.

DECISION RULES (FOLLOW ALL):
1. Set "is_alert_request" to true only if the user explicitly asks for a future notification.
2. The message must contain clear notification verbs such as: notify me, alert me, tell me when, let me know.
3. Questions about the current scene (for example, what do you see? or what is there?) are not alert requests.
4. If there is any doubt, set "is_alert_request" to false.
5. Do not guess or invent objects.
6. Extract target objects only if they are explicitly mentioned in the user message.
7. Set "is_motion_request" to true if the user asks for movement or motion detection.

OUTPUT FORMAT (JSON ONLY):
{
    "is_alert_request": true or false,
    "is_motion_request": true or false,
    "target_objects": ["object1", "object2"]
}

EXAMPLES:
User: "What can you see?"
Output: { "is_alert_request": false, "is_motion_request": false, "target_objects": [] }

User: "Notify me when you see a dog"}
Output: { "is_alert_request": true, "is_motion_request": false, "target_objects": ["dog"] }

User: "Alert me if a person appears"}
Output: { "is_alert_request": true, "is_motion_request": false, "target_objects": ["person"] }

User: "Do you see a cat?"}
Output: { "is_alert_request": false, "is_motion_request": false, "target_objects": [] }

User: "Notify me when there's movement"
Output: { "is_alert_request": true, "is_motion_request": true, "target_objects": [] }
```

PROMPT ENGINEERING: CONVERSATION

This strict prompt structure enables prevents the LLM from hallucinate or talk about visible objects even though not asked explicitly

The prompt are translated in Italian if the language is set to IT

```
HISTORY LABEL: {<conversation history>}

OBJECTS LABEL: {<currently visible object with position>}

USER MESSAGE LABEL: {<user_text}

INSTRUCTION: You must converse with the user while observing objects. Follow these rules strictly:
1. Always provide short and direct answers.
2. Answer general or common knowledge questions using only your general knowledge, without referring to visible objects or the visual context.
3. If the user asks questions such as: What are you seeing? Which objects are present in front of you? What objects are around you? refer exclusively to the objects listed above under CURRENTLY VISIBLE OBJECTS.
4. Never mention visible objects if the question does not directly concern them.
```

THE SPEECH-TO-TEXT MODULE

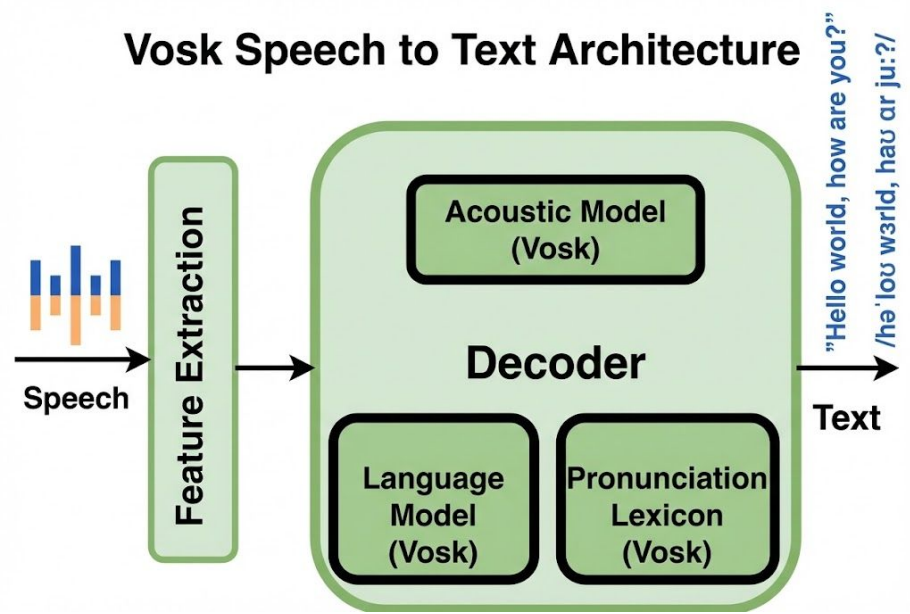
Model: vosk-model-small-it-0.22 or **vosk-model-small-en-us-0.15**

Vosk runs efficiently on CPUs and lightweight hardware, making it ideal for real-time decoding even on resource-constrained systems

- The streaming ability of Vosk has been fundamental to this project

We also tried a different family of models:

- **Whisper** often offers higher accuracy in benchmarks, but it generally needs more compute power and lacks true streaming



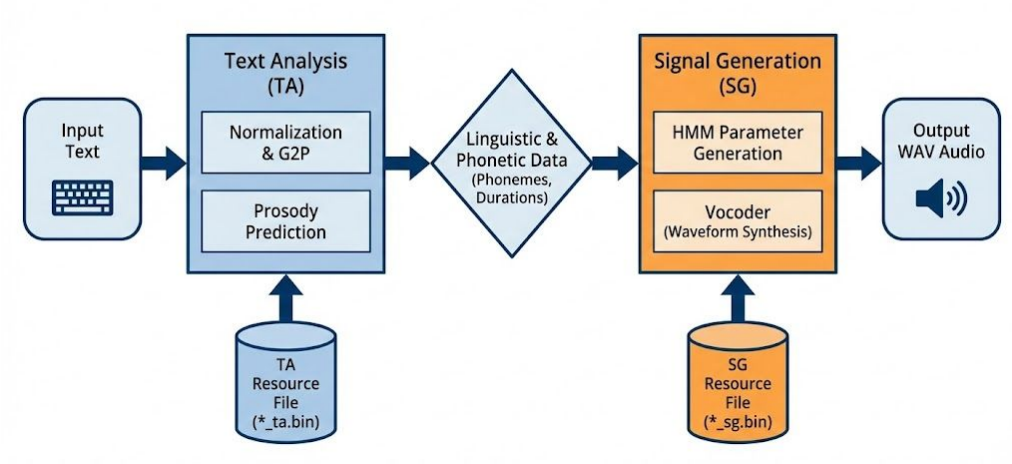
THE TEXT-TO-SPEECH MODULE

Model: pico2wave TTS

The engine underneath pico2wave is called SVOX Pico. Inside SVOX Pico, the architecture is split into two distinct parts: the text analysis and the signal generation.

- Works for Italian and English

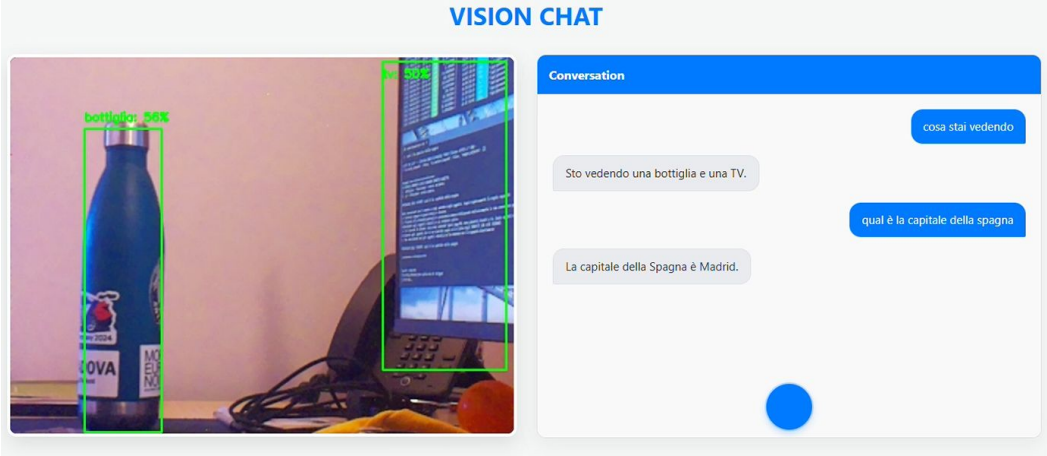
We also tried another small model for TTS called **Espeak**, but Pico2Wave produces a more natural, human-like voice



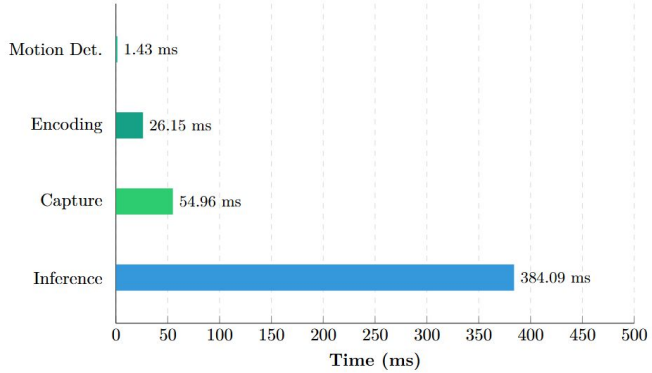
THE FLASK SERVER AND UI

Flask acts as a lightweight web interface for the Raspberry Pi:

- Allows the camera streaming with object detections and confidence
- Allows the start of a conversation via a button
- Allows to see the conversation’s history



PERFORMANCE EVALUATIONS



- The use of **JPEG encoding** reduces the latency of transmission

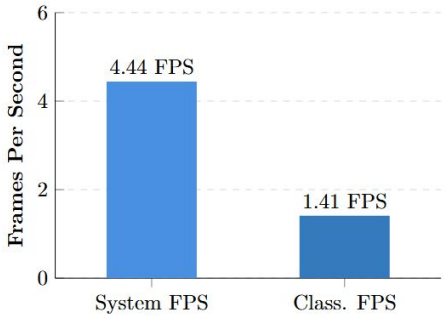


Figure 5.2: FPS Metrics: System vs. Classification

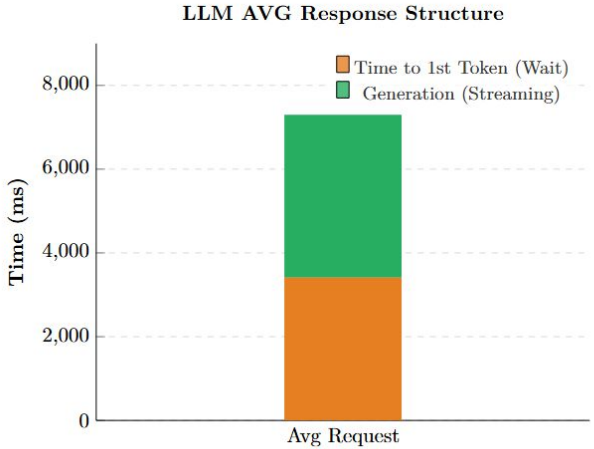


Figure 5.3: LLM Response Structure Metrics

- The use of **response streaming** allows the system to partially mask this latency
- The first query latency is mitigated by an initial **warm-up** query

PERFORMANCE COMPARISON

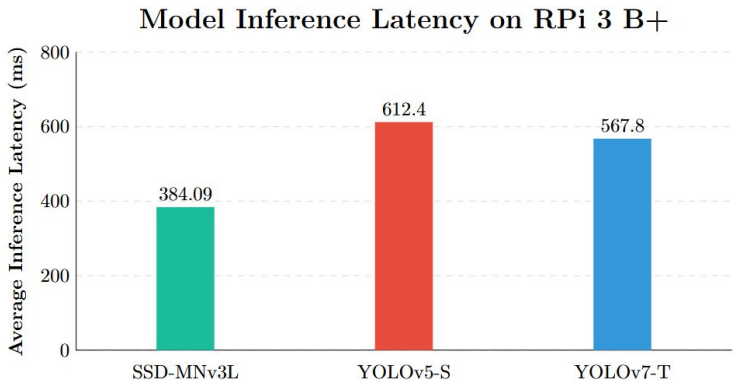


Figure 2.4: Average inference latency comparison in Raspberry Pi 3 Model B+. SSD MobileNet V3 Large (5.4M parameters), YOLOv5 Small (7.5M), YOLOv7 Tiny (6.24M).

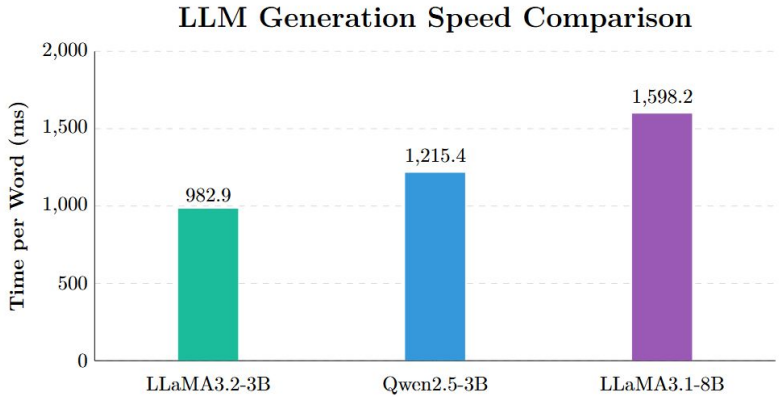


Figure 2.5: LLM time per generated word on a local system. LLaMA 3.2 3B achieves the lowest latency. Experiments conducted on an AMD Ryzen 7 8845HS CPU with 32 GB RAM and integrated GPU.