

PROJECT OVERVIEW

Abstract

Internet of Things (IoT) has enabled the development of smart buildings equipped with sensors to monitor occupant activities and environmental conditions. Despite the promise of these systems, they face significant challenges in achieving reliable performance across diverse environments due to their dependency on extensive labeled data, difficulty in generalizing across different settings, and the semantic gap between raw sensor data and meaningful, human-understandable information. This research aims to enhance the generalization and efficiency of smart ambient sensing systems through the development of the Maestro Testbed.

Methodology

Phase One:

- Image all Maestros with the proper dependencies from functioning Pis
- Cameras capturing video data of human activity
 - Automatic labeling

Phase Two:

- MAESTROS communicate with each other about what is happening in space using **zero-shot** or **few-shot** recognition.
- Create Dashboard for easy functionality

Phase Three:

- MAESTROS communicate with each other to **create a narrative** of given space
 - "memory" of the space
 - descriptions based on scope of time
- LLM is core of project

LOGURU



- The need:** better logging to help future developers understand errors, debug, and interpret the maestro experiments
- The solution:** Implementing Loguru, a library that enables human-readable logging.
- The impact:** Experimentation output is more clear and interpretable, useful for helping developers understand maestro output

```
2022-09-30 14:14:37.636 DEBUG   __main__:<module>:5 - A debug message.
2022-09-30 14:14:37.636 INFO     __main__:<module>:6 - An info message.
2022-09-30 14:14:37.637 SUCCESS  __main__:<module>:7 - A success message.
2022-09-30 14:14:37.637 WARNING  __main__:<module>:8 - A warning message.
2022-09-30 14:14:37.637 ERROR    __main__:<module>:9 - An error message.
2022-09-30 14:14:37.637 CRITICAL __main__:<module>:10 - A critical message.
~/dev/betterstack/community/demo/Loguru-demo via v3.10.6
```

MAESTRO FEATURES

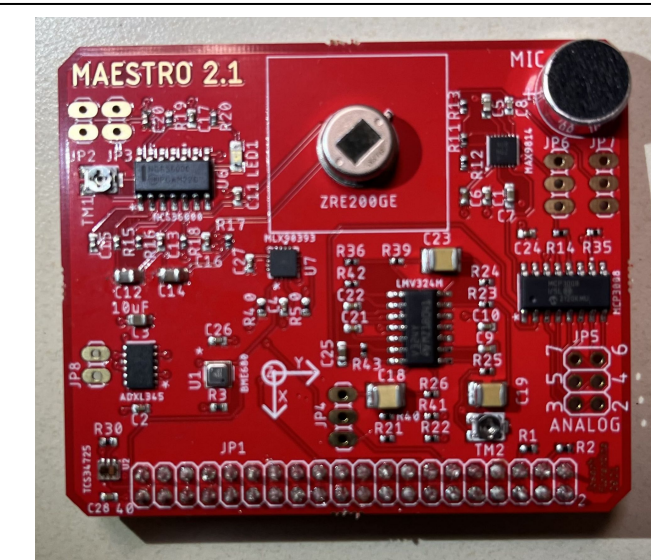
- Termination Control:**
 - Feature which allows user to control specific Maestro termination
 - Increases job efficiency by reducing runtime of the termination process
- Download Logs via API**
 - Allows the user to access Maestro-specific log files for better diagnosis of system health

```
for pi_num in rpis:
    i = rpis[str(pi_num)]
    status = end_maestro1(i)
    print(f"Maestro {i}: {status}")

@app.route('/clean_and_download', methods=['GET'])
def clean_and_download(log_file_path, log_folder):
    if not os.path.exists(log_folder):
        os.makedirs(log_folder)
    timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
    log_file_name = os.path.basename(log_file_path)
    backup_file_path = os.path.join(log_folder, f'{timestamp}_{log_file_name}')
    shutil.copy(log_file_path, backup_file_path)
    with open(log_file_path, 'w') as log_file:
        log_file.truncate(0)
    return print(f"Log file backed up to: {backup_file_path}")
```

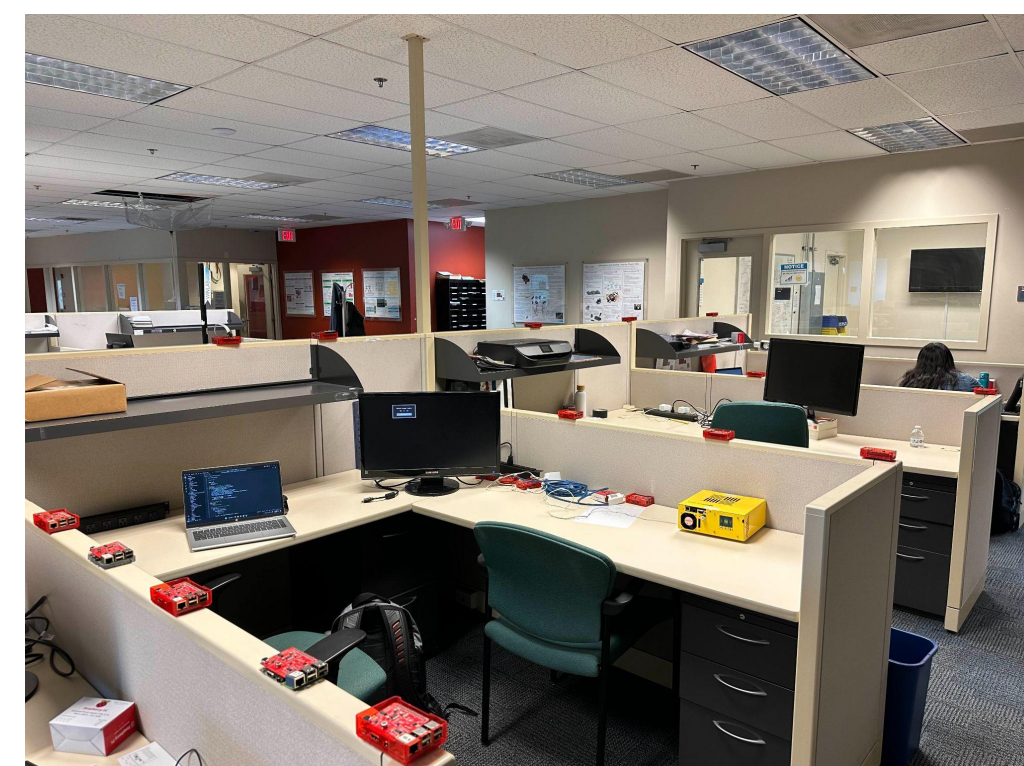
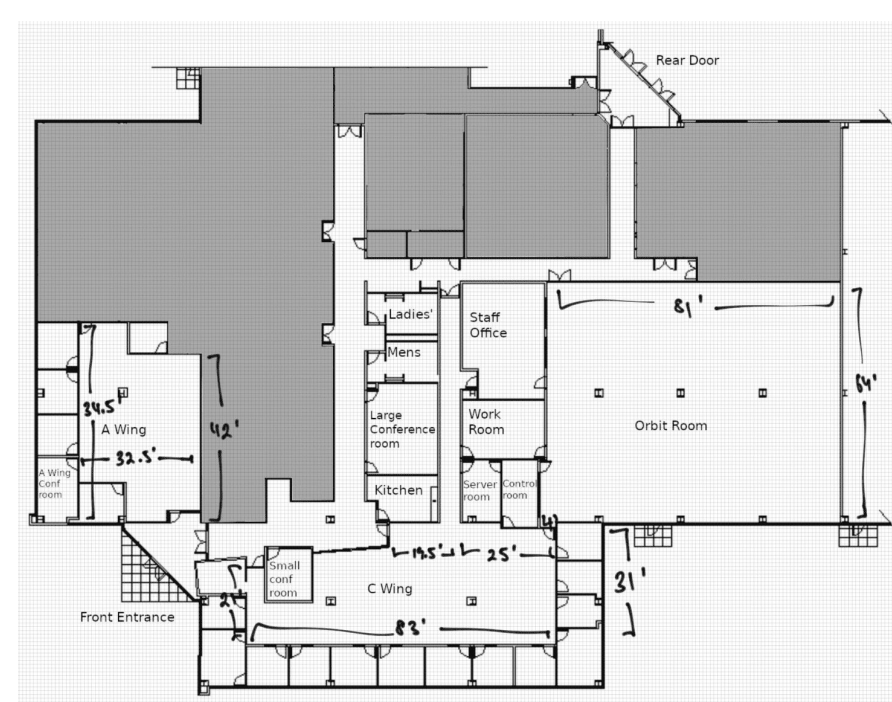
HARDWARE

Raspberry Pi Model 3B+: Microcomputer with Raspberry Pi OS Lite (Legacy)

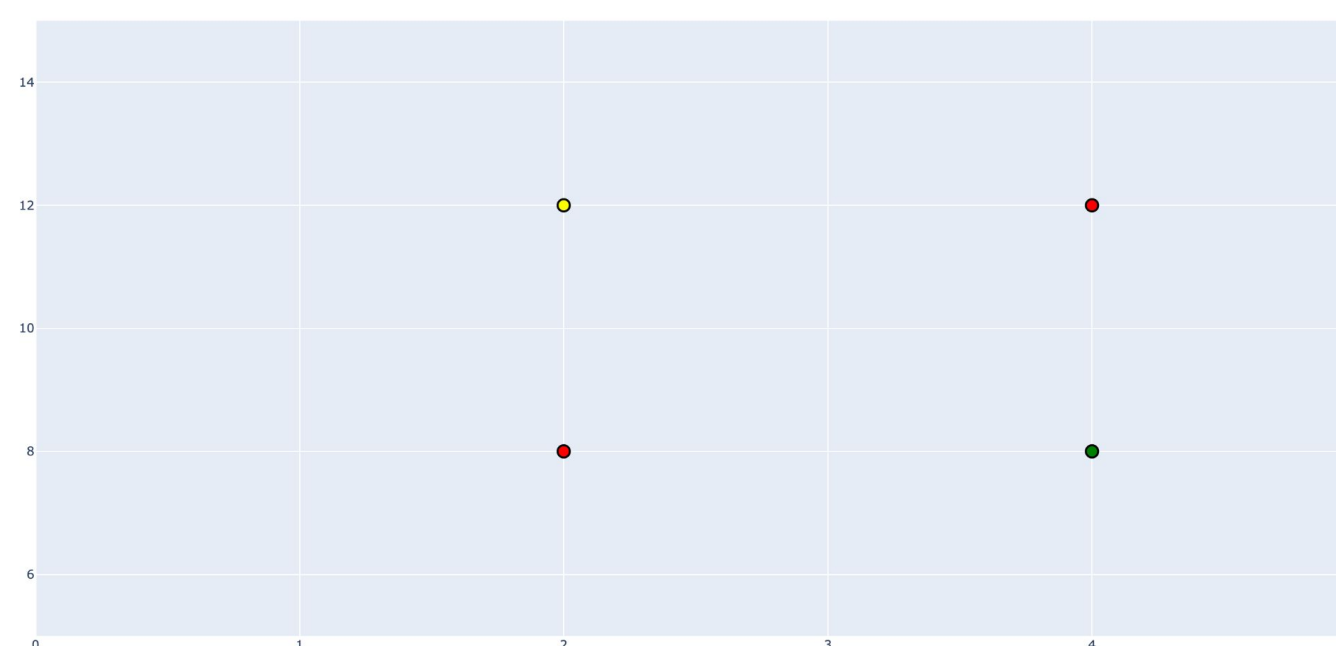


MAESTRO: a custom multi-modal sensor suite. Has NINE distinct sensors

COORDINATE GRID



- Grid layout of Maestros in the office space
- Color coordinated based on last time sensors were active



DASHBOARD

- 8 different** easy to use features
- Meant to make difficult and time consuming tasks such as running experiments possible through the click of a button

- First feature created: Shutdown Pis feature. Used to hard reset/hard shutdown all or select Pis in case of failed experiment
- Works using paramiko and "sudo shutdown -h now"

```
• Serving flask app "5002"
• Debug mode on
• [Start (timestamp)] This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
• Running on 127.0.0.1:5002
• [Press CTRL-C to quit]
• Restarting with PID 1
• Debugger is active
• Debugger PIN: 388-282-612
227.0.0.1 - [02/Aug/2024 23:37:11] "GET / HTTP/1.1" 200 -
327.0.0.1 - [02/Aug/2024 23:37:12] "[3]GET /favicon.ico HTTP/1.1" [b" 404 -
[3]Empty reply from server
Connecting to 38.61.2.134...
Failed to connect to 38.61.2.134: [Errno None] Unable to connect to port 22 on 38.61.2.134
Connecting to 38.61.1.155...
Failed to connect to 38.61.1.155: [Errno None] Unable to connect to port 22 on 38.61.1.155
Connecting to 38.61.1.156...
Failed to connect to 38.61.1.156: [Errno None] Unable to connect to port 22 on 38.61.1.156
Connecting to 38.61.1.139...
Failed to connect to 38.61.1.139: [Errno None] Unable to connect to port 22 on 38.61.1.139
Connecting to 38.61.1.136...
Failed to connect to 38.61.1.136: [Errno None] Unable to connect to port 22 on 38.61.1.136
Connecting to 38.61.1.140...
Failed to connect to 38.61.1.140: [Errno None] Unable to connect to port 22 on 38.61.1.140
Connecting to 38.61.1.148...
Failed to connect to 38.61.1.148: [Errno None] Unable to connect to port 22 on 38.61.1.148
Connecting to 38.61.1.154...
Failed to connect to 38.61.1.154: [Errno None] Unable to connect to port 22 on 38.61.1.154
Stopping empty 20
Connecting to 38.61.1.141...
```

GITHUB DOCUMENTATION

Installation of dependencies on a Pi to connect it to the testbed:

- Method 1: Imaging**
 - Fastest and most approachable way for Maestros setup
- Method 2: Building the Maestro From a Fresh Install**
 - Does not depend on having a physical microSD reader and allows for greater customization (e.g. package versions, no pre-existing settings, setup node for source experimentation, add only necessary components).

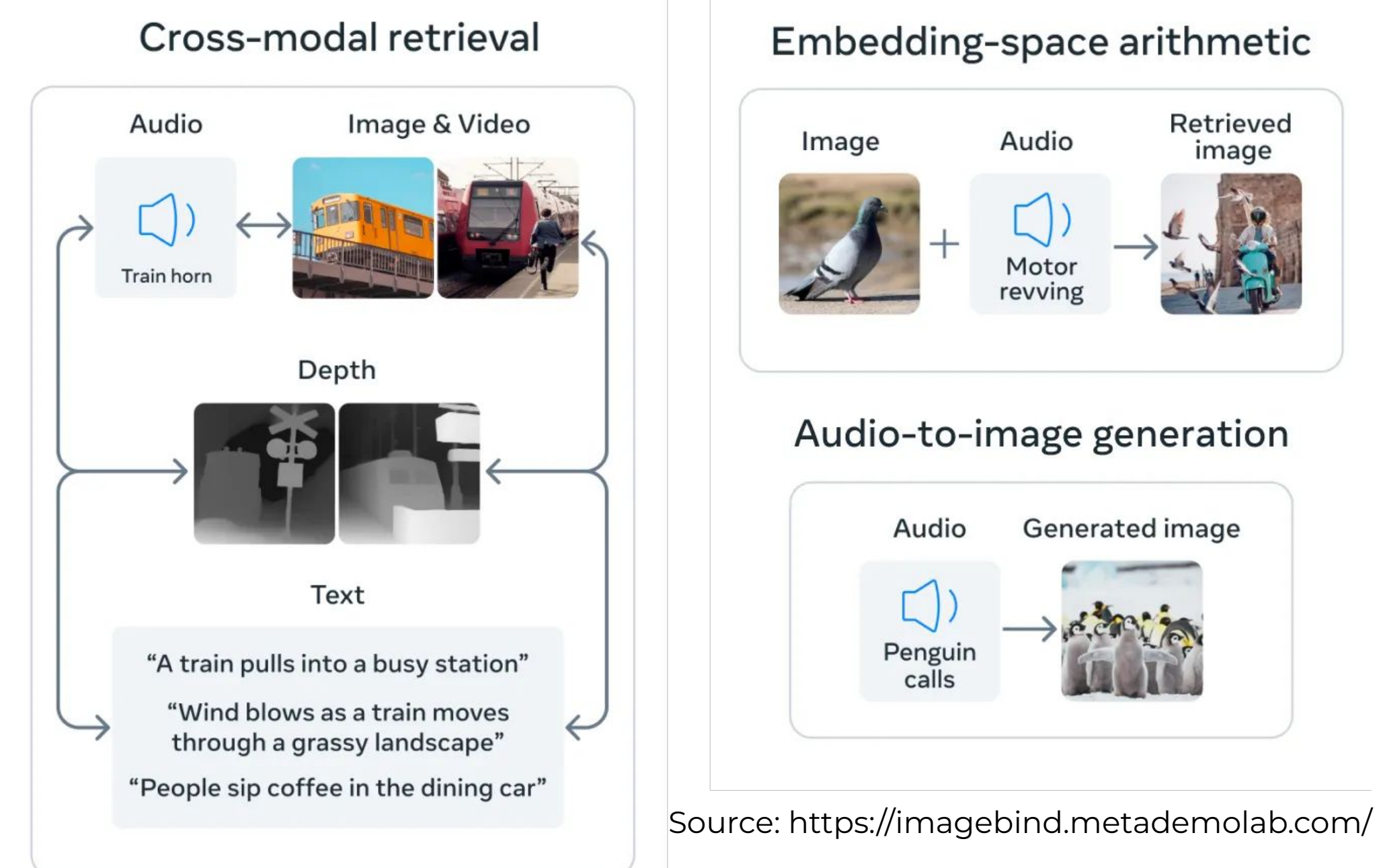
Visualization of streaming data:

- Issue with LED lights in the Raspberry Pi 3B+



FUTURE WORK

- Create another form of **visual appeal** for streaming data unless updates for managing LED lights in Raspberry Pi 3B+ are made.
- Get audio sensor data to stream to database
- Begin ML tagging of video and sensor data to allow for integration with LLMs



Automatic labelling: Label activity within environment using natural language descriptions of video data

- Train neural networks in encoder-decoder architecture for feature extraction
- Bridge the gap between sensor-to-text

ACKNOWLEDGMENTS

We would like to thank **Professor Jorge Ortiz** for being our mentor, as well as **Jenny Shane** and **Ivan Seskar** for suggesting advice for a few of our challenges. We would also like to thank our undergraduate students, **Matthew Grimalovsky**, **Emma Lin**, and **Sanchita Agrawal**, for guiding us through the entire research project process.