Thermal Imaging Application Documentation

Overview

The Thermal Imaging Application (TTIA) is a sophisticated web-based system designed to capture and display thermal images using a BeagleBone Black (BBB) connected to a Grove Thermal Camera (MLX9064x). The application provides a user-friendly interface to capture thermal images, view them in a carousel, and manage the saved images. The system is built using Python, Flask, and Bootstrap, with additional dependencies for interfacing with the thermal camera.

The application consists of the following components:

1. Backend: A Flask server that handles communication with the BBB, captures thermal data, and serves the web interface.
2. Frontend: A responsive web interface built with Bootstrap, allowing users to capture images, view them, and delete them.
3. Hardware Integration: Python scripts that interface with the Grove Thermal Camera via the BBB to capture thermal data.

System Architecture

1. Hardware Components

- BeagleBone Black (BBB): A low power, open-source single-board computer used to interface with the Grove Thermal Camera. The BBB runs a Linux-based operating system and provides GPIO, I2C, and SPI interfaces for connecting peripherals like thermal camera.
- Grove Thermal Camera (MLX9064x): A 24x32 pixel infrared thermal sensor that captures thermal data and sends it to the BBB. The MLX90640 is a factory-calibrated sensor with a digital I2C interface, capable of measuring temperatures between -40°C and 300°C. It has a refresh rate programmable between 0.5Hz and 64Hz, and it operates at a supply voltage of 3.3V.

2. Software Components

- Flask: A lightweight Python web framework used to create the backend server. Flask handles HTTP requests, serves the web interface, and manages communication between the frontend and the BBB.
- Bootstrap: A frontend framework for building responsive web interfaces. Bootstrap is used to create a user-friendly interface with a carousel for displaying thermal images, a capture button, and a thumbnail gallery.
- Paramiko: A Python library used for SSH communication between the host machine and the BBB. Paramiko allows the Flask server to execute commands on the BBB remotely, such as running the thermal data capture script.
- Matplotlib: A Python library used to generate heatmaps from the thermal data. Matplotlib processes the raw thermal data captured by the MLX90640 and converts it into a visual representation (thermal image).
- NumPy: A Python library used for numerical operations on the thermal data. NumPy is essential for reshaping the raw thermal data into a 24x32 grid, which is then used to generate the thermal image.
- Seed_mlx9064x: A Python library provided by Seeed Studio to interface with the MLX90640 thermal camera. This library is used to read thermal data from the sensor and send it to the BBB for processing.
- Subprocess: A Python module used for running external scripts (e.g., PC_capture.py). The subprocess module allows the Flask server to execute the thermal data capture script on the BBB.

Dependencies

1. Python Libraries

The following Python libraries are required to run the application:

- Flask: For creating the web server.
- Paramiko: For SHH communication with the BBB.
- Matplotlib: For generating thermal images from raw data.
- NumPy: For numerical operations on the thermal data.
- Seeed_mlx9064x: For interfacing with the Grove Thermal Camera.
- Subprocess: For running external scripts.

2. Frontend Libraries

The following Python libraries are required to run the application:

- Bootstrap 5: For styling and responsive design.

- Material Icons: For icons used in the interface.
- JavaScript: For dynamic behavior on the frontend

3. Hardware Dependencies

- BeagleBone Black: Must be connected to the same network as the host machine running the Flask server.
- Grove Thermal Camera (MLX90640): Must be connected to the BBB via I2C

File Structure

The application consists of the following files:

1. PC_capture.py:
   This script is responsible for capturing thermal data from the BBB. It uses the seeed_mlx9064x library to interface with the thermal camera and sends the captured data back to the host machine. The script establishes an SSH connection to the BBB, copies the capture.py script to the BBB, and executes it to capture thermal data.

2. Thermal_Imager.py:
   The main Flask application that serves the web interface and handles requests. It also manages the communication with the BBB via SSH. The Flask server provides routes for serving the web interface, capturing thermal images, and deleting saved images.

3. capture.py:
   A script that runs on the BBB to capture thermal data using the Grove Thermal Camera. It is copied to the BBB by the PC_capture.py script. The capture.py script uses the seeed_mlx9064x library to read thermal data from the MLX90640 sensor and prints the raw data to the console.

4. index.thml:
   The main HTML file for the web interface. It includes the carousel for displaying images, a capture button, and a thumbnail gallery. The HTML file uses Bootstrap for styling and includes JavaScript for dynamic behavior.

5. styles.css:
   Custom CSS styles for the web interface, including dark/light mode support and responsive design. The CSS file defines the layout and appearance of the web interface, including the carousel, thumbnails, and capture button.

6. scripts.js:
   JavaScript code for handling dynamic behavior on the frontend, such as carousel navigation, dark/light mode toggling, and image capture. The JavaScript code also handles the capture button click event, which triggers the thermal image capture process.

7. Run_Thermal_Imager(Windows).bat:
   A batch script for Windows users to start the Flask server and open the web interface in a browser. The script starts the Flask server and opens the default web browser to the local server address (http://127.0.0.1:5000).

8. Run_Thermal_Imager(Linux).sh:
   A shell script for Linux users to start the Flask server and open the web interface in a browser. The script starts the Flask server and opens the default web browser to the local server address (http://127.0.0.1:5000).

Setup and Installation

1. Prerequisites

- Python 3.x: Ensure Python 3.x is installed on both the host machine and the BBB.
- SSH Access: Ensure SSH access to the BBB is configured with the correct IP address, username, and password. To configure it use PC_capture.py.
- Network Configuration: Ensure the BBB and the host machine are on the same network.

2. Installing Dependencies

On the host machine, install required Python libraries using pip:

```
pip install flask paramiko matplotlib numpy seeed_mlx9064x
```

On the BBB, install the seeed_mlx9064x library:

```
pip install seeed_mlx9064x
```

3. Configuring the BBB

- Ensure the Grove Thermal Camera is connected to the BBB via I2C.
- Verify that the capture.py script can run on the BBB and capture thermal data

4. Running the Application

- Windows: Double-click the Run_Thermal_Imager(Windows).bat file to start the Flask server and open the web interface.
- Linux: Run the Run_Thermal_Imager(Linux).sh script to start the Flask server and open the web interface.

Usage

1. Capturing Thermal Images

- Click the "Capture" button on the web interface to initiate the thermal image capture process. The application will:
  1. Copy the `capture.py` script to the BBB. This is also how you can upgrade the code inside the BBB.
  2. Run the script on the BBB to capture thermal data.
  3. Process the raw data into a thermal image and save it in the *\Output* directory.

2. Viewing Images

- The captured images are displayed in a Bootstrap carousel. Users can navigate through the images using the carousel controls or the thumbnail gallery.

3. Deleting Images

- Users can delete images by clicking the trash bin icon on the thumbnail or carousel image. A confirmation dialog will appear before deletion.

4. Dark/Light Mode

- The web interface supports dark and light modes, which can be toggled using the icon in the header.

Troubleshooting

1. SSH Connection Issue

- Ensure the BBB is powered on and connected to the network.
- Verify the IP address, username, and password in the PC_capture.py script.

2. Thermal Camera Not Detected

- Ensure the Grove Thermal Camera is properly connected to the BBB via I2C. The default I2C pin is set to 1, therefore after installing grove libraries, you must set the correct I2C port from the library files. I2C port for BBB is Bus 2
- Verify that seeed_mlx9064x library is installed on the BBB.

3. Flask Server Not Starting

- Ensure all dependencies are installed on the host machine.
- Check for port conflicts (default port is 5000).