P. Suresh AP24122210002

G. Jaswanth Kumar AP24122210014

Report on Setting Up and Running the Real-Time Image Restoration Code

Introduction

This report outlines the steps necessary to set up and run the real-time image restoration code developed for the Hackathon titled "Real-Time Image Restoration in Automotive Systems." The application integrates sensor data with image processing techniques to provide a responsive dashboard for monitoring and intervention.

Project Structure

The project consists of several files organized in a folder. The key components include:

- **requirements.txt**: A file containing all the necessary libraries and dependencies required to run the code.
- **main.py**: The main Python script that executes the application, capturing images, processing them, and displaying results on a GUI.

Prerequisites

Before running the code, ensure that you have the following:

- Python 3.x installed on your system.
- Access to an Arduino or similar device for sensor data input.
- A webcam or camera connected to your system for image capture.

Installation Steps

1. Clone or Download the Project

Download the project folder containing requirements.txt and main.py. You can clone it from a version control repository or download it as a zip file.

2. Set Up a Virtual Environment (Optional but Recommended)

Creating a virtual environment helps manage dependencies specific to this project without affecting your global Python installation.

bash

```
# Navigate to your project directory
cd path/to/your/project
```

```
# Create a virtual environment
python -m venv venv


# Activate the virtual environment
# On Windows
venv\Scripts\activate
# On macOS/Linux
source venv/bin/activate
```

3. Install Required Libraries

Use pip to install all necessary libraries listed in requirements.txt. This file includes libraries such as OpenCV, NumPy, Tkinter, and others required for image processing and GUI functionality.

```
pip install -r requirements.txt
```

4. Configure Serial Port

Ensure that the SERIAL_PORT variable in main.py is set to match the port your Arduino is connected to. For Windows, it typically looks like COM8, while for Linux, it may look like /dev/ttyUSB0.

5. Run the Application

Once all dependencies are installed and configurations are set, you can run the application using Python:

```
python main.py
```

6. Using the Application

- Upon launching, the GUI will display sensor readings (temperature, humidity, light level) and images (original, noisy, restored).
- Click on the "Capture and Process" button to start capturing frames from your camera and processing them.
- The application logs data into a JSON file located in a timestamped folder under /logs, which includes sensor readings and processed images.

Data Logging

The application logs essential data such as:

- Sensor values (light level, temperature, humidity).
- Original, noisy, and restored frames.
- PSNR (Peak Signal-to-Noise Ratio) values for each processed frame.

Logs are saved in JSON format for easy analysis and can be found in:

Conclusion

This report provides a comprehensive guide to setting up and running the real-time image restoration application. By following these steps, users can leverage sensor data for enhanced image processing capabilities in automotive systems. The logging features implemented allow for further analysis of performance metrics over time, contributing to ongoing improvements in restoration techniques.