Developer's Guide to the Orion Model Introduction

Welcome aboard the Orion Model project. This system seamlessly integrates visual, IoT-based techniques, and virtual environments to provide a comprehensive monitoring and prediction system. This guide aims to bridge the gap between high-level architecture and code specifics, providing a structured pathway to understanding and contributing to the project.

Prerequisites

1. **Python Programming**: Python forms the backbone of this project.
2. **Machine Learning**: Familiarity with LSTMs, Regression models, and ML frameworks will be beneficial.
3. **Computer Vision**: Dive deep into the YOLO algorithm for object detection and pose estimation techniques.
4. **IoT**: Experience with IoT sensors is crucial.
5. **Data Clustering**: Knowledge of DBSCAN for clustering GPS data is essential.
6. **Unity & Virtual Environments**: Understand how Unity interfaces with the system, especially with **testServer.py**, to simulate and generate data.

Libraries and Tools

- **YOLOv7**: For human detection.
- **Kalman Filter**: Used in tracking and prediction.
- **LSTM**: Fundamental for sequence-based tasks in the IoT model.
- **DBSCAN**: Handles clustering of GPS data.
- **Standard Libraries**: Libraries like **numpy**, **pandas**, and **matplotlib** are commonly used.

Code Structure

The project is organized with each model in its respective folder, making them individual modules. These folders contain empty **__init__** files, allowing Python to recognize them as modules.

Directory Adjustments

A note for developers: Many directories in the code use local file paths, serving as placeholders. It's crucial to adjust these paths to point to the correct files or resources, ensuring seamless integration and operation.

Getting Started with the Code

High-Level Overview

1. **Visual Model**: Resides in files associated with computer vision. It handles detection, tracking, and prediction of human activities.
2. **IoT Model**: Processes data from IoT devices to perform monitoring, tracking, and predictions.
3. **Virtual Environment**: A Unity-based virtual environment generates simulated data, which is then sent to **testServer.py** for processing and integration with the main system.

Key Files and Their Roles

1. **HAR_train.py**: Focuses on training for human activity recognition.
2. **Co_Pred.py**: Manages collision prediction.
3. **Tracer.py**: Implements contact tracing.
4. **Dashboard.py**: Potentially a UI or dashboard for data visualization.
5. **MQTTManager.py**: Handles MQTT communications for IoT devices.
6. **TestServer.py**: Receives data from the Unity-based virtual environment and integrates it with the main system.
7. **HeartRateMonitor.py**: Monitors heart rates using IoT sensors.
8. **Tracker_run.py**: Records GPS coordinates and extracts time-space features.
9. **PMT_run.py**: Trains individualized LSTMs for prediction and analysis.
10. **Documentor.py**: A crucial file, assisting in generating documentation based on provided code. New developers should familiarize themselves with this for documentation purposes.
11. **PreReqGen.py**: Another vital script, it automates the generation of prerequisites based on the project's requirements

Datasets

There are two main dataset folders:

Clean Datasets: Use only these datasets for testing and training purposes. They've undergone preprocessing and are ready for machine learning tasks.

Raw Data Folder: A repository for new, unprocessed data. This data requires cleaning and analysis before usage.

Tips for Developers

1. **Begin with Architecture**: Familiarize yourself with the overall structure from the provided PDF.
2. **Understand Libraries**: Recognize the role and functionality of each library and tool.

3. **Communication**: Always be open to discussions and clarifications.
4. **Contribute Actively**: Start with small contributions like bug fixes or enhancing documentation.

Conclusion

The Orion Model is a multi-faceted system that integrates real-world data, simulated environments, and predictive analytics. This guide offers a comprehensive pathway for developers to understand, navigate, and contribute to the project. We're excited to have you on board!