REAL-TIME SURVEILLANCE SYSTEM - USER GUIDE

Overview

Welcome to the Real-Time Surveillance System, a sophisticated tool designed for users interested in harnessing the power of live camera stream analytics. This guide aims to provide a comprehensive walkthrough, ensuring a seamless and user-friendly experience.

1. Introduction

1.1 What is the Real-Time Surveillance System?

The Real-Time Surveillance System plays a crucial role in our surveillance infrastructure. Its primary purpose is to conduct real-time analytics on live camera streams. This system simplifies the process of video stream processing, allowing users to effortlessly specify a time period and retrieve the relevant footage.

1.2 Why is it Useful?

Designed with the user in mind, our system ensures accessibility for individuals with diverse technical backgrounds. The real-time interaction feature empowers users to request specific video footage based on timestamps and durations, providing a dynamic and tailored user experience. Intelligent storage mechanisms optimize resource usage, ensuring critical information is stored effectively without overwhelming the user.

2. Getting Started

2.1 System Requirements

Before diving into the Real-Time Surveillance journey, ensure your computer meets the following requirements:

- Operating System: Windows, macOS, or Linux
- RAM: 2 GB
- Python: Version 3.7 or higher installed

2.2 Installation Instructions

Let's get the system up and running:

- Download: Access the application from the GitHub repository.
- Extraction: Unzip the downloaded file to a location of your preference.
- Terminal/Command Prompt: Open the terminal or command prompt and navigate to the extracted folder.

- Now, run the Application by typing `python image_process.py` and press Enter.

2.3 Launching the Application

Upon launch, the application presents a straightforward interface displaying a live camera stream. It will record your live feed info and store the information into the Json file. It can be ended by pressing the 'q' key on keyboard.

3. Using the Application

3.1 Interacting with the Real-Time Surveillance System

The application interface provides a live camera stream. To retrieve video footage for a specific time:

- Timestamp Input: Enter the timestamp following the on-screen format.
- Duration Input: Specify the duration in seconds.
- Press Enter: Initiate the retrieval process.

- Note: To interact with the system, run user_interaction.py in a separate terminal or command prompt window. This allows you to input timestamps and durations for video retrieval.

3.2 Retrieving Video Footage

The system diligently processes frames, creating an output video file named "output.mp4" tailored to your specified timestamp and duration. patiently await the completion of the process. Might take time depending on the system capability.

3.3 Viewing the Output

The output video file is conveniently located in the application folder. Any standard video player on your computer can be used to view the footage.

4. Troubleshooting & Common Issues

If you encounter any issues, consider the following:

- Check Inputs: Verify the correctness of timestamp and duration inputs.
- While this project may not be entirely error-proof, it is designed to effectively run for a simple use case, meeting the requirements outlined in the PDF file.

5. Additional Information

5.1 File Structure

Explore the contents of the application folder:

- image_process.py: Main application file.
- user_interaction.py: User-driven interaction file.
- database_operations.py: Database-related operations file.
- frame_info.json: JSON file storing frame information.
- output_images/: Folder containing output frames.
- output_video.mp4: Output video file.
- batch_info.db: SQLite database file.

5.2 Tasks Not Completed

Certain tasks remain uncompleted due to time constraints:

- Concurrency and Performance: Concurrent processing of frames from multiple camera streams.

5.3 Acknowledgments

I extend My gratitude for the opportunity provided by HappyMonk to showcase my capabilities through this Data Science internship assignment. I have invested my best efforts and expertise within the stipulated timeframe. The code presented here is entirely my creation, with assistance sought from online resources to ensure its successful development. I trust that it aligns with the requirements outlined in the PDF document, and I eagerly await feedback from the HappyMonk team. Looking forward to the next steps in the process.

Best regards,

VISHAL