Task 3: Procrastination Preventer - Computer Vision and LLM Integration

Objective:

This task aims to develop a procrastination prevention system that utilises computer vision and Large Language Models (LLMs) to monitor and analyse a user's screen activity. The system will compare the user's actual screen content with their stated intentions, providing warnings when discrepancies are detected. This task will evaluate your skills in computer vision, natural language processing, and system integration.

Instructions:

1. User Input Processing:

- Develop a command-line interface to collect user input about intended activities (e.g., "I'm going to work with my coding editor and use my browser and terminal").
- Use an open-source LLM to parse and understand the user's intended activities, extracting key applications and purposes.

2. Screen Capture and Analysis:

- Implement a screen capture functionality that takes screenshots at regular intervals (e.g., every 5 minutes).
- Use computer vision techniques to analyse the screenshots and identify visible applications and content.

3. Content Classification:

- Develop a classification system that categorises screen content into "work-related" and "entertainment" categories.
- You have full liberty to choose your approach for content classification. This could involve using pre-trained models, developing custom models, rule-based systems, or any other method you deem appropriate for the task.
- Justify your chosen approach in your technical report.

4. LLM Integration for Context Understanding:

- Employ an open-source LLM to analyse the context of the captured screen content.
- Compare the LLM's interpretation of the screen content with the user's stated intentions.

5. Notification System:

- Implement a notification system that warns the user when detected activities deviate from stated intentions.

- Use system notifications or command-line outputs to alert the user.

6. Data Logging and Analysis:

- Log user activities, detections, and warnings for later analysis.
- Implement basic analytics to show productivity trends over time.

7. Privacy and Security:

- Ensure all screen captures and user data are processed locally and not stored or transmitted.
- Implement an option for users to pause monitoring or delete logged data.

8. Testing and Validation:

- Develop a comprehensive testing suite to validate the accuracy of the system.

Deliverables:

- Source Code: A GitHub repository containing all the code for the procrastination preventer system.
- README: A detailed README file explaining the setup process, dependencies, and how to run the system.
- Demonstration Video: A 5-10 minute video explaining the architecture of your solution, key challenges faced, and a live demonstration of the system in action.
- Technical Report: A PDF document outlining your approach, the models and libraries used, and potential future improvements.

Assessment Criteria:

- Functionality: The system should accurately detect and warn about procrastination activities.
- Integration: Effective combination of computer vision and LLM technologies.
- Code Quality: Well-structured, documented, and efficient code.
- Innovation: Creative approaches to solving challenges in real-time screen analysis and user intention matching.

Note: This assessment focuses primarily on backend functionality. While a basic command-line interface is required, an elaborate frontend is not necessary for this task.