Technical Report: AI-Powered Procrastination Preventer

1. Introduction

Procrastination is a common issue that affects productivity. This project aims to develop an **AI-powered system** to monitor user activities, detect potential distractions, and provide real-time alerts to help users stay focused on their intended tasks. The system integrates **computer vision** and **large language models (LLM)** for intelligent monitoring and analysis.

2. Approach

The system continuously captures and analyzes screen activities at regular intervals (default: every 5 minutes). It determines whether the user is engaged in productive tasks based on predefined categories and triggers alerts when distractions are detected.

Key Features:

- **User Intent Detection**: Users specify intended activities at the start.
- **Screen Capture & Analysis**: The system takes periodic screenshots and processes them using **OpenCV** and **image classification** techniques.
- **Text Classification with LLM**: Extracted text from screenshots is analyzed using an **LLM-based sentiment analysis model**.
- **Real-time Alerts**: If non-intended activities are detected, the system sends notifications using a **GUI pop-up**.
- **Activity Logging & Reporting**: Logs user activities and generates a productivity report.

3. Models and Libraries Used

The system utilizes the following models and libraries:

Machine Learning Models:

1. **Text Classification**: distilbert-base-uncased-finetuned-sst-2-english (Hugging Face Transformer pipeline)
2. **Computer Vision**:

- OpenCV (cv2) for screenshot analysis
- NumPy for image processing
- PIL (Pillow) for handling screenshots

Libraries Used:

Library	Purpose
torch	Runs the transformer model for text classification
transformers	Provides access to pre-trained LLMs
tkinter	GUI pop-up alerts for user notifications
cv2 (OpenCV)	Image processing and screenshot analysis
numpy	Numerical operations on images
PIL	Screenshot handling
json	Activity logging
threading	Multi-threaded execution
os	File management
time	Scheduling tasks

4. System Implementation

1. User Intent Input

Users input their intended work activities (e.g., coding, writing, research).

2. Monitoring & Screenshot Capture

- Screenshots are taken at regular intervals (ImageGrab.grab()).
- Screenshots are temporarily saved and analyzed.

3. Image Processing & Activity Classification

- Extracts text from screenshots (optional: OCR integration with Tesseract).
- Uses LLM for classification (work-related or non-work-related).

4. Alert System

- If an activity does not match user intentions, a warning pop-up appears using tkinter.

5. Logging & Reporting

- Logs all activities in logs/activity_log.json.
- Generates a final **Productivity Report** showing work time vs. distractions.

5. Potential Future Improvements

Short-Term Enhancements:

- **OCR Integration**: Use Tesseract-OCR to extract text from images.
- **Advanced Image Classification**: Implement deep learning-based scene recognition.
- **Customizable Alert Levels**: Users can define strict or relaxed monitoring.

Long-Term Enhancements:

- **Real-time Eye-Tracking**: Monitor user engagement through webcam-based tracking.
- **Voice Assistant Integration**: AI-powered assistant to encourage focus.
- **Adaptive Learning**: System adapts to user behavior over time using reinforcement learning.

6. Conclusion

The AI-powered **Procrastination Preventer** leverages **computer vision and LLM** technologies to help users stay productive. By continuously monitoring screen activities and classifying them based on user intentions, the system ensures a proactive approach to preventing distractions. Future improvements can further enhance its intelligence and adaptability.

Project Repository: GitHub - Procrastination Preventer

Developed by: Surabhi Chandrakant Bhor

Date: 6th February 2025