Assignment - Optimized Computer Vision System for Worker Activity Monitoring

Your task is to outline the complete system development process for a standalone vision-based solution that detects whether a worker is actively working. The system should be optimized for low hardware costs, be capable of running independently on boot, and meet industrial standards. Take any video from the internet of the worker working on the work floor.

Project: Worker Activity Monitoring System
Objective:
Develop a standalone vision system that determines whether a worker is actively engaged in a task. The system must run efficiently on low-cost hardware and be optimized for real-time performance. Upon power-up, it should auto-start and operate without human intervention.

1. System Architecture

Overview - The architecture includes:

- Camera Input: Captures real-time video stream.
- Core Vision Application:
  - Written in Python.
  - Uses computer vision and lightweight AI for inference.
  - Runs continuously on startup of system.
- Storage & Logging: Logs time-stamped activity events.
- Startup Service: Configured with system-level service (e.g., `systemd`) for auto-start.

Software Stack:

- Programming Language: Python 3
- Libraries: OpenCV, NumPy, PyTorch or TensorFlow (lite/ONNX variants)
- Optional Optimization Tools: ONNX Runtime, TensorRT, TFLite

2. Video Analysis and Activity Detection

Approach - The system analyzes live video to detect:

- Presence of worker
- Activity state (movement, pose, engagement)
- Idle/inactive state

Workflow:

1. Frame Capture:
   - Capture video frames in real time.
2. Preprocessing:
   - Resize and normalize frames.
   - Apply grayscale/background subtraction (optional).

3. Region of Interest (ROI):
    - Crop frames to relevant work areas.
    - Ignore background or irrelevant sections.
4. Detection & Classification:
    - Detects the worker using object detection or pose estimation.
    - Use heuristics or lightweight models to classify "working" vs. "idle":
        - Motion in ROI
        - Hand/tool movement
        - Body pose indicating active engagement
5. Decision Logic:
    - If activity is sustained, classify as "Working".
    - If static or absent for a period, classify as "Idle".

Code Repository & Resources

- Upload your full working code, including:
    - Model files (.onnx / .tflite)
    - Python script
    - Systemd config file
    - Instructions to follow

📂 **GitHub Repo:** https://github.com/yourusername/worker-monitor