Task Proposal: Jumbled Frames Reconstruction Challenge

Objective

Participants are provided with a 10-second, 1080p, 30 FPS video whose frames have been randomly jumbled.

The goal is to reconstruct the original video by restoring the correct frame order as accurately and efficiently as possible.

This challenge is designed to assess your problem-solving ability, algorithmic creativity, optimization skills, and practical understanding of performance and parallelism in code.

The Original Videos will be single shot videos with no cuts.

Task Description

- You will receive a short video file (jumbled_video.mp4) consisting of **300 shuffled frames** (30 frames per second × 10 seconds).
- Your task is to **write a program** that analyzes the frames and rearranges them into their **correct sequential order**, producing a video that closely matches the original.

Deliverables

Each participant must submit the following deliverables:

1. **Reconstructed Video**
    - Output file in .mp4 format showing the unjumbled video.
2. **Source Code**
    - Your complete source code in any programming language of your choice.
    - Include clear **instructions** (in a README.md or documentation file) on how to:
        - Install dependencies
        - Run the code
        - Test it using the provided evaluation video
3. **Algorithm Explanation (1–2 pages or README)**
    - Describe your approach and thought process:

        - What algorithm(s) or techniques you used (e.g., similarity metrics, AI/ML models, clustering, etc.)
        - Why you chose this particular method
        - Key design considerations (accuracy, time complexity, parallelism, etc.)

4. **Execution Time Log**
    - Log or report showing how long your algorithm took to reconstruct the video.
5. **Public GitHub Repository**
    - Create a public GitHub repository containing your project.
    - Perform **regular commits** (to show progress and development process).
    - Share the **Repository link and Reconstructed Video drive link** at the time of submission.

Rules and Guidelines

• Any programming language or framework is allowed.

• Use of **AI, ML, or image processing techniques** is permitted.

• You may utilize **multithreading** or **multiprocessing** for performance optimization.

• Ensure the program can be executed and tested on our evaluation system.

• The solution must be **your own work** — plagiarism or reuse of existing solutions will lead to disqualification.

Evaluation Environment

All submissions will be tested on the following benchmark system:

Processor: 12th Gen Intel(R) Core (T   M) i7-12650H (2.30 GHz)
Installed RAM: 16.0 GB
System Type: 64-bit operating system, x64-based processor

Your Code will be tested on a new Evaluation video and evaluated accordingly.

Thus, no use searching the original video online 😊

Evaluation Criteria

- Frame-wise similarity scores &Average similarity (%)
- Execution Efficiency (Processing Speed and Optimization)
- Algorithm Design and Innovation
- Code Quality and Documentation
- Explanation of Approach and Thought Process

Submission Instructions

Participants must:
- Create a public GitHub repository for their project.
- Commit code regularly with meaningful messages.
- Include a README file with clear setup and testing instructions.
- Submit the GitHub repository link before the deadline.

-Google Form will be shared for submission on the last day put your Repo Link and Reconstructed video Drive link in the submission.

JUMBLED VIDEO:

https://drive.google.com/file/d/1DbR9yap-vgUaPiI3hCEKUnniXr-TrdOt/view?usp=sharing