Assignment 2

In this assignment, we will perform several tasks relevant to video and audio media including Motion-Based hidden letter detection, audio extraction, and interlaced video simulation. You will work on the following file:

- video_with_letters.mp4 — a video that hides one letter per frame
- video_with_audio.mp4 — a short video that includes audio
- Both are found in this link https://drive.google.com/drive/folders/1I75xQcfYaZLRBCme-GIA_o11Capn9tJx?usp=sharing

PHASE 1: Hidden Letter Detection (Motion-Based)

Goal: Extract the message hidden in the video using motion detection.

You will:

- Read frames from video_with_letters.mp4
- Convert frames to grayscale
- Use **frame differencing** + contrast enhancement
- Apply **motion filtering**: skip frames with motion ratio > 5% as it would be too noisy
- Save binary difference masks to a folder motion_frames

Deliverables:

- The extracted message
- Code used to process and save the motion masks
- Screenshots of motion frames that clearly reveal the letters or the folder itself with the frames

PHASE 2: Audio Extraction

Goal: Extract the audio from video_with_audio.mp4 and save it as a .wav.2.2 Denoise Effect on Audio Write a Python script to apply a noise reduction effect on the extracted audio file.

Requirements:

- Read the audio file
- Apply a noise reduction filter to the audio file.
- Save the modified audio file

PHASE 3: Simulating Interlaced Video Scanning

Goal: Simulate the process of alternating scanlines in CRT monitors.

You will:

1. Load video_with_audio.mp4
2. Create:
    a. video_odd_interlaced.mp4: darken **even rows** (odd field)
    b. video_even_interlaced.mp4: darken **odd rows** (even field)
    c. Zoom effect: downscale by 25%, apply interlacing, upscale again
3. Extract the **first frame** from each video
4. Combine both into a side-by-side comparison
5. Display the combined image in Colab and save it as interlaced_frame_comparison.png

Deliverables:

- The two interlaced .mp4 videos
- A combined PNG image comparing the first frame from each version
- Code that displays this image inline using Matplotlib

What you need to submit:

- A working .ipynb notebook with:
    - All 3 phases implemented
    - Outputs shown inline (frames, audio, and interlaced comparison)
- A short PDF report that includes:
    - Extracted hidden message
    - Key screenshots (motion detection + interlacing)

- Denoised audio
- Brief summary of what each phase does

Submit a google drive link to this form and it must be via this form only:
https://docs.google.com/forms/d/e/1FAIpQLSdoV996rZ5r29QGLNi4RAdaNC7MYMoit6uMyHgXqRcBLgOSTA/viewform?usp=dialog

Bonus

- Create an animated **alternating scanline flicker effect** using odd/even frames

This is a team Assignment.
Same teams of two that submitted assignment 1 should submit assignment 2 together too.
Each team must submit one shared submission only.
If you have any comments, please communicate early.

Submission:

1. **Deliverables:**
   - Google Colab link
   - **.py** code file
   - Use matplotlib to show frames and images inside your notebook.
   - 2-3 page report (pdf) including your team info, discussion of the methods and results you used, and instructions of we can run your code
   - Output media saved
   - 1 minute video, every team member records himself/herself while discussing part of the implementation in 30 seconds.

*All deliverables must be uploaded to a **Google Drive folder**. Please ensure the sharing settings are set to **"Anyone with the link can view."** Include the Google Drive link in your submission*.

2. **Submission form:**
   https://docs.google.com/forms/d/e/1FAIpQLSeTeaQtWALSxK-neFUFCPgNqJv4dxhXvs-HdYLehiQPYbuv-w/viewform?usp=sharing

Deadline

The **deadline** is **Sunday 4th May, 2025**.

Good luck!