Submissions

⚠ **Group** : work by **pairs**.

⚠ **Deliverable:** your **pdf report** (**5 pages** at most) and the accompanying code (**a zipfile**).

⚠ **Submission:** by email to gaetan.leguelvouit@b-com.com.

⚠ **Deadline:** the 10th of Dec 2023 (11:59pm).

Expectations

This project requires a fair amount of **experimental work**. Please check that:

✓ Your report **describes your approach** in a **self-contained** manner (i.e., there must be no need to read other documents in order to understand your report).

✓ Your report explains precisely the **experimental environment** (e.g., programing language, hardware, external libraries, dataset) and the **experimental methodology** (e.g., number of repetitions, parameters).

✓ Your report **displays, describes, and thoroughly analyzes** the graphs you plot (e.g., the error according to the compression bitrate).

Indications

☞ We **strongly** suggest you to work with Python, but you are free to choose your favorite language if it is Python.

☞ This project will need FFmpeg[1] for video manipulation. We suggest to use one of its Python bindings: PyAV[2].

Your mission

You are now experts in image watermarking. You are ready to extend this knowledge to video watermarking in order to track movie piracy from the video-on-demand platform Nutflex. The approach is to assign a unique ID to each of **its millon (10^6) customers**. The watermarking process need to be done at the server side **using an A/B mechanism**. Each movie will be watermarked to get two versions: one containing the A symbol (e.g., 0) and the other the B symbol (e.g., 1). The data rate must be **1 bit for 2 seconds of video**.

Your mission might be split in the following tasks:

- **Build a movie dataset** downloading a dizain of movie trailers. Ensure to get the best video quality.
- **Implement** the algorithmic chain. (i) First, the embedding algorithm. Inputs are a movie file and a symbol, while output is a watermarked movie with similar video quality (check the video encoding rate). (ii) Next, the A/B mechanism. It should only manipulate compressed video frames (a.k.a. packets). (iii) The watermark reader.
- **Run** your experiments using the dataset as inputs and attacks. Attacks are limited to video re-compression, here H.264 compression from 2 Mbit/s (acceptable visual quality) down to 200 Kbit/ (poor visual quality).
- **Write** your report: describe and explain your choices, the results, the limitation of your approach and some ideas to go further.

[1] https://ffmpeg.org
[2] https://pypi.org/project/av/