Midterm Project: Image & Audio Integration

Python for Research — Universidad Católica de Oriente
Instructor: Juan Garzón

Overview and Constraints

For this midterm, you must select and complete one of the following projects. Both projects integrate image and audio processing, and they are designed so that you can complete them in about 6 working hours.

You are expected to design, implement, and document a complete application that produces meaningful results and shows your understanding of the concepts learned so far. The project should be creative, functional, and reproducible.

If you already have the knowledge and skills, I encourage you to go further and include machine learning techniques in your solution. Doing so is optional, but it can enrich your project and show your initiative.

The project must be submitted through Moodle, with the following structure:

- One video explaining the project and results (20 pts).
- A report in PDF format (uploaded separately).
- All code, media, and outputs in a compressed folder.

Project 1 — Multimedia Art Analyzer

Goal

In this project, you will extract image color features and audio timbral/rhythmic features, then present a unified artistic profile (plots and a short written interpretation) connecting sound and colors.

Learning Outcomes

- Image: color spaces (RGB/HSV), histograms, dominant colors (via k-means or histogram peak picking).
- Audio: RMS energy, spectral centroid (brightness), tempo/onset rate, mel-spectrogram basics.
- Fusion: normalization of heterogeneous features, clear visualization design, concise interpretation.

Suggested Tools

NumPy, Matplotlib, OpenCV/Pillow, Librosa, SoundFile or PyDub.

Inputs

- One image (JPG/PNG).
- One audio clip (15–45 s; WAV preferred).

Core Requirements

1. Image features (at least two): dominant colors (3–5) with RGB/HSV values and color swatches; and/or global color histograms.
2. Audio features (at least three): RMS energy, spectral centroid, tempo or onset rate; optionally band-limited energies.

3. Fusion and visualization: one figure or dashboard combining the extracted features from both media types.
4. Interpretation (maximum 200 words): relate color warmth/saturation to timbral brightness/energy.

Suggested Steps

1. Load media; compute histograms and dominant colors.
2. Load audio; compute RMS, spectral centroid, tempo; compute mel-spectrogram.
3. Normalize features; design a clear multi-panel figure.
4. Write the interpretation connecting both modalities.

Deliverables

- One video explaining the project and results.
- A report in PDF format (uploaded separately through Moodle).
- All code, media, and outputs compressed in a single folder.

Rubric (100 pts)

Correct feature extraction (image and audio)	35
Clear, well-labeled visualizations	20
Sensible normalization and fusion	15
Interpretation quality (evidence-based)	10
Video explanation	20

Project 2 — Image–Audio Synchronizer (Visual Equalizer)

Goal

In this project, you will produce a short animation or video where an image reacts to an audio clip. The image should be split into tiles or bars, and each region must change (brightness, size, transparency, or another visual parameter) according to the frequency-band energies of the audio over time.

Learning Outcomes

- Audio: framing, STFT/mel-band energies, smoothing and normalization.
- Image: tiling, per-tile transforms, frame synthesis.
- Video: writing frames to MP4 at fixed FPS; basic audio-visual synchronization.

Suggested Tools

NumPy, Matplotlib, OpenCV/Pillow, Librosa, SoundFile or PyDub.

Inputs

- One base image (JPG/PNG).
- One audio clip (20–40 s; WAV preferred).

Core Requirements

1. Band analysis: choose 6–12 frequency bands (e.g., mel bands); compute band energy per hop.
2. Mapping design: map bands to image tiles and define smooth visual transformations.
3. Animation: generate an MP4 (at least 10 fps, 10–20 s long). Interpolate energies for smoothness.
4. Synchronization: ensure visible reaction to beats or onsets.

Suggested Steps

1. Load audio; compute a mel-spectrogram; normalize and smooth band energies.
2. Tile the image (for example, 3x4 or 4x6).
3. For each frame: fetch band energies; map to tile parameters; apply transforms; compose the frame.
4. Write frames to MP4 with a fixed FPS. Optionally mux audio later.

Deliverables

- One video explaining the project and results.
- A report in PDF format (uploaded separately through Moodle).
- All code, media, and outputs compressed in a single folder.

Rubric (100 pts)

Correct band computation and normalization	30
Mapping design (clear, justified, smooth)	20
Visual quality and perceived synchronization	20
Code structure and performance	10
Video explanation	20

Academic integrity: cite any third-party media and avoid copyrighted materials without permission.