Classroom Lecture Video Analysis Model

User Instruction Manual

Andy Franck, Brendan Ng, Ben Fitzgerald, Zane Derrod

Version: v0.1

Hosted on Hugging Face: https://huggingface.co/your-repo

Generated: Monday 17th November, 2025

Contents

1 Overview

The Classroom Lecture Video Analysis Model automatically evaluates classroom lecture videos using the COPUS framework, generating standardized behavioral codes and summary reports without manual grading. This manual provides end users with instructions for installation, usage, and interpretation of system outputs.

2 Introduction

The Classroom Lecture Video Analysis Model is designed to automatically evaluate college-level STEM lectures using the COPUS (Classroom Observation Protocol for Undergraduate STEM) framework. Traditionally, COPUS coding required trained observers to watch an entire lecture and manually record instructor and student behaviors in two-minute intervals. This manual process is both time-consuming and labor-intensive, often requiring one to two hours of analysis per lecture.

The system automates this workflow with over 95% accuracy, using a vision-language model to interpret video segments and identify COPUS-defined behaviors. After processing, it automatically generates an Excel spreadsheet containing the full COPUS matrix for the lecture, along with additional report files for further analysis. This enables instructors, researchers, and evaluators to conduct large-scale or repeated analyses efficiently and consistently.

Scope

This user manual provides complete guidance for installing, configuring, and using the Classroom Lecture Video Analysis Model...

3 System Requirements

3.1 Hardware Requirements

- Recommended: 32GB RAM, NVIDIA GPU with 16GB+ VRAM
- Minimum: CPU-only mode supported but significantly slower

3.2 Software Requirements

- Python 3.9 or later
  - Download and setup tutorial: https://youtu.be/7GWWBywHhRo?t=9
- FFmpeg installed and added to PATH
  - Download and setup tutorial: https://youtu.be/eRZRXpzZfM4
- Windows, macOS, or Linux

4 Installation

4.1 Downloading the Software

1. Visit the repository: https://github.com/ajfranck/COPUS-ML
2. Click the big green **Code** button.
3. Select **Download ZIP**.

Figure 1: Navigation Command Example

4. Go to the ZIP file download location.
5. Select the ZIP and make sure to EXTRACT to a convenient location.

4.2 Setting Up the Environment

1. Open PowerShell (Windows) or Terminal (macOS/Linux).
2. Navigate to the `app/` folder:

   ```
   cd [paste path to extracted folder]/COPUS-ML-main/app
   ```

   Or see Figure 1 above:
3. Install dependencies with the following command:

   ```
   pip install -r requirements.txt
   ```

4.3 Verify Installation

```
python --version
ffmpeg -version
```

5 Preparing Your Video

5.1 Supported Formats

- MP4 (recommended)
- MTS (auto-converted to MP4)

5.2 Recommended Recording Conditions

- Clear view of instructor and students
- Stable camera position
- Minimal visual obstruction

6 Running the Application

6.1 Graphical User Interface (Recommended)

To launch the GUI type the following command (It will take a minute to appear):

```
python copus_gui.py
```

The interface includes:

- Video file picker
- Output directory selector

- Device selection (GPU Recommended for faster processing)
- Evaluation start button

Processing time varies based on video length and hardware speed.

6.2 Command Line Interface (CLI)

Basic Evaluation

```
python copus_evaluation_app.py lecture.mp4
```

Specify Output Directory

```
python copus_evaluation_app.py lecture.mp4 -o results/
```

CPU Mode

```
python copus_evaluation_app.py lecture.mp4 --device cpu
```

7 Output File Descriptions

The system generates three output files:

7.1 Excel COPUS Matrix

A human-readable summary including:

- 2-minute intervals across columns
- 24 COPUS behaviors across rows
- Binary values indicating presence/absence
- Totals, percentages, and summary metrics

7.2 JSON Output

Contains:

- Video metadata
- Sliding-window predictions
- Aggregated interval scores

7.3 Text Summary Report

Includes:

- Total behavior counts
- Percentage of intervals with each behavior
- Processing statistics

8 Troubleshooting

Common issues and fixes:

- **Out of memory:** reduce batch size or fps, enable '–cpu'.
- **Missing dependencies:** check 'pip install -r requirements.txt'.
- **Model download blocked:** set 'HF$_TOKEN$'$forprivaterepos$.

9 FAQ

Do I need a GPU? No, but it speeds up processing dramatically.

Can I use a non CUDA-enabled GPU? No, our model only works with Nvidia's CUDA-enabled GPU. Please select the CPU option if this is the case.

Can I change the 2-minute COPUS interval? No — COPUS protocol requires 2-minute bins.

Is internet required? Only on the first run when downloading the base model.

10 Appendix: COPUS Codes

Code	Description
L	Listening
Ind	Individual thinking
CG	Clicker group work
WG	Worksheet group work
OG	Other group activity
AnQ	Answering question
SQ	Student question
WC	Whole class discussion
Prd	Prediction
SP	Student presentation
TQ	Test/quiz
W	Waiting
O	Other
Lec	Lecturing
RtW	Real-time writing
FUp	Follow-up response
PQ	Posing question
CQ	Clicker question
MG	Moving/guiding
1o1	One-on-one help
D/V	Demo/video
Adm	Administration
W (Instructor)	Waiting
O (Instructor)	Other

10.1 COPUS Code Definitions

To avoid ambiguity in the COPUS parameters used in this project, we provide the following clarified definitions for FUp, RtW, Adm, and Lec.

FUp — Instructor Follow-up An interval is coded as FUp when the instructor provides verbal feedback, clarification, or discussion related to a question, activity, or idea previously posed by either the instructor or the students.

RtW — Instructor Real-time Writing An interval is coded as RtW when the instructor is actively and visibly writing or drawing in real time (using a marker, pen, stylus, chalk, or digital annotation tool) on a medium that is visible to students.

Adm — Instructor Administration An interval is coded as Adm when the instructor is performing logistical or procedural classroom tasks, such as distributing or collecting materials, managing resources, or addressing administrative matters.

Lec — Instructor Lecturing An interval is coded as Lec when the instructor is primarily engaged in continuous verbal delivery of course content. This includes explaining concepts, presenting new material, and providing extended descriptions without significant student interaction.

COPUS Article and Code Definitions On Page 3