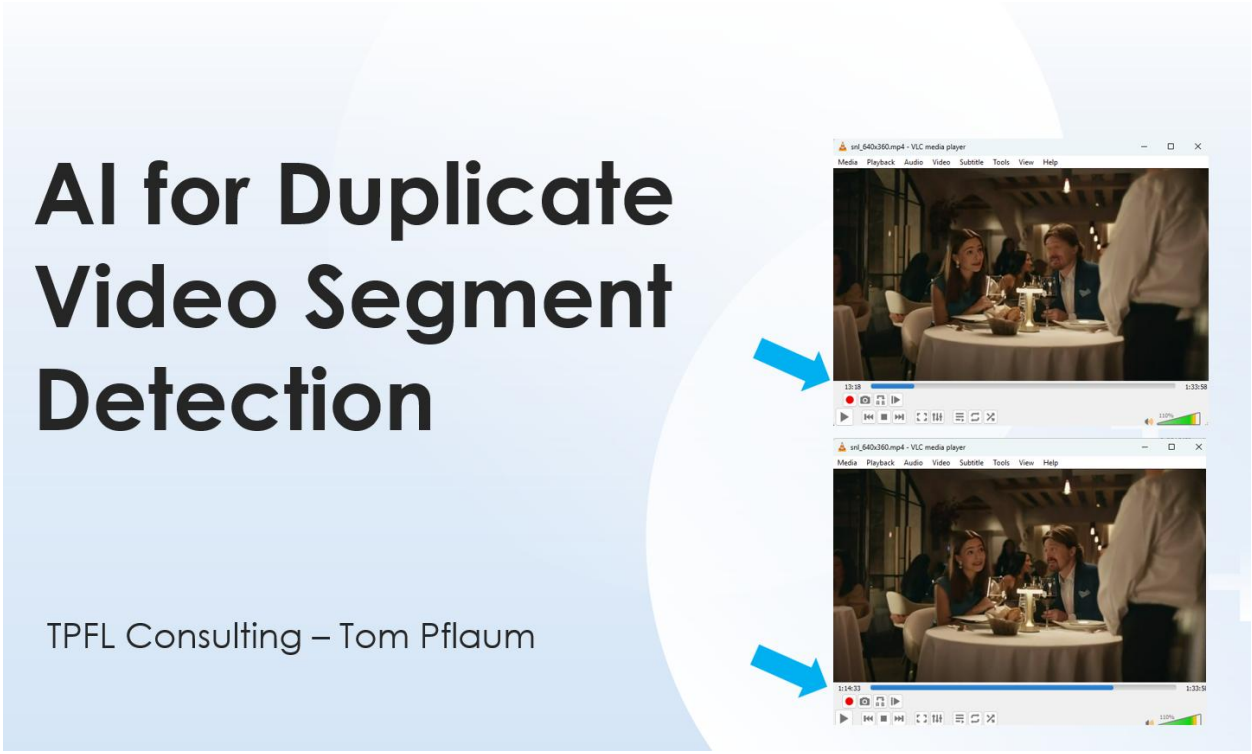


Post

I've been exploring how semantic embeddings can do more than just text-based video search — they can actually detect repeated video segments inside a single clip or across an entire archive.

Think repeated ads, promos, replay packages, reused B-roll, syndicated segments… all automatically identified using AI.

To show how accessible this is, I built an open-source workflow that finds repeated commercials in an over-the-air ATSC broadcast of SNL — running on a modest GPU, no enterprise stack required.

If you're working in broadcast, media engineering, AI/ML, or video analytics, this is a capability worth paying attention to.

Article

Detecting Repeated Video Segments with Semantic Embeddings

Modern video archives are growing at an unprecedented pace — from broadcast recordings and streaming libraries to surveillance footage and production assets. In my previous article, I explored how video embeddings enable semantic search using text prompts. In this follow-up, I focus on a closely related capability: identifying duplicate or repeated video segments within a single clip or across an entire archive.

This technique is far more practical than many people assume. It's not just a research curiosity or an overhyped AI demo - it's a real, accessible workflow that anyone can run today.

Why I'm Sharing an Open-Source Workflow

Before diving into the technical details, I want to highlight *why* I'm presenting an open-source implementation. Technologies like semantic embeddings, scene segmentation, and cosine-similarity search often get portrayed as futuristic or experimental. In reality, they're practical tools you can run on a laptop with a modest GPU.

By providing an open-source workflow, my goal is to show that repeated-segment detection is something any developer, researcher, or media engineer can experiment with cheaply and easily - no enterprise infrastructure required. This article concludes with a working implementation you can try yourself.

Why Detect Similar Segments?

Repeated-segment detection unlocks value across multiple industries. Here are the most impactful categories:

Media & Broadcast

- De-duping large video libraries
- Spotting syndicated segments reused across multiple shows
- Detecting commercials or promos that air multiple times
- Verifying ad delivery for compliance and billing
- Spotting reused B-roll or syndicated segments
- Tracking replay packages in sports broadcasts

AI/ML & Data Quality

- Deduplicating training datasets
- Finding visually similar scenes for retrieval-augmented generation
- Detecting repeated gestures or actions in behavioral datasets
- Clustering similar segments for downstream analysis

Security & Compliance

- Identifying repeated patterns in surveillance footage
- Detecting recurring individuals or objects
- Spotting repeated suspicious activity across long recordings

What Makes This Hard?

Traditional video-matching techniques struggle in real broadcast environments:

- Time shifts between repeated segments
- Compression artifacts distort pixel-level comparisons
- Noise and interference in over-the-air recordings
- Graphics overlays (tickers, bugs, lower thirds)
- Audio variations between airings

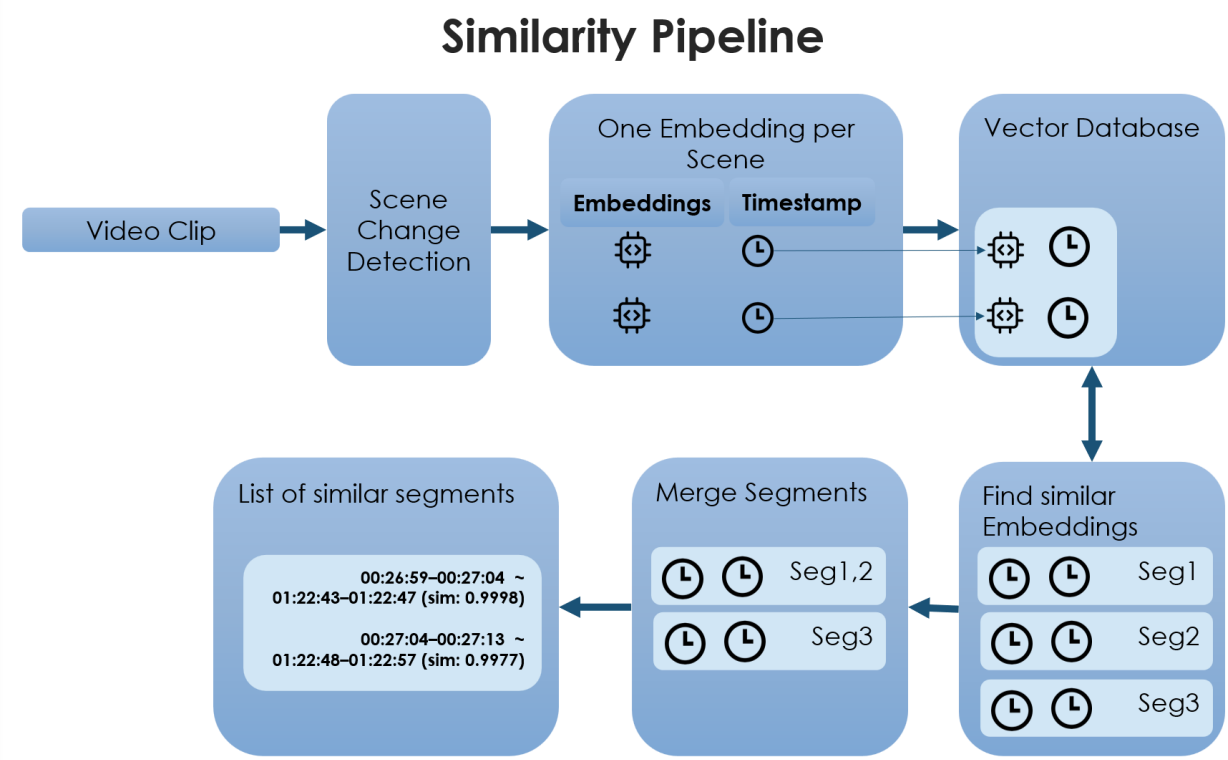
➔ Semantic embeddings solve these problems by focusing on meaning, not pixels.

The Approach – Semantic Embeddings

Modern multimodal models can generate dense embedding vectors that capture the semantic meaning of a video frame or segment. These embeddings enable efficient comparison between segments using cosine similarity — the same metric commonly used in semantic search and KNN-based retrieval systems.

Because similar or identical content produces similar embeddings, matching segments naturally cluster close together in the high-dimensional vector space.

Workflow Overview



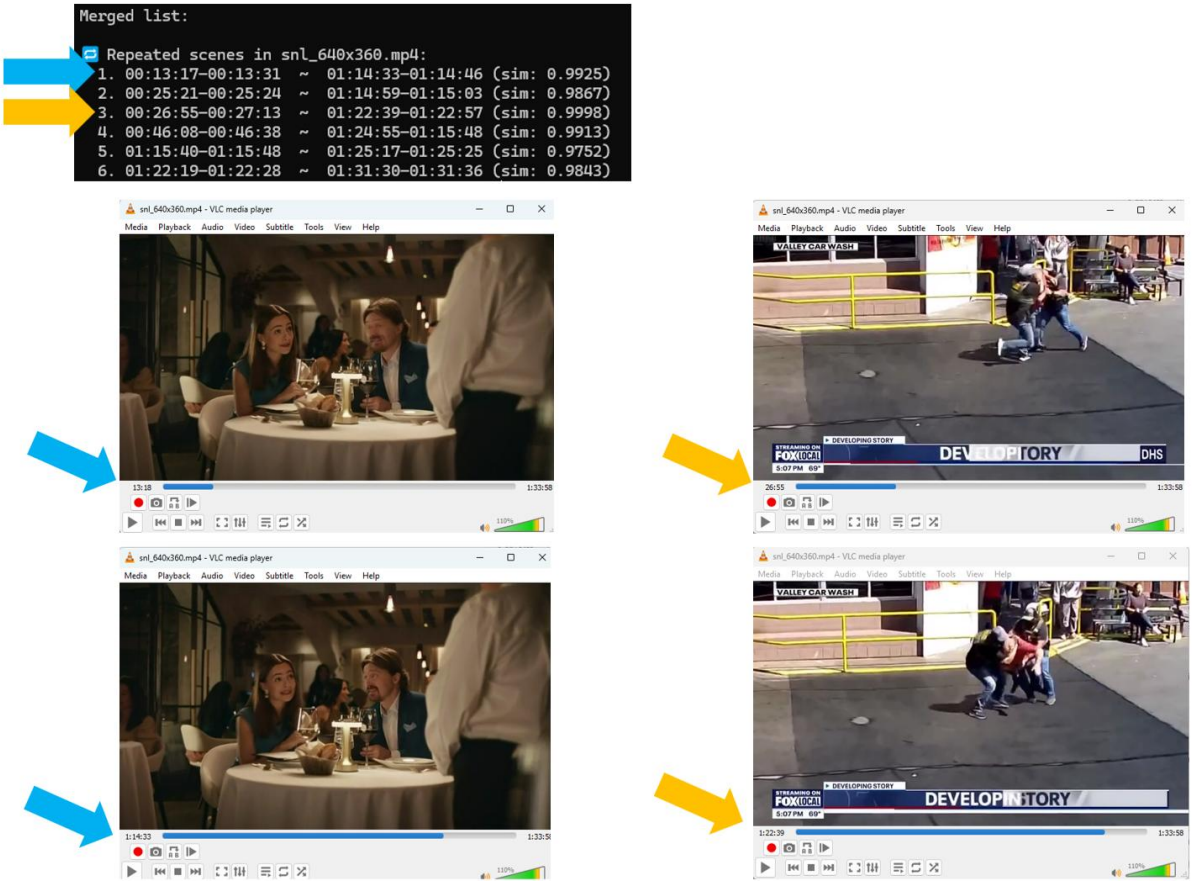
Here is the pipeline used in this article:

1. **Scene change detection** to break the video into segments
2. **Generate an embedding** from the midpoint frame of each segment
3. **Search for similar embeddings** using cosine distance
4. **Retrieve timecode ranges** of matching segments
5. **Merge adjacent matches** into longer repeated blocks

Example: ATSC Broadcast of SNL

I tested this pipeline on an over-the-air ATSC recording of an episode of *Saturday Night Live*. Despite broadcast noise, compression artifacts, and timing variations, the system successfully identified commercials and promos that aired twice during the show.

Semantic similarity is robust to all of that — making it ideal for real-world broadcast analysis.



Open-Source Implementation

To make this workflow accessible, I implemented the entire ATSC/SNL similarity-detection pipeline using an open-source model. Anyone can clone the repository and experiment with repeated-segment detection on their own video files.

Model

The sample code uses OpenAI's CLIP-ViT-B/32, a lightweight 151M-parameter vision embedding model. For context, this is far smaller than the 8B-parameter Qwen3-VL-Embedding model used in my previous article.

Scene Detection

Scene boundaries are detected using Color Histogram Correlation in HSV space, a fast and reliable traditional computer-vision technique.

Performance

The current (unoptimized) implementation runs at ~24× real-time speed on a desktop machine — fast enough for large-scale analysis.

GitHub Repository: tpflaum/CLIP_similarity

Why This Matters

As video libraries continue to grow, semantic embeddings unlock powerful new capabilities in content management, compliance, analytics, and AI workflows. They allow us to treat video not just as pixels, but as structured, searchable information.

Repeated-segment detection is just one example — but it's a compelling demonstration of how accessible and practical modern video-embedding technology has become.

#VideoAI #SemanticSearch #Embeddings #BroadcastTech #AdTech #OpenSourceAI #ComputerVision #VectorSearch #SimilaritySearch #PracticalAI #TechInnovation