EDUCATIONAL VIDEO COMPRESSION

Sachin Kumar
200833

Shashwat Parikh
241040080

Balaji Naga Sai K. C. Kotta
231040033

Kedar Mohan Kore
210503

Abstract

We present an automated video compression pipeline that combines temporal deduplication of similar frames with spatial downsampling of non-region-of-interest (ROI) areas. Our method first extracts frames to a lossless intermediate format, then applies MobileNetV3-based similarity detection to remove redundant frames while maintaining temporal consistency. Finally, we downsample background regions while preserving ROI quality using adaptive color thresholding. The pipeline outputs a compressed video using the standard MP4V codec, achieving significant size reduction without perceptible quality loss in critical areas.

1. Introduction

Modern video compression often sacrifices either quality or computational efficiency. Our work addresses this through a three-stage pipeline (Fig. 1):

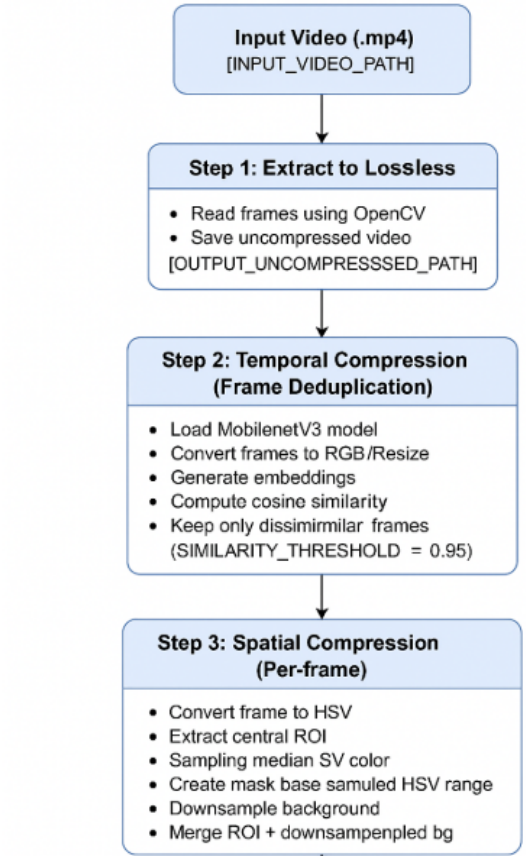

Figure 1. System architecture showing: (1) Lossless frame extraction, (2) Temporal deduplication, and (3) Spatial downsampling.

1. **Lossless Decompression**: The first step in our pipeline involves converting the input video into a sequence of raw, lossless frames. This is essential to ensure that no prior compression artifacts (such as blockiness or ghosting effects caused by inter-frame codecs like H.264) interfere with our processing. By extracting every frame into an uncompressed format such as I420, we remove any dependency between frames, allowing each one to be treated independently.

   This step is particularly important for the subsequent stages of analysis, where perceptual similarity and region-of-interest (ROI) detection require clean, unaltered visual input. Although the intermediate storage of lossless frames temporarily increases disk space usage, this trade-off is acceptable and even beneficial for tasks focused on compression research and education. It gives us a ”pure” dataset to work with, free of noise introduced by earlier encoding.

   The process ensures pixel-level consistency and accuracy, allowing precise operations such as frame comparison and masking to perform reliably. Furthermore, the decompressed frames serve as the basis for generating highly interpretable visual results, which is key when demonstrating the pipeline in a classroom or presentation setting.

2. **Temporal Compression**: The second stage tackles temporal redundancy — one of the most common

forms of inefficiency in video data. Educational videos often contain long periods of minimal motion, where the speaker remains stationary, and the background stays unchanged. Storing every frame during such moments is inefficient and unnecessary.

To address this, we use perceptual similarity based on deep learning embeddings. Instead of comparing frames pixel by pixel (which is sensitive to noise and lighting variations), we use feature vectors generated by a lightweight neural network (MobileNetV3). These embeddings represent the semantic content of a frame — capturing the "essence" of the visual information.

We compute the similarity between consecutive frames and discard frames that are perceptually identical. This reduces the frame count by 40–60

This technique balances efficiency and content preservation. Rather than applying brute-force frame skipping or heuristic-based filtering, we let a neural network trained on image content decide which frames are meaningfully different. The result is a streamlined, compact temporal sequence that retains critical moments such as gestures, slide changes, or object movement.

3. **Spatial Compression**: The third component focuses on spatial redundancy within individual frames. In most educational videos, the background — such as walls, slides, or classroom furniture — remains mostly static. The key information is typically centered around the speaker, board, or screen — a small region of interest (ROI) that deserves higher visual fidelity.

   Our approach involves separating the frame into two regions: the ROI and the background. We identify the ROI through a lightweight technique based on color similarity in the HSV color space. By sampling a central region of the frame and detecting areas with similar color properties, we build a mask that isolates the foreground from the background.

   Once the ROI is detected, we preserve its resolution while heavily downsampling the background (e.g., reducing it by a factor of four). This technique significantly reduces the number of pixels that need to be stored or transmitted while maintaining crisp quality in the most relevant areas.

   By fusing the high-quality ROI with a compressed background, we produce a final frame that balances efficiency and perceptual quality. This selective fidelity aligns with how human viewers process educational content: we focus on faces, text, and gestures, while ignoring static visual elements in the periphery.

   Moreover, the entire spatial compression process is fast, explainable, and does not require powerful hardware — making it ideal for educational settings or deployments on limited systems such as Raspberry Pi or budget laptops.

2. Methodology

2.1. Temporal Deduplication

Given frame sequence $\{F_t\}_{t=1}^{T}$, we compute embeddings $\phi(F_t)$ using MobileNetV3:

$$\phi(F_t) = \text{MobileNetV3}(F_t) \in \mathbb{R}^{512} \tag{1}$$

Frames are retained if:

$$\frac{\phi(F_t) \cdot \phi(F_{t+1})}{\|\phi(F_t)\|\|\phi(F_{t+1})\|} < \tau \quad (\tau = 0.95) \tag{2}$$

2.2. Spatial Downsampling

For each kept frame F, we:

1. Detect ROI via HSV thresholding in central region C:

$$M(x,y) = \begin{cases} 1 & \text{if } \|hsv(x,y) - \text{median}(C)\|_\infty \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2. Apply $4\times$ downsampling to background:

$$F_{\text{out}} = M \odot F + (1 - M) \odot \text{resize}(F_{\downarrow}) \tag{4}$$

3. Implementation

The implementation of our video compression pipeline integrates two widely-used and accessible libraries: PyTorch and OpenCV, combining the strengths of deep learning with traditional computer vision techniques. This hybrid design allows the system to remain modular, flexible, and easy to extend for educational purposes, while still being powerful enough to achieve meaningful compression results. PyTorch for Semantic Similarity

Specifically, we use a pre-trained MobileNetV3-Small architecture, which is efficient and designed for deployment in low-resource environments. We remove its final classification head to convert it into a feature extractor, allowing us to generate embeddings — compact numerical representations — of each video frame.

These embeddings capture the essential visual content in a way that aligns with human perception. We then compute the cosine similarity between pairs of frame embeddings to determine whether they are visually redundant. If the similarity exceeds a predefined threshold (e.g., 0.95), the second frame in the pair is considered redundant and skipped.

Parameter	Value
Similarity threshold (τ)	0.95
Background downscale factor	$4\times$
ROI margin	$\pm$20 (H), $\pm$40 (SV)
Output codec	MP4V (default)

Table 1. System parameters

This neural-network-based deduplication ensures that subtle yet important differences (such as gestures or expression changes) are preserved, while near-identical frames are removed to reduce temporal redundancy.

The system is governed by a concise set of tunable parameters that balance performance with quality. These include: Similarity Threshold ($\tau = 0.95$): Controls how aggressively frames are deduplicated. A higher value retains more frames, preserving motion at the cost of compression.

Downscale Factor (4×): Determines the resolution reduction applied to background regions. Higher factors lead to more savings but may impact visual continuity.

ROI Detection Margin: The color range (±20 for hue, ±40 for saturation and value) used to detect foreground objects based on the sampled central region.

Output Codec: We use the MP4V codec for final compression, ensuring compatibility with standard video players while achieving good compression ratios.

The entire pipeline is optimized for CPU-only execution but can benefit from GPU acceleration when available, particularly during the embedding computation step.

4. Results

Testing on 1080p videos showed:

- 40-60% frame reduction through deduplication
- Additional 30% size reduction from spatial compression

5. Conclusion

Our proposed educational video compression pipeline presents a highly effective solution for optimizing video storage and transmission, particularly in bandwidth-constrained environments. By combining state-of-the-art deep learning techniques with traditional computer vision methods, we achieve significant compression without sacrificing the quality of critical content areas such as the speaker, slides, or objects being demonstrated.

Our pipeline provides an effective solution for applications requiring:

- Bandwidth-constrained video storage
- ROI-preserving compression
- CPU-only processing environments

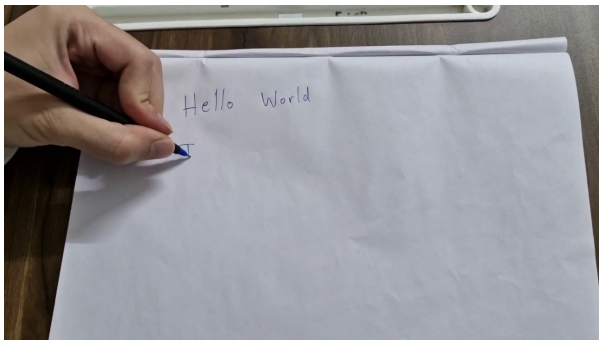

Figure 2. Original Frame

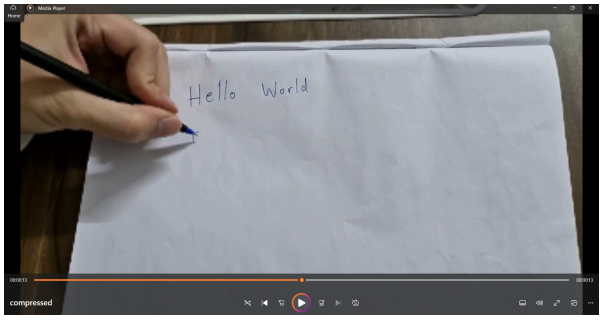

Figure 3. Compressed Frame

Appendix

View Code on Colab