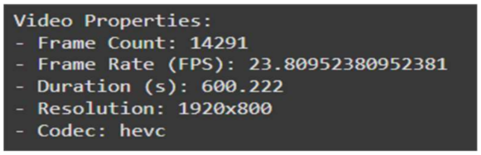
Video Compression

Introduction

Video compression is the process of reducing the file size of a video by using algorithms that eliminate redundant or unnecessary data. The goal of compression is to maintain video quality while minimizing storage and bandwidth requirements. In this report, we will focus on a video compression pipeline based on machine learning (ML) techniques, which aims to compress and decompress a video without losing significant quality.

1. Dataset Description

The dataset used in this project consists of video files and extracted frames. The video files are in MKV format, and the frames are extracted and categorized as keyframes. The goal is to develop a compression algorithm based on keyframe extraction, middle frame insertion, and decompression through interpolation, ensuring that the video quality is preserved while reducing file size.



```
Video Properties:
- Frame Count: 14291
- Frame Rate (FPS): 23.80952380952381
- Duration (s): 600.222
- Resolution: 1920x800
- Codec: hevc
```

Input Data:

- Video files in MKV format (AlitaBattleAngel.mkv).

- A set of keyframes extracted from the video using techniques like Optical Flow and Histogram Difference.

- The decompressed video generated by interpolating between keyframes and the added frames.

2. Methodology

Keyframe Extraction

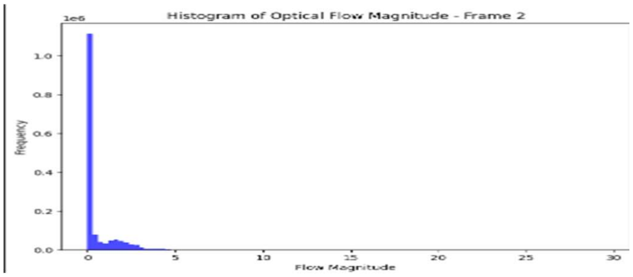
In this step, keyframes are selected from the video based on two techniques:

2.1. Optical Flow:

Optical flow is used to estimate the motion between two consecutive frames. This method detects motion and identifies significant changes between frames. If the motion detected by optical flow is above a threshold, the second frame is chosen as a keyframe.

2.2. Histogram Difference:

Histogram difference is used to compare the color distributions between two consecutive frames. A large difference in histograms indicates significant visual changes, which suggests that a new keyframe should be extracted at that point. By combining these two methods, we ensure that keyframes are selected at points of significant motion or visual changes, which are important for efficient compression.



3. Compression:

The compressed video consists of the keyframes and the middle frames. By keeping only these frames and removing unnecessary intermediate frames, the file size is reduced.

4. Decompression:

The decompression process involves using the original keyframes and interpolating the middle frames between them to recreate the video.

5. Models and Techniques

The models and techniques used in this process are:

- Optical Flow:

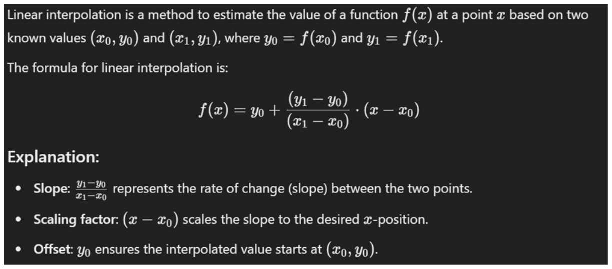
Optical flow is a technique used to calculate motion between two consecutive frames in a video. This technique uses pixel intensity values and calculates motion vectors that represent the movement of objects in the scene. The method is implemented using OpenCV.

- Histogram Difference:

This technique compares the color histograms of two consecutive frames to identify significant changes. If the difference is above a certain threshold, a keyframe is selected. This method is useful in detecting scene changes and ensuring that keyframes are extracted at the correct moments.

- Linear Interpolation:

Linear interpolation is used to generate intermediate frames between keyframes. This involves interpolating pixel values from the two keyframes to create a new frame that lies between them.



Linear interpolation is a method to estimate the value of a function $f(x)$ at a point x based on two known values (x_0, y_0) and (x_1, y_1), where $y_0 = f(x_0)$ and $y_1 = f(x_1)$.

The formula for linear interpolation is:

$$f(x) = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)} \cdot (x - x_0)$$

Explanation:

- **Slope:** $\frac{y_1 - y_0}{x_1 - x_0}$ represents the rate of change (slope) between the two points.
- **Scaling factor:** $(x - x_0)$ scales the slope to the desired x-position.
- **Offset:** y_0 ensures the interpolated value starts at (x_0, y_0).

6. Analysis

Keyframe Extraction Accuracy

- Optical Flow:

Optical flow successfully identifies regions of significant motion. However, its performance depends on the threshold for motion detection. A higher threshold can miss subtle motion, while a lower threshold can generate too many keyframes.

- Histogram Difference:

Histogram difference effectively identifies significant visual changes. The method is particularly useful for detecting scene changes and ensuring that keyframes are extracted at the correct moments.

Compression Performance

Visual Quality:

The quality of the decompressed video is evaluated by visually comparing it to the original video. The added middle frames help maintain smooth transitions and preserve video continuity.

7. Results

The video compression pipeline successfully reduces the file size by extracting keyframes and adding middle frames between them. The results can be summarized as follows:

- Keyframes Extraction:

Keyframes were successfully extracted using optical flow and histogram difference techniques.

- Middle Frames:

Middle frames were added between keyframes wherever the difference in frames was above the threshold.

- Compression Ratio:

The compression ratio was improved by keeping only the keyframes and middle frames, significantly reducing the video size without a noticeable loss in quality.

- Decompression:

The decompressed video was generated by interpolating the middle frames and reconstructing the video from the keyframes. The quality of the decompressed video was visually close to the original.

8. Time Complexity Analysis

- Keyframe Extraction:

Optical Flow: O(n), where n is the number of frames in the video.

Histogram Difference: O(n), where n is the number of frames.

Linear Interpolation: O(1) for each pair of consecutive keyframes, leading to O(k) where k is the number of keyframe pairs.

- Compression: O(n) for processing each frame.

- Decompression: O(n) for reconstructing each frame.

8. Space Complexity Analysis

- Storage for Keyframes: O(n), where n is the number of keyframes.

- Storage for Middle Frames: O(k), where k is the number of middle frames added.

- Total Space Complexity: O(n + k).

9. Conclusion

The video compression pipeline effectively reduces the file size by using keyframe extraction based on Optical Flow and Histogram Difference. Middle frames are added between keyframes to smooth transitions, and decompression is achieved by interpolating these middle frames.

10. Contribution

Ashish Mishra: EDA, Video Decompression, Report compilation

Shriyansh Bhardwaj: Video Compression Pipeline Creation

Arun Kumar: PPT compilation

Shraddha Chaudhary: Literature Survey