Image Processing - Exercise 1

Hadar Tal, hadar.tal, 207992728

Introduction

The primary objective of this exercise is to identify scene cuts in videos, with a focus on analyzing grayscale histograms. The goal is to discern distinct patterns in the distribution of grayscale values within video frames, particularly between different scenes. The main technique employed revolves around leveraging the insights provided by grayscale histograms, which serve as a visual representation of pixel intensity frequencies. By examining the nuances in these histograms, the exercise aims to develop algorithms capable of robustly detecting scene transitions based on the changing grayscale characteristics.

The key distinction between videos in category 1 and category 2 lies in the nature of their grayscale histograms. In category 1, the histograms for each frame within a scene exhibit a relatively similar pattern. In contrast, videos in category 2 present a notable shift in the distribution of grayscale values within the same scene. This dissimilarity in histogram characteristics necessitates different approaches for effective scene cut detection in each category. Notably, in category 2, a direct calculation of the difference in histograms may not adequately capture the nuanced changes in grayscale distribution, requiring a more intricate analysis.

Algorithm

Category 1:

1. Frame Preprocessing: Convert each video frame to grayscale.
2. Histogram Calculation: For each frame, compute a grayscale histogram.
3. Compute the distance between consecutive frames based on the histograms.
4. Scene Cut Identification: Identify the frame with the maximum histogram distance as the scene cut point.

Category 2:

1. Frame Preprocessing.
2. Quantization: Optimal quantization of video frames is performed to reduce the number of intensity levels. This step involves clustering pixel intensity values using K-means clustering.
3. Cumulative Histogram Calculation.
4. Compute the distance between consecutive frames based on the histograms.
5. Scene Cut Identification.

Implementation Details

In the first stage, Frame Preprocessing involves converting each video frame to grayscale using the *mediapy* and *PIL* libraries. Subsequently, for each frame, a grayscale histogram is computed. In this context, a histogram is a distribution of pixel intensity values, where each bin represents a specific intensity level, and the bin value corresponds to the frequency of pixels with that intensity. This computation is based on *numpy*'s histogram method, chosen for its ability to accelerate the speed of mathematical calculations.

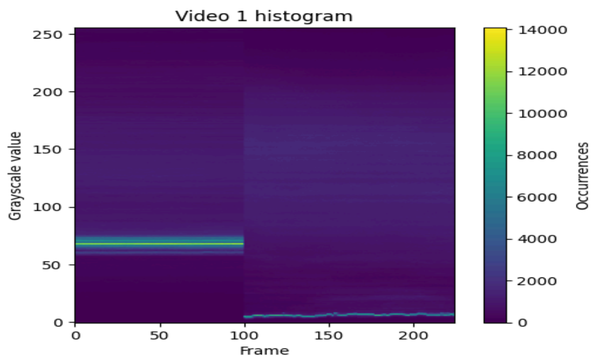
Following that, the distance between consecutive frames is computed based on their histograms. The distance is calculated by summing the absolute differences between corresponding bin values in two consecutive frames, with *numpy* being utilized for this purpose. The frame with the maximum histogram distance signifies the transition between scenes, facilitating Scene Cut Identification.

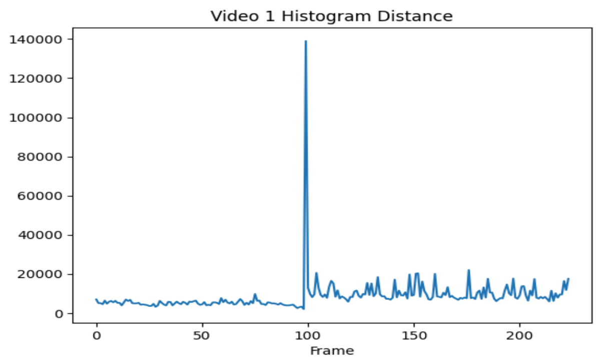
Moving on to Category 2, Optimal Quantization of video frames is performed to reduce the number of intensity levels. This step involves clustering pixel intensity values using K-means clustering. To expedite processing time, a random but uniform subset of frames is chosen for clustering, constituting 5% of the frames. This subset ensures representative quantization levels while maintaining computational efficiency. During this process, there was a need to determine the necessity and quantity of intensity levels. Tests were conducted to find the number of levels that achieve maximal difference from the distance between scene changes to the distance within the same scene. Despite variations in the number of levels between videos, applying their mean (10) resulted in a greater difference compared to scenarios without quantization. If I had access to a larger dataset of videos, it would allow for a more comprehensive analysis, enabling the selection of a more accurate number of intensity levels. The choice of using the *sklearn.cluster* library was made due to its well-known reputation for calculating these values, and the decision not to implement a custom solution was driven by the course's focus and the algorithm's appropriateness even without this step. For Cumulative Histogram Calculation, *numpy*'s cumsum method was employed.

The main challenge encountered was determining if additional methods should be applied to yield better results. While acknowledging that many methods, such as object recognition or image tagging, are beyond the scope of this assignment, an exploration was conducted to ascertain their relevance. Attempts to calculate the histogram's distance after applying Gaussian filters or reconstructing the image by FFT were made, but these efforts did not lead to improved results.

Category 1 Results

The scene cut in video1 is identified within frames (99,100) and in video2 within (149,150). Visualizations elucidating the scene transition for Video 1 are provided below. The histogram distribution across frames is depicted in the left figure, while the right figure illustrates the calculated distances between consecutive frames.

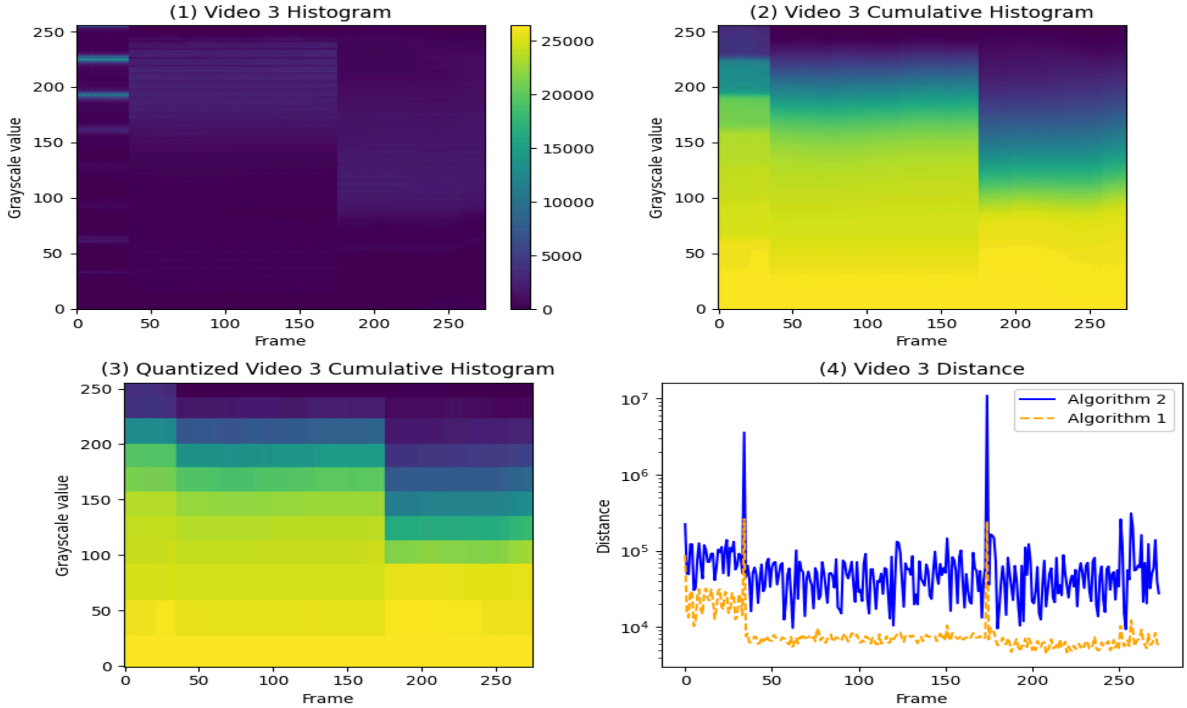




Category 2 Results

The scene cut in video3 is identified within frames (174, 175) and in video4 within (74, 75). As mentioned in the introduction, the key distinction between videos in category 1 and category 2 lies in the nature of their grayscale histograms. In category 1, the histograms for each frame within a scene exhibit a relatively similar pattern. In contrast, videos in category 2 present a notable shift in the distribution of grayscale values within the same scene. As we can see in Figure (1), even when there is no scene change, in the 35th frame, there is a major change in the histogram of the frames.

In Figure (2), it is evident that the cumulative sum amplifies the contrast between different scenes while maintaining a relatively subtle difference in regions where there is a shift in the distribution of grayscale values within the same scene. Moving on to Figure (3), the cumulative histogram of the quantized video is presented, offering insights into the altered representation. Finally, in Figure (4), the modified algorithm's distance between consecutive frames is illustrated in blue, providing a clear comparison to the category 1 algorithm depicted in orange.



Conclusion

In conclusion, this exercise focused on detecting scene cuts in videos through grayscale histogram analysis, revealing distinct characteristics in category 1 and 2 videos. Successful scene cut identifications were achieved with tailored algorithms for each category. Visualizations showcased histogram distributions and frame distances, emphasizing differences in grayscale patterns. Challenges in exploring additional methods were encountered, but the exercise provided valuable insights. Overall, this foundational exploration underscores the importance of tailored approaches and sets the stage for further advancements in video analysis.