Scene Change Detection Using TAD and Custom SVD in Video Analysis

Regaip Cuma Erdihan
Istanbul Technical University
Faculty of Computer and Informatics
Student ID: 150230022
Email: erdihan23@itu.edu.tr

I. Introduction

Scene change detection is a important step in video analysis, enabling efficient video indexing, browse etc.. This project implements a scene change detection based on the provided article, ”Fast Video Shot Boundary Detection Based on SVD and Pattern Matching” by Zhe-Ming Lu and Yong Shi. The detection of scene change includes a two-stage detection process: firstly detecting candidate scene changes using the Total Absolute Difference (TAD) method, and then testing these canditates are whether real changes or not by using Singular Value Decomposition (SVD).

II. Methodology

The scene change detection process follows this implementation:

A. Extracting frames

The extract_frames function in both scene_detection.py and calc_all.py is reading the video files frame by frame and then each frame is converted to grayscale to reduce complexity.After that stored as a float32 NumPy array for further implementation.

B. Total Absolute Difference (TAD) for Candidate Scene Changes

The TAD is used as an first filter to detect potential scene changes. For consecutive frames F_i and F_{i+1}, the TAD is calculated as:

$$TAD(F_i, F_{i+1}) = \sum |F_i - F_{i+1}| \tag{1}$$

A frame is noted as a candidate scene change if its TAD value with the subsequent frame exceeds an adaptively determined threshold. This threshold is calculated as a multiplier of the standard deviation of all TAD values across the video, allowing for effective implementation.

C. Singular Value Decomposition (SVD) for Detecting Final Changes

The frames firstly detected as TAD candidates are then move to SVD analysis. The SVD is performed on the grayscale image data, which is treated as a matrix. The dominant singular values are extracted, and the Euclidean distance between the singular value vectors of consecutive candidate frames is computed. If this distance bigger than predefined threshold, the scene change is confirmed.

The SVD algorithm is implemented from scratch as specified in the project requirements.

The custom SVD implementation, svd_custom(A), is a critical component of this project. It is based on the relationship between SVD and Eigenvalue Decomposition.

A separate gram_schmidt function is also provided, which implements the Gram-Schmidt process for QR decomposition, and qr_algorithm_symmetric which implements the QR algorithm for finding eigenvalues and eigenvectors of a symmetric matrix. While numpy.linalg.eig is used in svd_custom for practical reasons (as implementing a robust eigenvalue solver from scratch is a project in itself), these functions demonstrate an understanding of the underlying numerical methods.

III. Code Structure and Usage

The project consists of two primary Python scripts:

scene_detection.py: This is the main script for performing scene change detection. It contains:

The gram_schmidt, qr_algorithm_symmetric, and svd (renamed to svd_custom in calc_all.py for clarity) functions.

extract_frames: Extracts grayscale frames from a video.

calculate_tad: Computes the Total Absolute Difference between two frames.

calculate_svd_distance: Computes the Euclidean distance between the dominant singular values of two frames.

detect_shot_boundaries: Orchestrates the entire detection process, applying TAD and then SVD to confirm boundaries. It saves detected transition frames as JPG images and lists them in a transitions.txt file within a specified output directory.

The script is configured to run detect_shot_boundaries for video1.mov and video2.mov with adjustable tad_std_multiplier and svd_threshold parameters.

calc_all.py: This utility script is designed to help in determining appropriate tad_std_multiplier and svd_threshold values by generating analysis data. It calculates and saves all TAD values and SVD distances between consecutive frames to text files. This allows for post-analysis to identify typical distributions and effective thresholds for different video types.

IV. Experimental Results and Analysis

To determine optimal tad_std_multiplier and svd_threshold values, calc_all.py was used to analyze the video1.mov and video2.mov files.

Analysis with calc_all.py: The script generates videox_diff.txt that includes TAD and SVD values for each video.. For instance, a tad_std_multiplier of 1.0 was chosen for video1.mov and video2.mp4 after observing the spread of TAD values and identifying a reasonable cutoff point for candidate selection. Similarly, svd_std_multiplier values of 0.3 for video1.mov and 0.5 for video2.mp4 were selected based on the SVD distance distributions, aiming to confirm significant changes.

Scene Detection Performance (scene_detection.py): The detect_shot_boundaries function uses these empirically determined thresholds. The output includes console messages indicating candidate transitions based on TAD and then confirmed transitions based on SVD. The detected transition frames are saved as transition_X.jpg in the respective output folders (results_video1, results_video2), and a transitions.txt file lists the frame numbers.

The "TAD THEN SVD" logic implemented ensures that a frame is only considered a confirmed shot boundary if it was a TAD candidate and its SVD distance is above the SVD threshold. This two-stage filtering helps in reducing false positives.

Example output from scene_detection.py for video1.mov:

```
Adaptive TAD threshold: 7766429.00
Adaptive SVD threshold: 1094.01

-Scene Change detected: Frame 149
TAD: 61292352.00, SVD Distances: 15530.99
-Scene Change detected: Frame 299
TAD: 42249904.00, SVD Distances: 6914.53
-Scene Change detected: Frame 449
TAD: 40120676.00, SVD Distances: 3040.19
-Scene Change detected: Frame 621
TAD: 45354440.00, SVD Distances: 5664.81
-Scene Change detected: Frame 747
TAD: 44696280.00, SVD Distances: 1297.95
```

```
And transitions.txt:

Detected Shot Transitions:
Frame 149
Frame 299
Frame 449
Frame 621
Frame 747
```

The generated .jpg files included in scene_detection.zip.

V. Conclusion

This project successfully implements a video shot boundary detection system utilizing Total Absolute Difference (TAD) for candidate selection and Singular Value Decomposition (SVD) for confirmation. The SVD function was implemented from scratch, adhering to the project requirements. The use of a two-stage approach enhances the accuracy of detection by filtering out noise and confirming only significant scene changes.

VI. Example Image Outputs

Fig. 1. Detected scene change at Frame 149

Fig. 2. Detected scene change at Frame 449

References

[1] Z. M. Lu and Y. Shi, "Fast video shot boundary detection based on SVD and pattern matching," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2006.