

**Spatio- Temporal segmentation**

COURSE - IMAGE AND VIDEO ANALYTICS DATE - 11/10/2024

Submitted By: Submitted To:

Mohamed Riyaz S Dr. Saranyaraj D 21MIA1138

**Objective:**

* The objective of this assignment is to perform spatio-temporal segmentation on a provided video by analyzing its frames over time.
* The segmentation aims to distinguish moving objects from the background and detect scene transitions (hard and soft cuts).
* The results will be visualized to showcase the segmented frames and scene cut boundaries.

**Problem Statement:**

In this task, a video is provided, and the goal is to extract meaningful information by processing individual frames. The problem can be divided into three primary areas:

1. Frame Extraction: Extract individual frames from the video to prepare them for further analysis.
2. Spatio-Temporal Segmentation: Perform segmentation of each frame to track and analyze objects in motion while distinguishing the foreground from the background.
3. Scene Cut Detection: Detect scene changes, including abrupt (hard) and gradual (soft) cuts, by analyzing pixel differences and intensity changes between consecutive frames.

**Expected Output:**

1. **Frame Extraction**: A series of images representing individual frames from the video.
2. **Spatio-Temporal Segmentation**: Segmented frames where moving objects (foreground) are separated from the background.
3. **Scene Cut Detection**: A list of frames where scene transitions are detected, along with a summary showing these boundaries.
4. **Result Visualization**: Display key frames with segmentation results and highlight scene cuts, visualizing both hard and soft transitions.

**Algorithm for Spatio-Temporal Segmentation:**

1. **Load the Video**:
   * Open the video file and load it frame by frame for analysis.
2. **Frame Extraction**:
   * Loop through the video and extract each frame as an image.
   * Store the frames for further processing.
3. **Spatio-Temporal Segmentation**:
   * Convert each frame to grayscale.
   * Apply a segmentation technique, such as edge detection or color thresholding, to identify the regions of interest.
   * Track objects in motion across consecutive frames to analyze changes in motion and shape.
4. **Foreground-Background Separation**:
   * Use pixel intensity differences between consecutive frames to identify areas that remain constant (background) and areas with movement (foreground).
5. **Scene Cut Detection**:
   * Compute pixel-based or histogram differences between consecutive frames.
   * Detect hard scene cuts by identifying abrupt pixel changes.
   * Identify soft scene transitions by analyzing gradual intensity changes over multiple frames.
6. **Result Visualization**:
   * Display frames with detected scene cuts.
   * Visualize segmentation results, highlighting foreground objects and scene cuts.

**Pseudo Code:**

**Step 1: Load Video**

Start

Open video file

If video cannot be loaded, return error

**Step 2: Frame Extraction**

For each frame in the video:

Read frame

Save frame as an image

End loop

**Step 3: Spatio-Temporal Segmentation**

For each extracted frame:

Convert frame to grayscale

Apply edge detection or color thresholding for segmentation

Track segmented regions across frames to detect motion

End loop

**Step 4: Foreground-Background Separation**

For each consecutive frame pair:

Compute pixel intensity differences

Separate moving objects (foreground) from static areas (background)

End loop

**Step 5: Scene Cut Detection**

For each consecutive frame pair:

Compute pixel or histogram differences

If pixel difference exceeds a threshold:

Mark as hard scene cut

Else if gradual intensity change occurs:

Mark as soft scene transition

End loop

**Step 6: Result Visualization**

For frames with scene cuts:

Display frames

Highlight scene boundaries and segmented objects

End loop

End

**Python Implementation:**

**Task 1:**

**Load Video:**

**Load the provided video file.**

****

**Task 2:**

**Frame Extraction:**

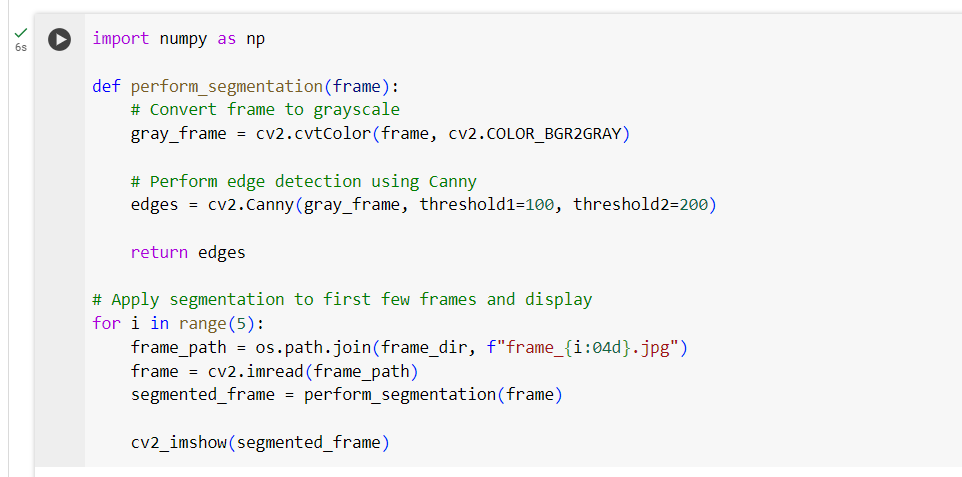
**Extract individual frames from the video.**

****

**Task 3:**

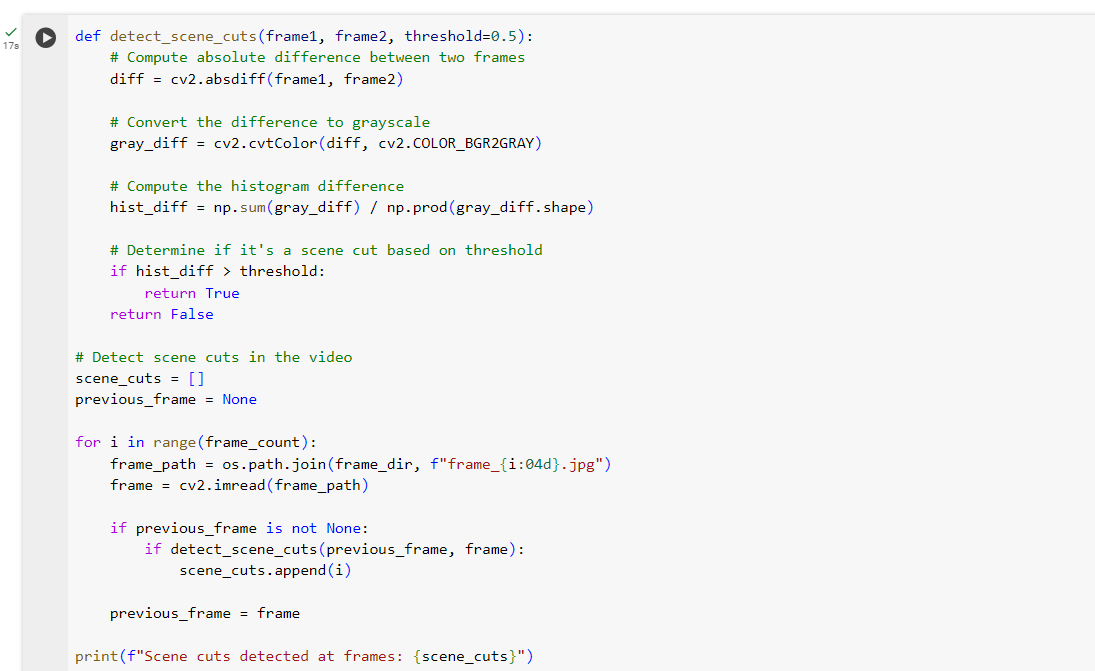
**Spatio-Temporal Segmentation:**

**Perform segmentation on each frame using a technique like color thresholding or edge detection. Track the segmented objects across frames to observe changes in motion and shape. Identify the regions that remain consistent over time (foreground vs. background segmentation).**

****

**Scene Cut Detection:**

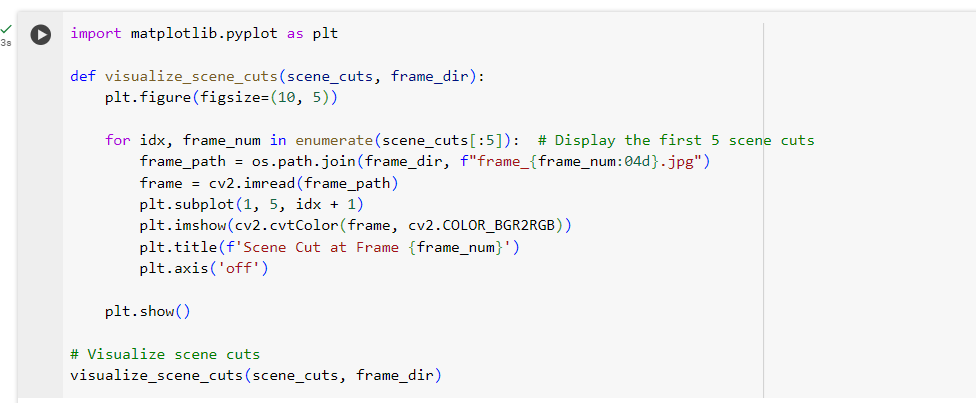
**Use pixel-based comparison or histogram differences between consecutive frames to detect abrupt changes (hard cuts). Detect gradual scene transitions (Soft cuts) by analyzing frame-to-frame intensity changes over time.**

****

**Task 5**

**Mark Scene Cuts:**

**Highlight the frames where scene cuts are detected. Create a summary displaying the detected scene boundaries.**

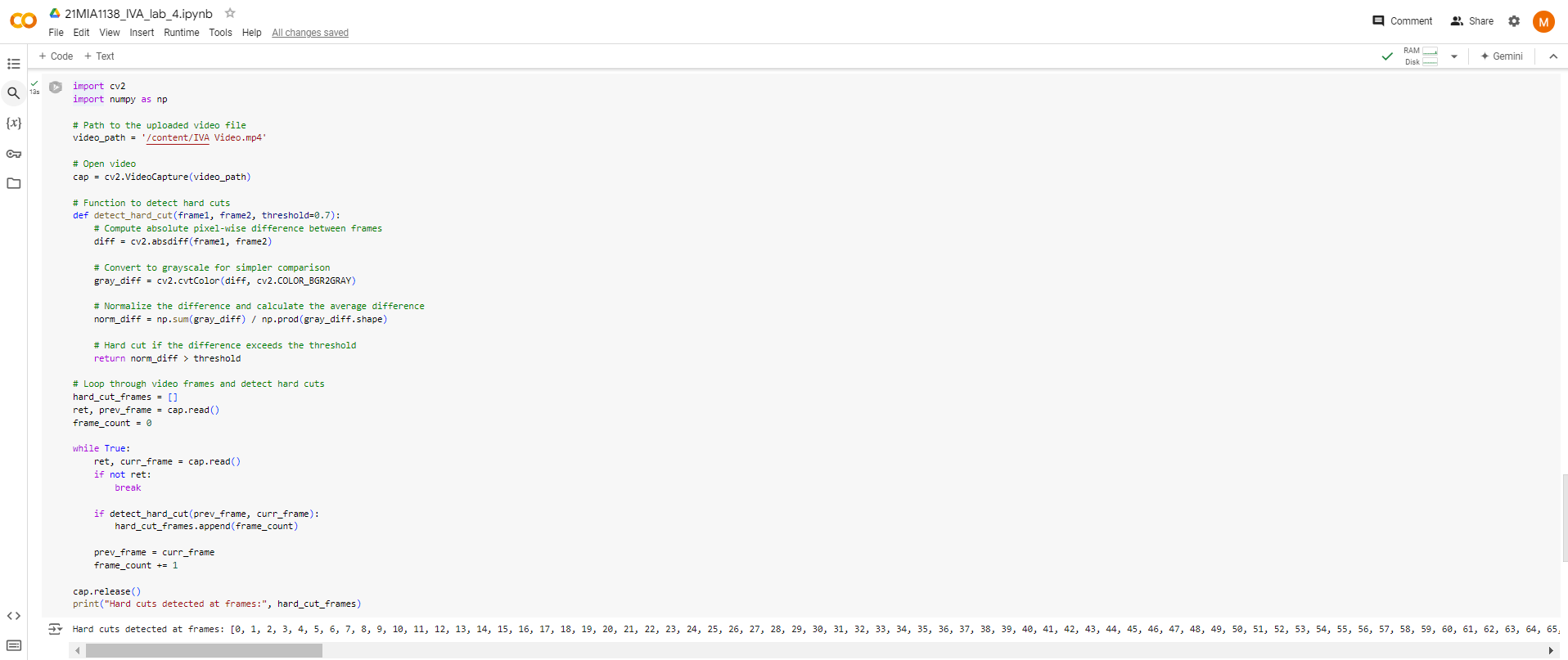
****

**Result Visualization:**

****

**Hard Cut Detection**

Hard Cut Detection involves identifying abrupt transitions between consecutive frames. We'll calculate pixel-wise differences between consecutive frames and check if the difference exceeds a threshold, indicating a hard cut.



### Hard Cut Detection:

### Using histogram correlation, the following frames were identified with significant changes:

Hard cuts detected at frames: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365]

**Soft Cut Detection**

Soft Cut Detection identifies gradual transitions (like fades or dissolves) by analyzing intensity changes over a sequence of frames. We will use a sliding window to compute these changes over multiple frames.

Soft cuts detected at frames: [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363]



**GitHub Link:**

<https://github.com/Siuuu07/21MIA1138_IVA_lab_4.ipynb/blob/main/21MIA1138_IVA_lab_4.ipynb>