Part1: Video Smmarisation

1. The model used for the purpose

The approach used in the code is **Principal Component Analysis (PCA)** to identify key frames in a video. The key steps include:

- Extracting frames from the video.
- Converting frames to grayscale and resizing them for uniformity.
- Computing motion between consecutive frames using frame difference.
- Applying **PCA** on the motion frames to find dominant variations in the video.
- Selecting key frames by analyzing the most significant principal components.

PCA helps to **reduce dimensionality** and capture frames that have significant changes, making it effective in summarizing video content without deep learning.

2. The summarized final video

The summarized final video is created by:

- Segmenting the video into **10 parts**.
- Extracting **2 key frames** per segment using PCA-based motion analysis. On a need basis, we can increase it beyond this, but in my case, it didn't help as the optimal was 2
- Stitching these frames together into a new video.
- The output is saved as `"summarized_video.mp4"`.

3. Experimentation with different numbers of frames

To experiment with different numbers of keyframes I did the following stuff:

- Varied the **n_segments** and **n_key_frames_per_segment** values in `segment_and_summarize()`.
- Observed how the summarization quality changes with different values.
- I found 10 segments and 2 frames per segment worked in our case

4. How do you select the optimal number of frames?

The optimal number of frames can be selected based on:

- **Visual analysis**: We need to check if the keyframes are able to capture the significant events. In our case, 2 frames per segment were optimal and the output was fast, increasing more was not giving better results.
- **Motion analysis**: By doing the motion analysis, If the motion between selected keyframes is too low, increasing frames helped us as we started initially with just 1 frame.

- **Variance explained by PCA**: A higher explained variance means more information is retained.

5. Evaluation measure for summarization goodness

Possible evaluation metrics that we can use:

- **Reconstruction Error**: We can measure how well key frames reconstruct the original video using PCA-inverse transformation to evaluate the model's summarization abilities
- **User Feedback**: We can ask users to rate the summary's effectiveness.

Part3: BG Subtraction Video Summarisation

1. **Impact of Tuning Parameters**

   GMM Parameters:

   - `n_components`: Increased from 3 to 5 for summarized video to handle more background variations
   - `learning_rate`: Increased from 0.01 to 0.05 for faster adaptation between keyframes
   - `threshold`: Increased from 0.7 to 0.8 for more aggressive background modeling

   Detection Sensitivity:

   - Changed `background_prob < 0.1` thresholds to adjust foreground detection sensitivity
   - Modified the matching threshold `2.5 * np.sqrt(self.covars[:,:,i])` to control component matching

2. **Comparison**

   i.) Without Summarization (Original Video):

   - Advantages:
     - Clearer separation between foreground and background objects
     - Less flickering in the output video
     - More stable background model due to continuous frame processing
     - Better temporal consistency since all frames are processed sequentially
     - Higher quality foreground masks due to more information available for GMM
     - More accurate object boundaries in the foreground detection

- Technical Analysis:
  - The GMM model has more frames to learn the background distribution
  - Gradual changes in lighting and scene are better captured
  - The K-means refinement step works better with higher-quality input
  - Background updates are smoother due to frame-by-frame processing

ii.) With Summarization:

- Limitations:
  - Less clear separation between foreground and background
  - Increased flickering in the output
  - Reduced quality of foreground/background separation
  - Potential loss of temporal consistency between keyframes
  - More challenging for the GMM to establish a stable background model
- Technical Analysis:
  - Fewer frames available for background modeling
  - Large temporal gaps between frames can cause abrupt changes
  - The GMM may struggle to adapt to sudden changes between keyframes
  - K-means refinement may be less effective due to temporal discontinuity