Interactive Video Summarization:

Objectives:

Input : video

Output:

→summerize transcript

→give time stamps for that key point where the keypoint discuss

Models used:

whisper→ audio to text

### **Steps for Video Summarization**

1. **Video Input**
   * **Load the Video**: Start by loading the video file into your program using libraries like OpenCV or MoviePy.
2. **Audio Extraction**
   * **Extract Audio**: Extract the audio track from the video. This step is crucial for converting spoken content into text later on.
   * **Example**: Use FFmpeg or MoviePy to extract audio from the video.
3. **Speech Recognition**
   * **Convert Audio to Text**: Use a speech recognition library (like SpeechRecognition) to transcribe the extracted audio into text. This text will serve as the basis for summarization.
   * **Clean the Transcribed Text**: Remove filler words, stutters, and unnecessary noise from the transcription.
4. **Text Summarization**
   * **Summarize the Transcribed Text**: Apply Natural Language Processing (NLP) techniques to generate a concise summary of the transcribed text. This can be achieved using pre-trained models from libraries like Hugging Face’s Transformers (e.g., BART, T5).
   * **Example**: Use a summarization pipeline to condense the text into key points.
5. **Scene Extraction (Optional)**
   * **Keyframe Extraction**: Analyze the video frames to identify key scenes or moments that are visually significant. This can involve extracting frames at regular intervals or detecting changes in scenes.
   * **Visual Features Analysis**: Use methods like color histograms, motion detection, or shot boundary detection to identify important segments in the video.
6. **Summarization Output**
   * **Combine Text and Visuals**: Present the summarization results by combining the textual summary with the identified keyframes or segments from the video.
   * **Format the Output**: Structure the summary output to make it user-friendly, possibly including timestamps for the highlighted scenes.
7. **User Interface (Optional)**
   * **Create a User Interface**: If desired, develop a web or desktop application that allows users to upload videos and view summaries.
   * **Display Summary and Keyframes**: Provide a clear view of the summary and the corresponding key scenes or segments.
8. **Testing and Evaluation**
   * **Test the Summarization**: Evaluate the effectiveness of the summarization on various types of videos (e.g., lectures, tutorials).
   * **Collect Feedback**: Gather user feedback on the quality and relevance of the summaries to make improvements.
9. **Iteration and Improvement**
   * **Refine the Process**: Use feedback and evaluation results to enhance the summarization algorithms and user interface.
   * **Explore Advanced Techniques**: Consider incorporating more advanced methods such as reinforcement learning or deep learning techniques for better summarization quality.