Lecture video key frame identification via local open-source AI models

Sunwoo Baek, Supia Park, Ashley Li, Charitha Nannapaneni; University of Illinois, Urbana-Champaign

Motivation

Creation of alternative, equivalent learning resources to aid learning.

Problem Statement and Context

- How can we use local open-source AI models to accurately identify the key frames in instructional media?
- Videos are a prevalent form of instructional media, but those with visual/auditory impairments and/or those who prefer text-based media face barriers to accessibility. Converting lectures into structured digital books, with organized chapters, images, and transcripts, aligns with UDL principles.
- Such a digital book would feature segments of lecture content organized into chapters, each accompanied by representative images and transcriptions
- Beyond traditional books, the scene detection process enables novel content formats, such as short, engaging video summaries or entertainment visuals
- Overly sensitive detection captures minor changes like scrolling text or facial movements, cluttering the output (false positives) while a stricter approach can miss subtle but important updates, resulting in lost content (false negatives).

Existing Methods

- Structural Similarity Index Measure (SSIM) compares frames based on visual similarity
- An enhanced metric (SSIM No Face) masks out faces and bodies
- Optical Character Recognition (OCR) is also applied to detect changes in visible text (helps with false positives)
- Support Vector Machine (SVM) model uses the above metrics to classify and distinguish frame transitions; trained on labeled lecture videos. Utilizes an early-dropping technique to skips calculations when frames have high structural similarity, reducing computation time without significantly impacting accuracy.

Such methods yield high accuracy, with the SVM model achieving over 99% accuracy in test cases. However, limitations remain. Scene detection struggles with incremental changes, such as when instructors write on-screen, leading to unnecessary scene divisions. Runtime optimizations make this approach feasible on CPU-only setups, yet improvements in handling complex annotations and fine-grained changes are still needed for robust scene detection across diverse lecture styles.

What We've Tried So Far

Large Multimodal Models

LLaVA 7B

- Basic prompts to describe images managed to describe general concepts, but struggled with details (e.g., hallucinated Python code from a CS 341 lesson screenshot)
- Prompting for differences between two images was unsuccessful

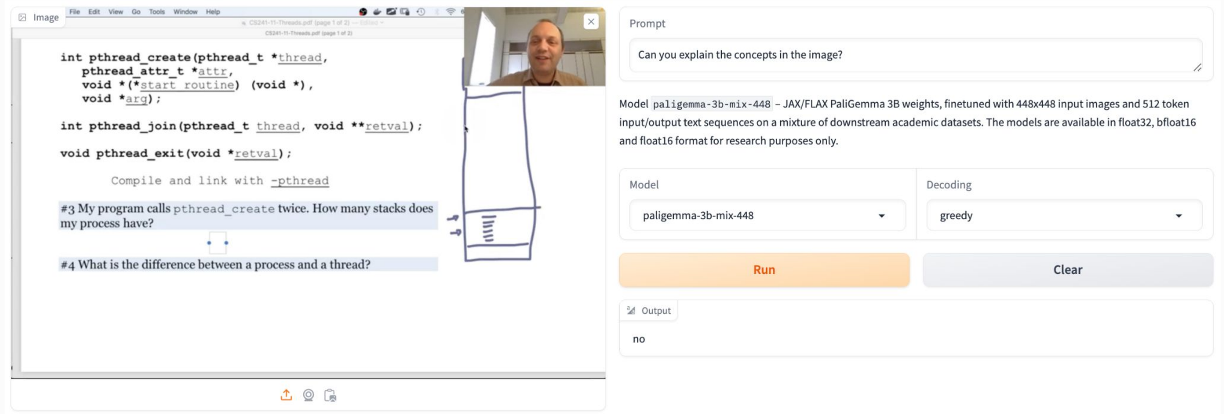
```
>>> ollama run llava "please transcribe the text following images: ./image2.jpg, ./image3.jpg"
Added image './image2.jpg'
Added image './image3.jpg'
 Na imagen, vemos um computador com uma janela aberta, mostrando código de programação. O código parece ser
de uma interface de usuário (UI) que está sendo desenvolvida em PHP usando o framework Laravel. A UI tem
elementos de login e registro de usuários, bem como um carrossel com várias imagens, incluindo algumas
fotografias. O código também parece ser parte de uma tarefa de programação.

A pessoa em foco parece estar analisando o código e discutindo-lo, possivelmente explicando o funcionamento
ou como foi desenvolvido. Ela usa o Zoom para acomunicação, que é um software popular para reuniões remotas
de trabalho.

Se você tiver perguntas específicas sobre este código ou precisar de ajuda com a programação em PHP,
ficarei feliz em ajudar!
```

PaliGemma

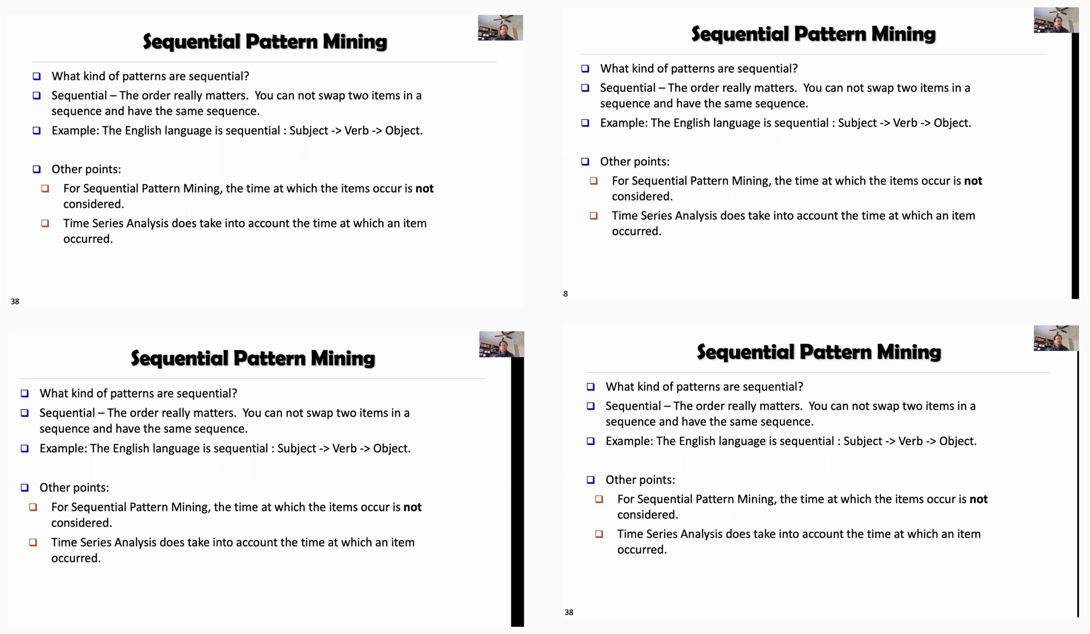
- Did not hallucinate as badly as LLaVA 7B, but did so by refusing to perform some tasks



VGG-16

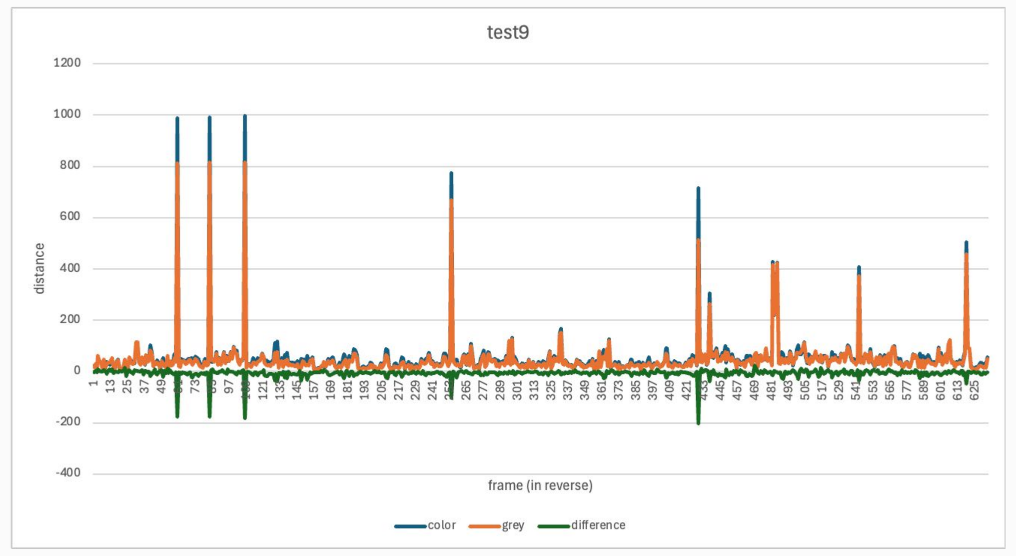
In color

- Calculated euclidean distance between consecutive pairs of frames in a video as a measure of difference
- Was able to identify scene changes, but still had many false positives



In grayscale

- Emulating [1], we tried converting images to grayscale to improve scene detection and reduce processing time
- Had the opposite effect



Potential Future Approaches

- Investigate new models:
  - Llama vision 3.2, Nvidia NVLM 1.0, InternVL2, and other local open-source AI models
- We would like to focus on newer models, in the assumption that they are better trained with larger data seat
- In the future, we would like to consider combining multiple approaches, since it is unlikely that one algorithm will do everything we require

Up Next

- We're learning more about our current CNN approach to understand where our false positives are coming from and how we can improve
- We're also working on labeling more videos so we have more information to test on

References

[1] Angrave, L., & Li, J., & Zhong, N. (2022, August), *Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem* Paper presented at 2022 ASEE Annual Conference & Exposition, Minneapolis, MN. 10.18260/1-2--41185

CS STARS
STUDENT AMBASSADORS/RESEARCH SCHOLARS