# Enhancing Keyframe Extraction in Lecture Videos Using Optical Flow

**Sunwoo Baek, Supia Park**
**Siebel School of Computing and Data science, University of Illinois Urbana Champaign**

## INTRODUCTION

Lecture videos have a variety of forms, including a mix of static slides and dynamic annotations, which makes it challenging to extract keyframes that best represent the content. In this research we are addressing the specific problem: given an hour-long lecture video, how do we extract the key frames to create a concise PDF without redundant images from the lecture? (Probably add something about improving accessibility in education here).

Traditional methods like pixel similarity and OCR-based text extraction are not the most effective to capture gradual, non-discrete changes such as handwritten notes, highlights, and incremental annotations.

To address this, we explore a series of methods to complete the stated task. We utilize methods such as optical flow, masking and subtraction in order to efficiently and accurately select representative frames from a lecture video.

## AIM

**How to use this template**
Highlight this text and replace it with new text from a Microsoft Word document or other text-editing program. The text size for body copy and headings and the typeface has been set for you. If you choose to change typefaces, use common ones such as Times, Arial, or Helvetica and keep the body text between 26 and 32 points.

The text boxes and photo boxes may be resized, eliminated, or added as necessary. The references to the department, college and university, including the logo, should remain.

Refer to the next page for logos commonly used on campus posters. You can drag and drop them to your personal PowerPoint scrapbook for use in subsequent posters; refer to PowerPoint help documents for more specific information regarding how to use the scrapbook.

You can replace the Block I Wordmark in the lower right with your unit lockup.

## METHOD

**Exploring Optical Flow**

By analyzing motion vectors between consecutive frames, optical flow allows us to track scrolling movements rather than relying on absolute pixel differences. Our experiments with Lucas-Kanade and Dense Optical Flow show that this method effectively detects the gradual evolution of content, and we are working on refining this approach to better differentiate meaningful updates (e.g., new annotations) from insignificant transitions. However, challenges remain in filtering out background movement artifacts and distinguishing intentional content changes from noise. This approach provides a complementary perspective to static frame comparisons, offering a motion-based strategy for scene detection.

*Captions set in a serif style font such as Times, 18 to 24 size, italic style.*

*Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.*

**Selecting the proper frames (can prob be a better title)**

To determine the importance of each frame, we apply masking and subtraction methods to quantify the differences between selected frames. By analyzing implementations using K-Means Clustering, Spectral Masking, and Thresholding algorithms, we have found that thresholding is significantly more accurate and efficient, operating in a fraction of the time relative to the input size. After detecting the differences between frames, we are able to accurately retrieve the key frames--successfully identifying scene changes in annotated videos.

*Captions set in a serif style font such as Times, 18 to 24 size, italic style.*

*Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.*

## RESULTS

**Images**
TIFFs are the preferred file format for images appearing in printed posters. Avoid the use of low-resolution jpgs, especially those downloaded from the Internet, as they will reproduce poorly.

In order to insert an image, use the menu toolbar at the top of your screen.

Select:
1 Insert
2 Picture
3 From file
4 Find and select the correct file on your computer
5 Press OK

Be aware of the image size you are importing.

*Captions set in a serif style font such as Times, 18 to 24 size, italic style.*

*Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.*

*Captions set in a serif style font such as Times, 18 to 24 size, italic style.*

*Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.*

*Captions set in a serif style font such as Times, 18 to 24 size, italic style.*

*Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.*

## PRINTING

We initially applied the Lucas-Kanade Optical Flow method due to its efficiency and suitability for tracking small, gradual movements. However, we observed that it performed poorly in cases involving abrupt screen transitions or rapid scene changes—common in lecture videos when slides change or the camera view shifts. This is due to its underlying assumption of small motion and brightness constancy, which is frequently violated in educational video contexts.
To address this, we explored Dense Optical Flow methods, including Farneback, which provide a more detailed, pixel-wise estimation of motion across frames. These approaches were more robust in detecting the onset of new annotations or scrolling movements, and they helped better highlight areas of meaningful content change.
Farneback Optical Flow offered a good balance between computational efficiency and motion sensitivity. While it still picked up some noise—particularly during transitions with background movement—it allowed us to define motion magnitude thresholds that filter out less informative regions.
Our results suggest that dense optical flow provides more accurate detection of educational content changes, especially when combined with post-processing techniques like thresholding and noise reduction. These methods enable a more reliable identification of

## References

[1] Angrave, L., Li, J., and Zhong, N. 2022. Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem. In Proceedings of the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN.

[2] OpenCV. Optical Flow. OpenCV Documentation. https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html

ILLINOIS

# Enhancing Keyframe Extraction in Lecture Videos Using Optical Flow

**Sunwoo Baek, Supia Park**
**Siebel School of Computing and Data Science, University of Illinois Urbana Champaign**

## INTRODUCTION

Lecture videos have a variety of forms, including a mix of static slides and dynamic annotations, which makes it challenging to extract keyframes that best represent the content. In this research we are addressing the specific problem: given an hour-long lecture video, how do we extract the key frames to create a concise PDF without redundant images from the lecture? Traditional methods like pixel similarity and OCR-based text extraction are not the most effective to capture gradual, non-discrete changes such as handwritten notes, highlights, and incremental annotations.

To address this, we explore a series of methods to complete the stated task. We utilize methods such as optical flow, masking, and subtraction in order to efficiently and accurately select representative frames from a lecture video.

## APPROACH

In our approach, a lecture video will go through a series of methods, refining the work from Angrave, Li, and Zhong, 2022, that is currently implemented in ClassTranscribe.

First, we use Farneback dense optical flow to track how pixels move from one frame to the next. This helps us detect scrolling by measuring how much of the screen moves vertically between frames.

Next, we calculate how much of the screen is involved in this movement and how consistent the motion is across different parts of the screen. This allows us to detect whether the video is truly scrolling or just experiencing noise, such as shaky camera footage.

By repeating this process for every pair of frames, we build a set of motion statistics that guide how we group scenes. These motion cues form the foundation for identifying clean content boundaries, improving the accuracy and reliability of our video segmentation pipeline.

## DETECTING SCROLLING

### Exploring Optical Flow

We analyze motion vectors between consecutive frames; optical flow allows us to track vertical movements. Applying Dense Optical Flow by Gunnar-Farneback effectively detects the gradual evolution of content, and we are working on refining this approach to better differentiate meaningful updates from insignificant motion.

### Applying Semantic Scene Segmentation with CNN

Pixel-wise differences for scene changes gave false positives from trivial visual variations (e.g., small text updates) that are not meaningful. To address this, we use CNN segmentation *(Figure 4)*, which captures high-level content rather than low-level noise. This enables us to group semantically similar frames, even when visual differences are subtle.

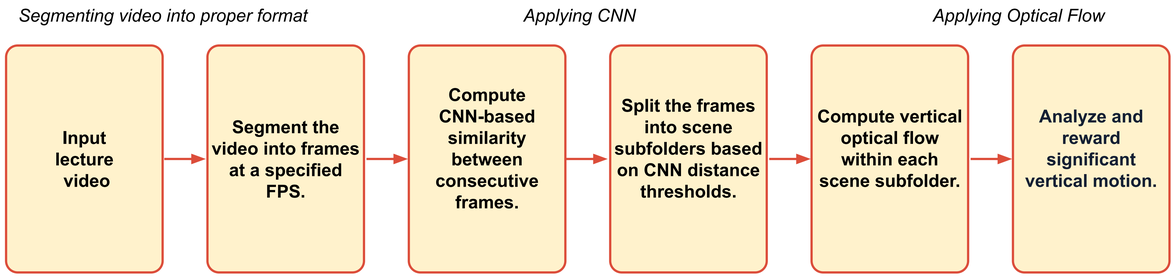
### Vertical Motion Estimation via Optical Flow

To accurately measure vertical motion such as scrolling, we first divide the video into subfolders of semantically similar frames using CNN-based scene segmentation *(Figure 5)*. This ensures that motion analysis is performed only within visually coherent segments, avoiding misleading spikes caused by scene changes. Within each subfolder, we apply Farneback optical flow to consecutive frame pairs to quantify vertical motion while excluding irrelevant transitions.

### Rewarded Vertical Motion (RVM)

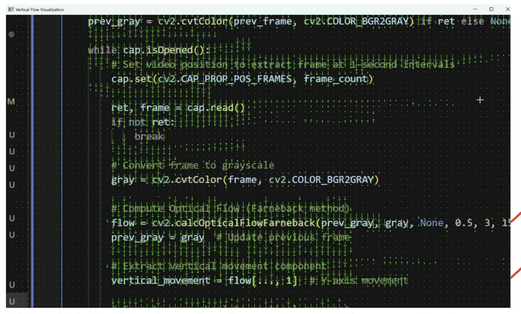
To prioritize widespread vertical movement over localized spikes, we define **Rewarded Vertical Motion (RVM)**, which amplifies motion scores when many pixels exhibit small movements *(Figure 6)*. This is crucial in lecture videos, where subtle scrolling affects a large area, while brief transitions or jitters may involve only a few regions *(Figure 7)*. RVM captures both the magnitude and the coverage of motion, rewarding consistency across the screen.

$$RVM = \left(\frac{Sum\ of\ Significant\ Motion}{Total\ Pixels}\right) \times \left(\frac{Moving\ Pixels}{Total\ Pixels}\right)^{\alpha}$$

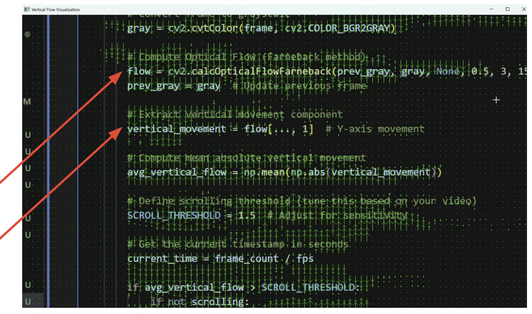



*(Figure 1) Pipeline illustrating how our method detects scrolling behavior*

## VISUALIZATION OF OPTICAL FLOW

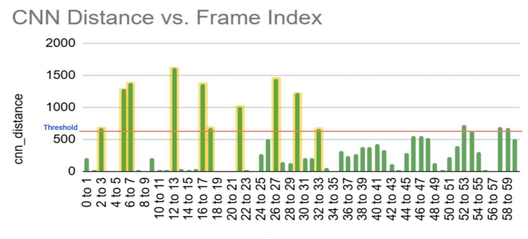



*(Figure 2) Optical Flow: Dense Flow Farneback method- Screen scrolling down*

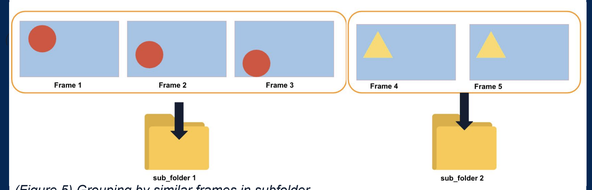



*(Figure 3) Optical Flow contd.*

## VISUALIZATION OF SEGMENTATION




*(Figure 4) CNN score between each consecutive frames*

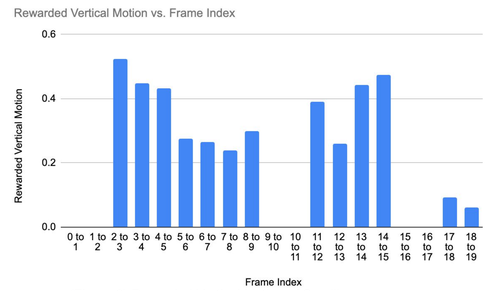



*(Figure 5) Grouping by similar frames in subfolder*

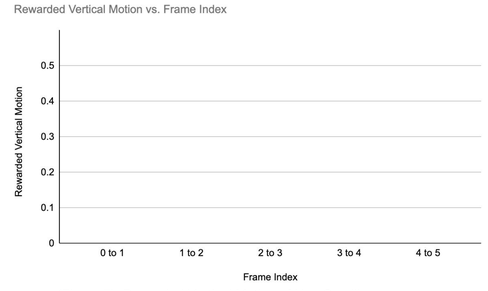
## FUTURE WORK

We plan to enhance all key components of our system: scroll detection, masking and subtraction, and mean shift. Particularly, we aim to improve the scroll detection module to handle a wide range of scrolling speeds, as extremely slow or fast scrolling can prevent accurate frame detection. Additionally, we plan to explore improved thresholding strategies to better balance false positives and false negatives, ensuring robust scene segmentation across diverse types of videos. Once these improvements are in place, our next step is to integrate these methods to develop a robust scene detection pipeline. Ultimately, we aim to apply this system to **ClassTranscribe**, our university's lecture video platform, to enable more accurate and efficient segmentation of lecture content.

## RESULTS




*(Figure 6) Rewarded Vertical Motion of Scrolling scene*




*(Figure 7) Rewarded Vertical Motion of Non-Scrolling scene*

## REFERENCES

[1] Angrave, L., Li, J., and Zhong, N. 2022. Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem. In Proceedings of the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN.

[2] OpenCV. Optical Flow. OpenCV Documentation. https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html

ILLINOIS

# A Multi-Method Approach to Scene Detection in Lecture Videos

**Sunwoo Baek, Supia Park, Ashley Li, Enya Chen, Charitha Nannapaneni**
**Siebel School of Computing and Data Science, University of Illinois Urbana Champaign**

## INTRODUCTION

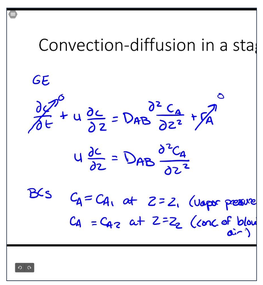
Lecture videos have a variety of forms, including a mix of static slides and dynamic annotations, which makes it challenging to extract keyframes that best represent the content. In this research we are addressing the specific problem: given an hour-long lecture video, how do we extract the key frames to create a concise PDF without redundant images from the lecture? Traditional methods like pixel similarity and OCR-based text extraction are not the most effective to capture gradual, non-discrete changes such as handwritten notes, highlights, and incremental annotations.

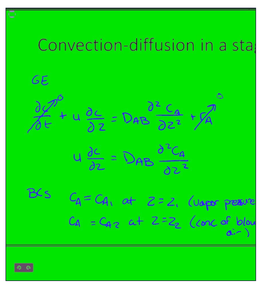
To address this, we explore a series of methods to complete the stated task. We utilize methods such as optical flow, masking and subtraction in order to efficiently and accurately select representative frames from a lecture video.

## HANDLING ANNOTATIONS

### Masking

For our masking technique, we masked each frame to remove the background. We did this through a series of color methods–first binning the pixels into 6 equally spaced bins in RGB space (simplifying our picture into 125 colors), then applying Otsu thresholding to obtain a mask of the foreground. Ideally, the remaining content is representative of the information present on each slide. This method reliably runs in O(nmk) time, (where nm represents the dimensions of the frame, and k represents the the amount of frames). This is roughly 0.21 seconds for a 1440x990 sized frame.




*(Figure 1) Original screenshot from a lecture video vs. (Figure 2) masked version.*

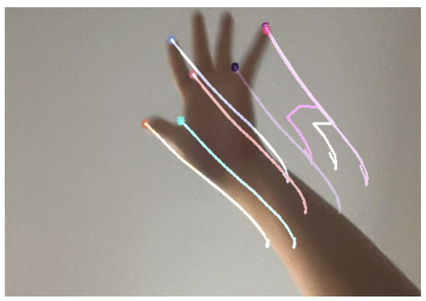
### Quantifying Differences for efficient frame extraction

Afterwards, we compared a current mask with a temporally previous mask and treated the difference representative of the amount of the unique information on the frame. Thus, we quantified the degree of unique information on each frame. With this metric, we will identify representative frames from a video containing annotations.

## HANDLING SCROLLING

### Exploring Optical Flow

By analyzing motion vectors between consecutive frames, optical flow allows us to track scrolling movements rather than relying on absolute pixel differences. Our experiments with Lucas-Kanade and Dense Optical Flow show that this method effectively detects the gradual evolution of content, and we are working on refining this approach to better differentiate meaningful updates (e.g., new annotations) from insignificant transitions (aka scrolling). However, challenges remain in filtering out background movement artifacts and distinguishing intentional content changes from noise. This approach provides a complementary perspective to static frame comparisons, offering a motion-based strategy for scene detection.

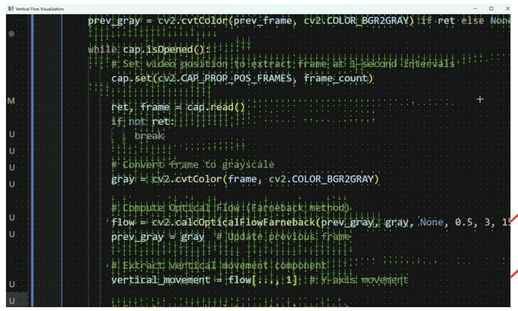


*Optical Flow: Lucas-Kanade method- Hand moving from Right to Left*

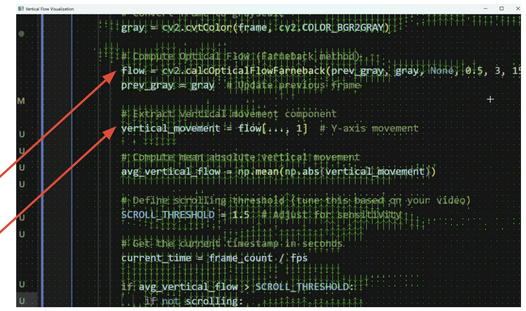
### Exploring Template Matching

Template matching takes a target to search for and a template to search in. It does this by sliding the target over the template and calculating a similarity value at every position. However, again potentially due to the homogeneity of lecture videos, the difference between a good and poor match may be difficult to distinguish, resulting in frequent incorrect matches. Thus, we also found this method ineffective.

## Experiments

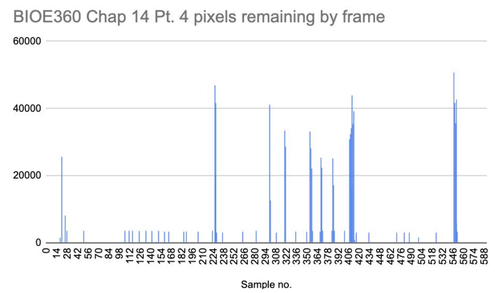



*(Figure 3) Optical Flow: Dense Flow Farneback method- Screen scrolling down*




*(Figure 4) Optical Flow contd.*

*(Figure 5) Meanshift: here, the brightest part should be around the middle–but it's not*




*(Figure 6) (Using a previous method) Example output graph of masking + subtraction. Spikes are potential scene changes*

## RESULTS

We first applied Lucas-Kanade Optical Flow, which estimates motion for a sparse set of feature points. While computationally efficient, this method struggled to capture broader visual changes, particularly during abrupt transitions, such as scene changes.

To address this, we implemented the Farneback method, which computes optical flow densely for all points in the frame. This provided a more comprehensive view of motion, enabling more accurate detection of gradual changes, such as scrolling.

Though dense estimation introduced some background noise, we applied thresholding techniques to filter out irrelevant motion. Overall, Farneback proved more effective for detecting meaningful content updates in lecture videos.

As seen in figure 5, meanshift did not reliably calculate a similarity metric between a target and source image. Template matching also yielded similar results where the desired heatmap was not created. Thus, we proceed with optical flow to detect scrolling.

We found our masking method to be more successful, creating clean masks of each frame. We opted for a simple single-color background removal since more complex backgrounds can also offer information.

## Future Approach

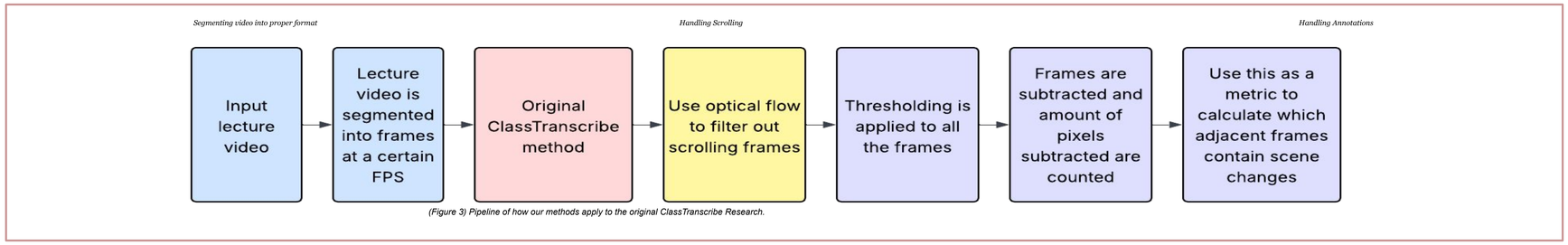
We plan to enhance all key components of our system: scroll detection, masking and subtraction, and mean shift. Once these improvements are in place, our next step is to integrate these methods to develop a robust scene detection pipeline. Ultimately, we aim to apply this system to **ClassTranscribe**, our university's lecture video platform, to enable more accurate and efficient segmentation of lecture content.

## References

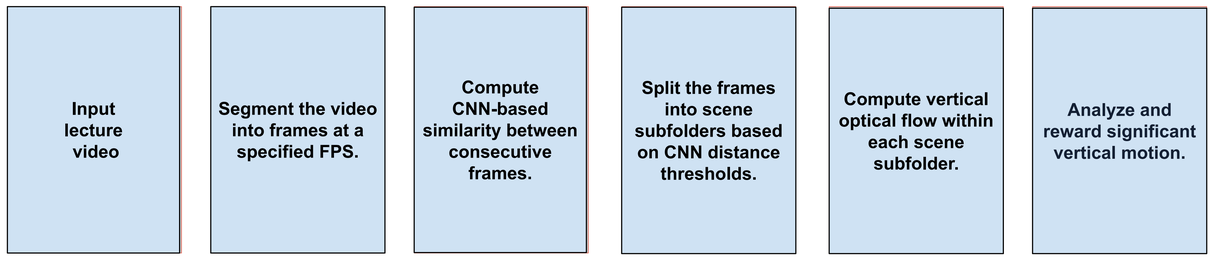
[1] OpenCV. Optical Flow. OpenCV Documentation. https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html

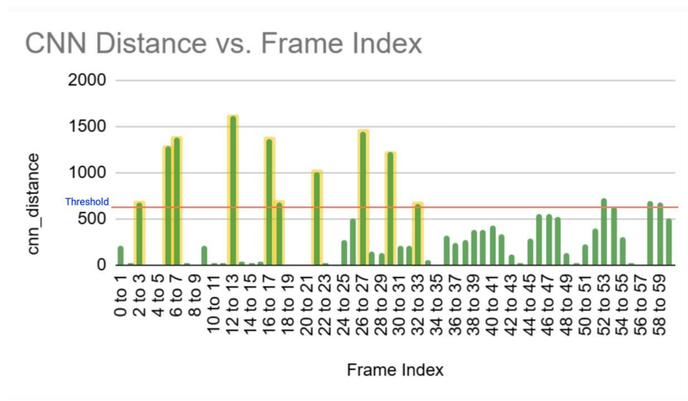
[2] Angrave, L., Li, J., and Zhong, N. 2022. Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem. In Proceedings of the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN.

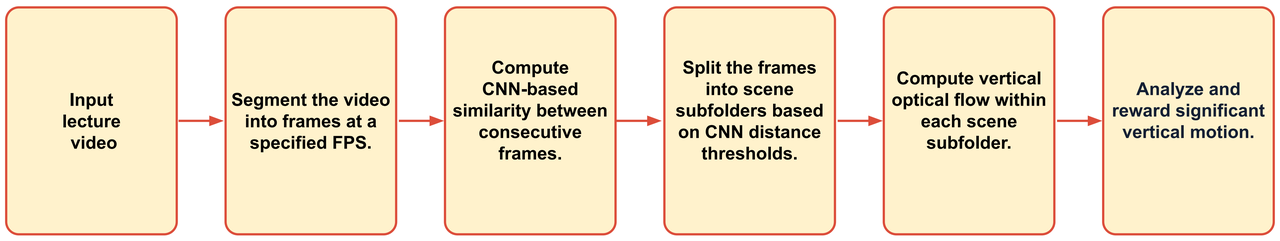
ILLINOIS

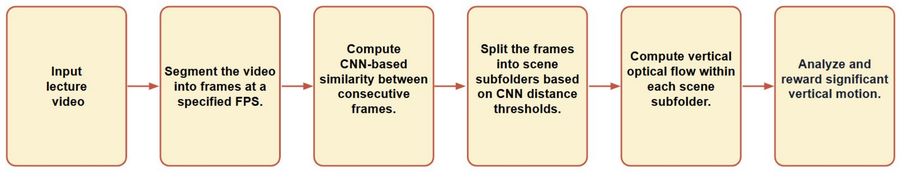


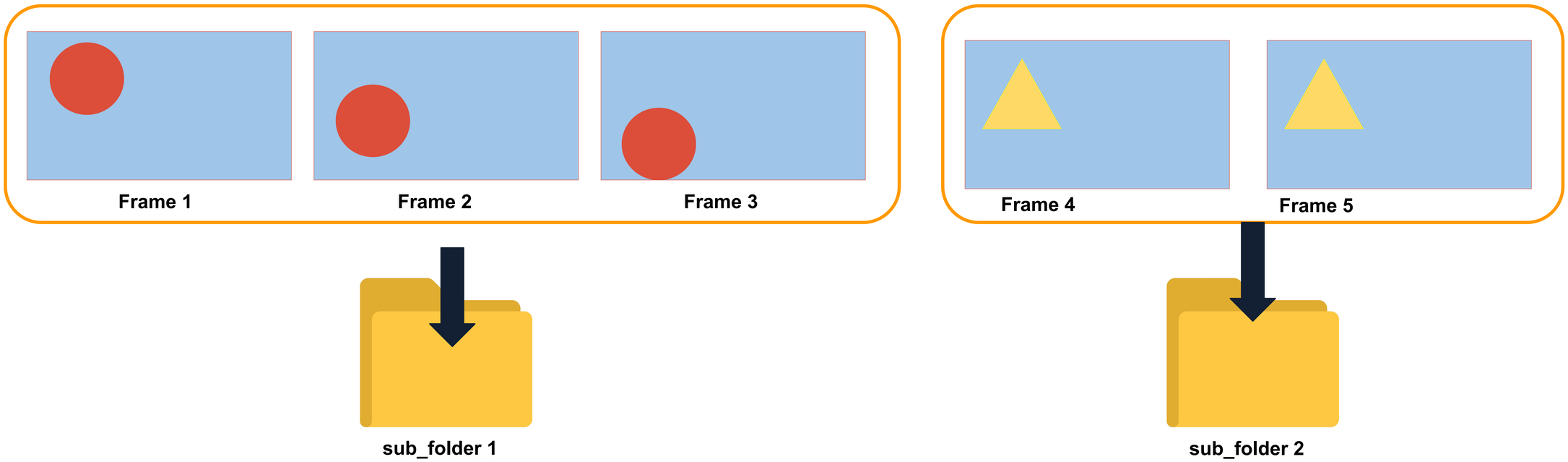

(Figure 3) Pipeline of how our methods apply to the original ClassTranscribe Research.




Frame 1
Frame 2
Frame 3
Frame 4
Frame 5
sub_folder 1
sub_folder 2