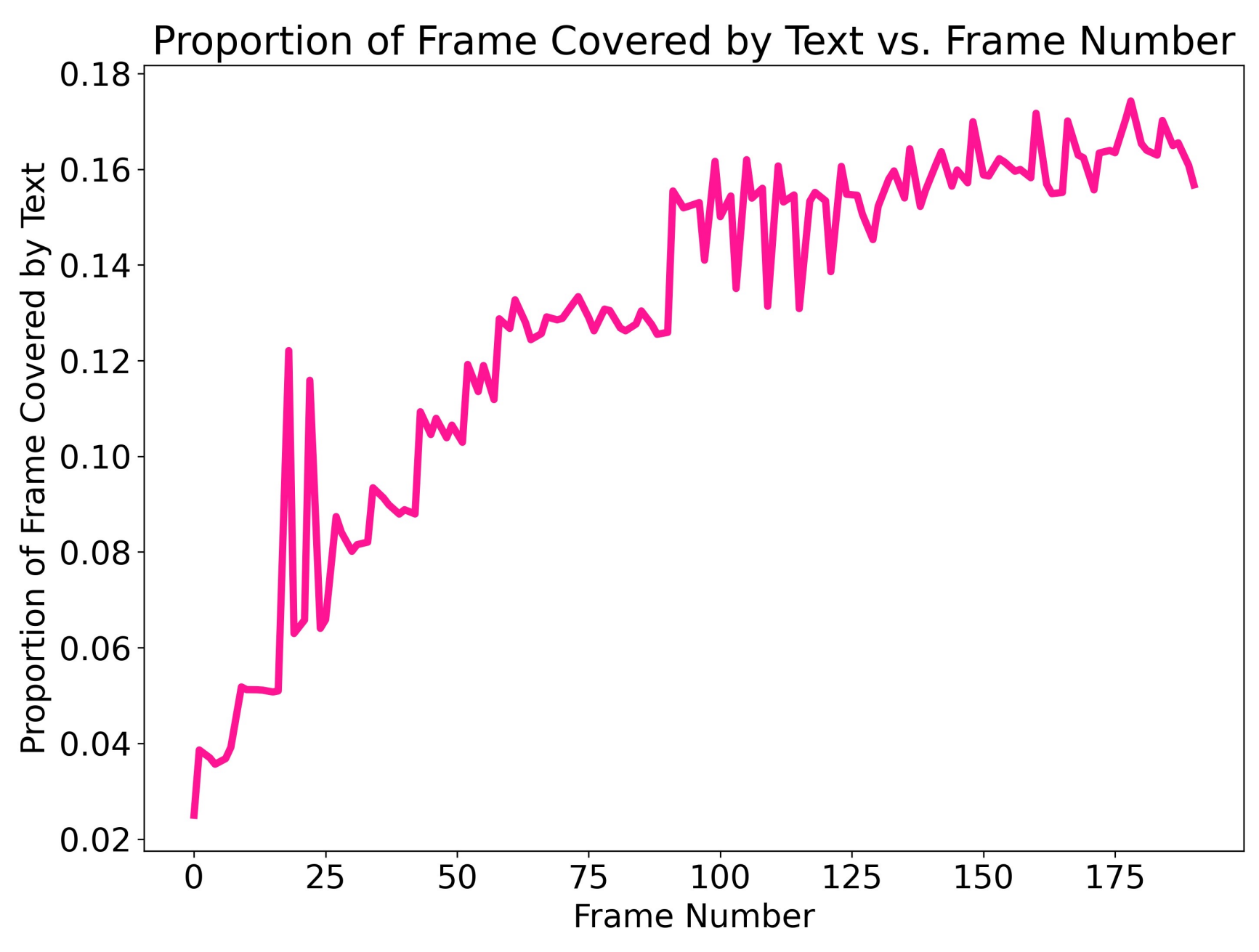
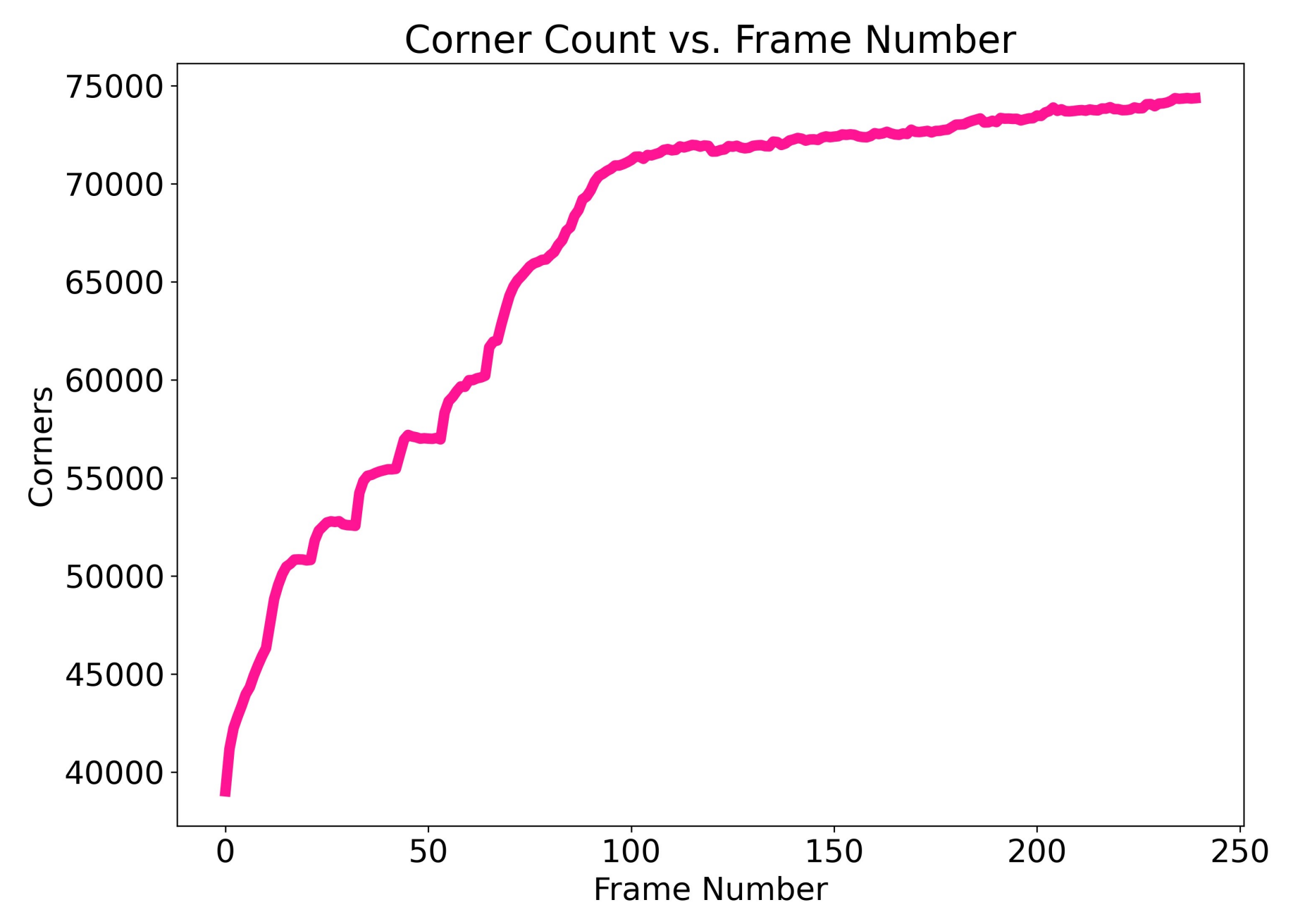
Efficiently Extracting Text From Political Ad Videos

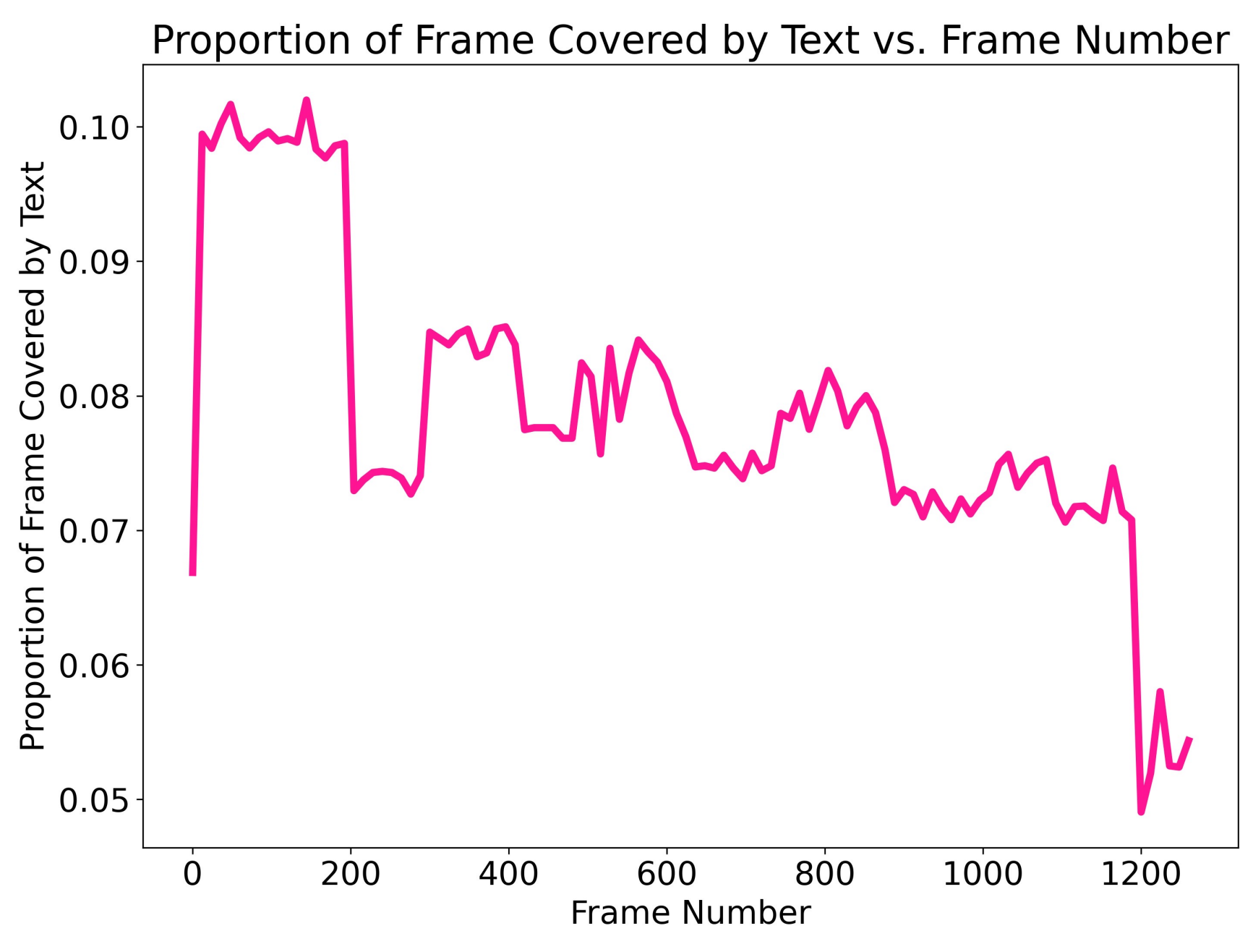
Aaron Foote
Advisor: Pavel Oleinikov

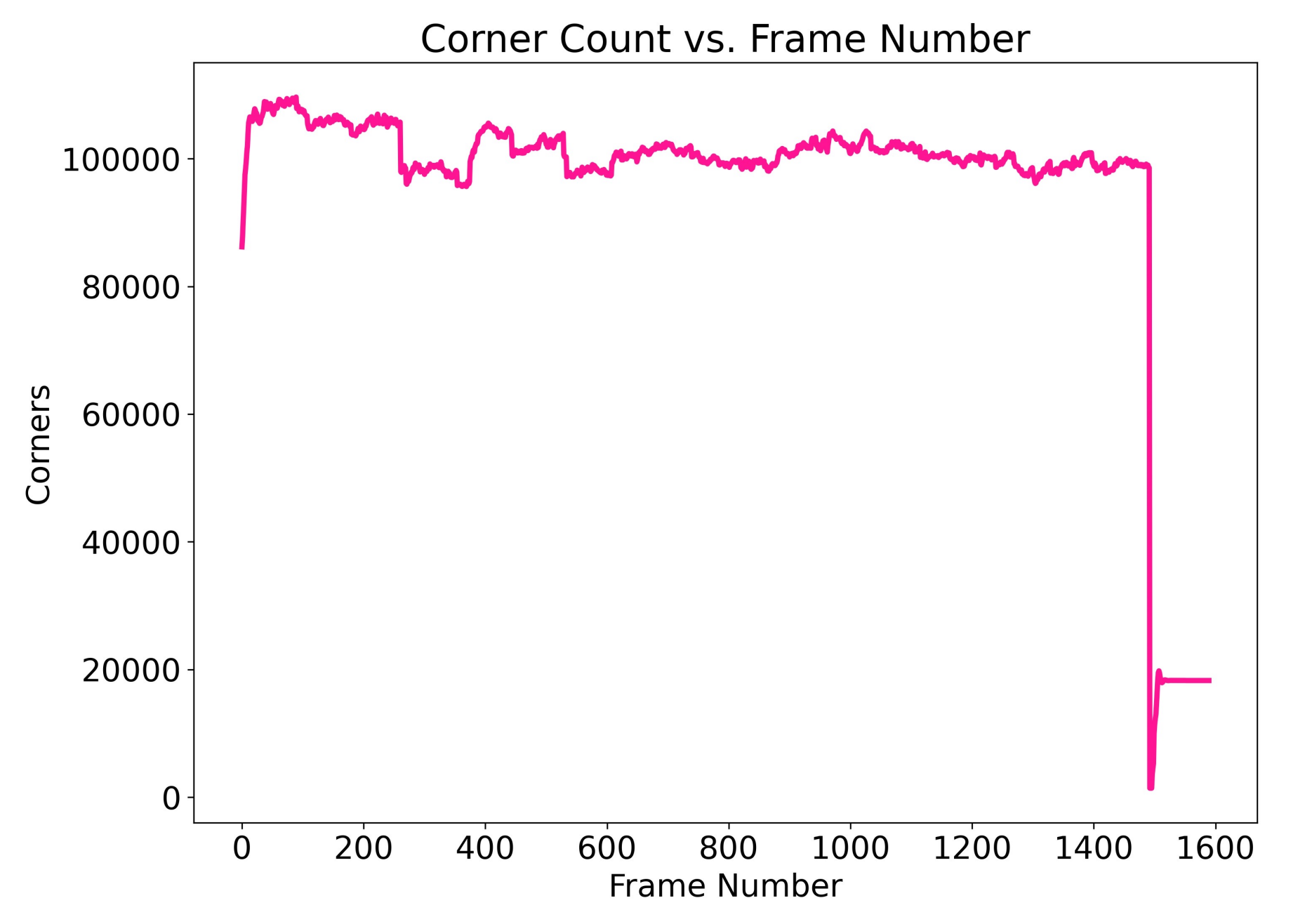
Background:

TV and online political ads frequently contain text in the frames. The goal of this project was to create a tool to extract the text for analysis.









Problem:

Videos can contain several thousands of frames so running OCR on every frame is not a reasonable approach. Instead, a method needs to be devised to identify the handful of frames that can be collected to accurately show the text from the entire video.

Identifying key frames rather than submitting the entire video to OCR was necessary because Google Vision costs $0.10 per minute but only $0.0015 for each photo.

Solution:

Construct a metric that reflects the amount of text in a given frame. Calculate this metric for every frame and extract the local maxima as key frames. Two metrics that were attempted were counting the number of corners in each frame versus calculating the proportion of each frame covered by text.

Findings:

Corner detection is a more robust approach. It is not disrupted by text that grows and shrinks, while the proportion of text approach is.

Additionally, the proportion of text approach is not useful for videos that don't change the region of the frame where text appears. This happens in videos where the text is the close captions of the words spoken in the video.

To determine the key frames, the local peaks of the corner count were used. The thought was that in a video, text is added to the frame until a new message is to be displayed. Then the frame is cleared and the process repeats again. Only the peak frame is needed because it includes the text in all of the frames in the increasing portion before the peak.

Post-Processing:

Since OCR is not perfect, taking more frames than one per peak is preferred. Then, clustering and cleaning can be performed for more accurate results.

Clustering of the text in the frames is done based on Levenshtein distance. This is a measure of how similar two strings are based on the number of edits (insertions, deletions, and substitutions) necessary to change one string to the other.

After grouping, some minor differences can be removed by getting rid of non-alphanumeric characters and leading/trailing whitespace. Finally, the most common words from each group are output.

Next Steps:

An improvement that could be made would be to prevent text that is present in the entire video from being output multiple times.