RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à Toulouse INP

Analyse des données multimodales enregistrées lors de visioconférences

Thèse présentée et soutenue, le 16 juin 2025 par

Travis SENG

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Vincent CHARVILLAT

Composition du jury
Mme Géraldine MORIN, Présidente, Toulouse INP
M. Laurent HEUTTE, Rapporteur, Université de Rouen Normandie
M. Mounim EL-YACOUBI, Rapporteur, Telecom SudParis
Mme Muriel VISANI, Examinatrice, La Rochelle Université
M. Vincent CHARVILLAT, Directeur de thèse, Toulouse INP

Membres invités
M. Axel Carlier, Toulouse INP
M. Wei Tsang OOI, co-encadrant, National University of Singapore

Summary

Video has become a central medium in education, with lecture videos playing an increasingly important role in knowledge dissemination. Lecture videos have evolved from simple classroom recordings to professionally produced MOOCs, reflecting wider shifts in both teaching methods and digital infrastructure. Despite their growing presence, lecture videos still pose challenges for learners and educators, particularly in terms of accessibility, navigability, and content engagement. These challenges are heightened by the multimodal nature of lecture videos, which combine visual, auditory, and textual elements. This thesis investigates how multimodal analysis can improve the usability and educational value of lecture videos. It builds on recent progress in deep learning, especially transformer-based models, to develop new techniques for analyzing and segmenting educational video content. The work makes two main technical contributions. First, it introduces a tool for generating synthetic slide datasets with detailed and consistent annotations. Existing slide datasets suffer from inconsistencies, incomplete labeling, and ambiguous class definitions —issues that compromise the training of layout and object detection models. The tool addresses this by producing diverse, high-quality synthetic slides that closely imitate real-world presentations. This synthetic data complements existing resources, reduces the need for manual annotation, and allows for tailored dataset creation depending on the research task. The tool supports the creation of standardized and customizable slide corpora, enabling more reliable and focused model training. Second, the thesis proposes a novel framework for temporal segmentation of lecture videos using multimodal embeddings. The method combines visual features from video frames, OCR-extracted slide content, and text embeddings from transcripts, aligning them through pre-trained vision-language models. By integrating transition cues and adapting to different lecture formats—such as screen recordings or blackboard sessions—the framework segments videos faster and more accurately than existing approaches. It achieves strong performance without relying on large-scale annotated training sets, making it both scalable and generalizable across diverse lecture types. Finally, a proof-of-concept application demonstrates how these methods can enhance video search, interactivity, and accessibility. Overall, this research contributes tools and methods that support more robust, multimodal processing of lecture videos, with the goal of making them more navigable, inclusive, and pedagogically effective.

Remerciements

I would like to express my deepest gratitude to my supervisors, Axel Carlier, Vincent Charvillat, and Wei Tsang Ooi, for their continuous support, patience, and knowledge. Their guidance has been invaluable throughout my research and the writing of this thesis. I am also grateful to my thesis committee, Mounim Al-Yacoubi, Laurent Heutte, and Muriel Visani, for their insightful comments, thoughtful discussions, and encouragement, which inspired me to broaden my research from different perspectives. My thanks also go to Géraldine Morin for kindly accepting to serve as president of my jury.

During my PhD, I had the opportunity to spend time in Singapore, and I am especially thankful to Wei Tsang Ooi for welcoming me there and for his constant help and support.

I also want to thank all the members of the REVA team for their friendship and for the many stimulating discussions we shared, often over coffee. Working in the lab was always a pleasure thanks to the people I met there. In particular, I would like to thank Thomas for his help in setting up my defense, as well as Rémy and Marie, with whom I shared the ups and downs of the PhD. It was always reassuring to walk this path together.

Beyond the lab, I am grateful to my friends who stood by me, always ready to support me and to help me disconnect from work when I needed it most.

Finally, I owe my deepest thanks to my family for their unconditional support and encouragement throughout my life. I am especially grateful to my parents, who always believed in me and supported me in all my choices. I owe them everything I am today. And of course, to my partner Sherry: thank you for being my confidant, my greatest support, and for always believing in me, no matter where in the world we were.

Contents

1 Introduction 11
1.1 Context and Importance of Lecture Videos in Modern Education 12
1.2 Challenges in Lecture Video Understanding 14
1.3 Potential of Analysis in Lecture Videos 16
1.4 The Need for Multimodal Analysis 17
1.5 Objectives and Contributions of This Research 17
2 State of the Art 19
2.1 Applications of Lecture Video Analysis 20
2.1.1 Content Summarization and Indexing 20
2.1.2 Interactivity and User Experience 27
2.1.3 Accessibility 29
2.2 Modalities in Deep Learning 31
2.2.1 Vision 31
2.2.2 Audio 34
2.2.3 Text 35
2.3 Transformers: A Unifying Architecture 37
2.4 Multimodal Large Language Models 40
3 Synthetic Slides Generation for Robust Slide Analysis 45
3.1 Introduction 46
3.2 Related Work 47
3.2.1 Existing datasets inconsistencies 50
3.3 Pipeline 53
3.4 Experiments 59
3.5 Qualitative results 64

3.6 Other types of annotations ... 67
3.6.1 Segmentation masks ... 67
3.6.2 Slide to Markdown ... 68
3.7 Discussion and Future Work ... 68
4 Temporal Video Lecture Segmentation ... 71
4.1 Introduction ... 72
4.2 Related Work ... 75
4.2.1 Text Segmentation ... 75
4.2.2 Temporal video segmentation ... 76
4.2.3 Lecture video segmentation ... 76
4.3 Datasets ... 78
4.4 System Overview ... 82
4.4.1 Preprocessing ... 82
4.4.2 Chunking ... 86
4.4.3 Audio Transcription and Matching. ... 86
4.4.4 Multimodal embeddings ... 86
4.4.5 Embeddings alignment ... 87
4.4.6 Clustering ... 87
4.4.7 Summary ... 89
4.5 Experiments ... 89
4.5.1 Experiment Setup ... 89
4.5.2 Evaluation Metrics ... 89
4.5.3 Findings on Yale Dataset ... 90
4.5.4 Baselines ... 92
4.5.5 Results on AVLectures Dataset ... 93
4.5.6 Results on VideoLectures and Yale Dataset ... 95
4.5.7 Qualitative Results ... 96
4.6 Ablation Study ... 99
4.6.1 Choice of contrastive loss ... 99
4.6.2 Use of modalities in the clustering process ... 100
4.6.3 Use of transition information during clustering ... 100
4.7 Discussion ... 102
4.8 Conclusion and Future Work ... 102
5 Lecture Video Analysis Toolkit ... 105
5.1 Introduction ... 106
5.2 Design considerations ... 107

5.3 System Architecture 109
5.4 Core Modules 110
5.5 User Interface 114
5.6 Conclusion and Future Work 119
6 Conclusion 123
6.1 Contributions 124
6.2 Future work 125
Bibliography 127

Chapter 1

Introduction

1.1 Context and Importance of Lecture Videos in Modern Education

Video has become a defining medium in today's digital landscape. Recent estimates suggest that over 80% of global internet traffic is now attributed to video content (Cisco 2022). This rapid growth has been enabled by significant technological advancements, particularly in video delivery systems. Adaptive streaming protocols, such as DASH (Dynamic Adaptive Streaming over HTTP) and HLS (HTTP Live Streaming), have made seamless playback possible across a wide range of devices and network conditions. Similarly, efficient codecs like H.265 (ITU-T Video Coding Experts Group (VCEG) 2013) and AV1 (Alliance for Open Media 2023) have substantially reduced the bandwidth required for high-quality video, broadening accessibility and driving global adoption. The democratization of smartphones has played a crucial role in this transformation, putting video recording technology in everyone's pocket and making videos accessible.

In education, technological advancements have significantly transformed the ways in which knowledge is shared. Lecture videos, in particular, have emerged as a valuable resource for both students and educators. Early examples were often unstructured classroom recordings (Figure 1.1a), uploaded to platforms such as YouTube with minimal editing. However, 2012, referred to by (Pappano 2012) as "The Year of MOOCs," marked a turning point with the emergence of Massive Open Online Courses (Figure 1.1d,e). These courses, designed to provide widespread access to high-quality educational content at little or no cost, rapidly gained popularity due to their accessibility, scalability, and potential to democratize education.

This development signaled a transition toward lecture videos that were intentionally designed and carefully structured to enhance learning outcomes. Platforms like Coursera, edX, and Udacity pioneered the delivery of curated content from leading institutions, complemented by supplementary materials, assessments, and certifications. This shift redefined the role of lecture videos, establishing them as comprehensive educational tools. Today, MOOCs enable learners worldwide—irrespective of geographic location—to engage with high-quality instruction from institutions such as MIT and Stanford.

The importance of digital education became especially apparent during the COVID-19 pandemic, which forced institutions to transition to online teaching with unprecedented speed. Synchronous tools, such as video conferencing platforms (e.g., Zoom), and asynchronous platforms, including Learning Management Systems (LMS), were rapidly adopted to maintain continuity (Turnbull et al. 2021). This transition, however, exposed existing disparities in infrastructure, digital literacy, and reliable internet access, particularly in under-resourced areas. Within this challenging context, lecture videos played a crucial role, offering a scalable and

flexible solution that allowed educators to continue teaching asynchronously, accommodating students in diverse circumstances.

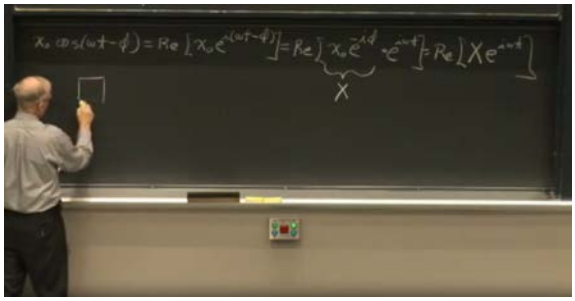
One of the greatest strengths of lecture videos is their adaptability. They provide learners with the ability to revisit and review material at their own pace, which is particularly valuable for complex or technical subjects. (Robertson and Flowers 2020) note that when paired with supplementary materials, such as instructor notes or annotated slides, lecture videos significantly improve comprehension and retention. Moreover, their accessibility benefits a wide range of learners, including individuals with disabilities and non-native speakers, who can tailor the learning experience to suit their unique needs.

Despite their benefits, the design and production of lecture videos remain critical factors in their effectiveness. (Guo et al. 2014) show that shorter, more focused videos, informal "talking-head" presentations, and visual aids such as tablet drawings tend to engage students more effectively than traditional formats like static slides or long classroom recordings.

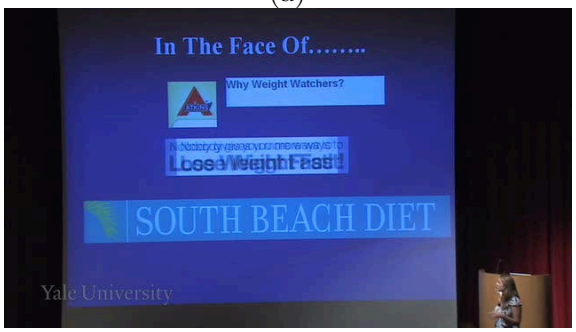
However, engagement alone is not enough. Lecture videos face other challenges, such as the risk of cognitive overload, barriers to accessibility, and limited opportunities for interactivity. Poorly designed videos can overload learners' cognitive resources, making it harder to integrate new information (Costley et al. 2021), while the lack of interactivity further reduces attention and engagement over time. Adding interactive elements has been shown to significantly increase viewer retention and attention spans, even for longer videos (Geri et al. 2017). Addressing these challenges is essential for enhancing their educational impact and ensuring they serve the diverse needs of learners around the globe. As education continues to evolve in an increasingly digital context, it is imperative to explore solutions that maximize the potential of lecture videos as transformative tools for learning.



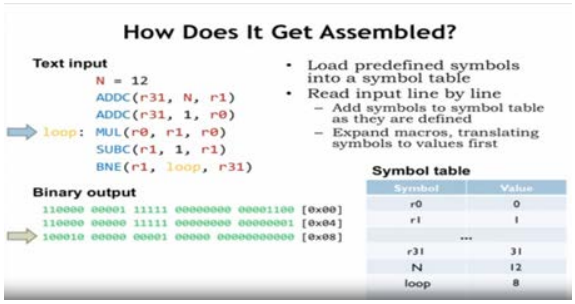
(a)



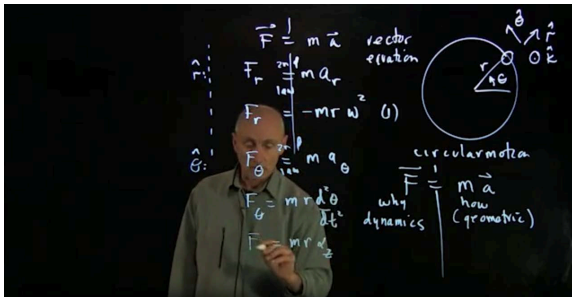
(b)



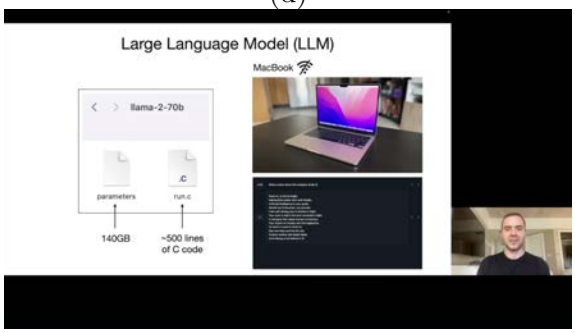
(c)



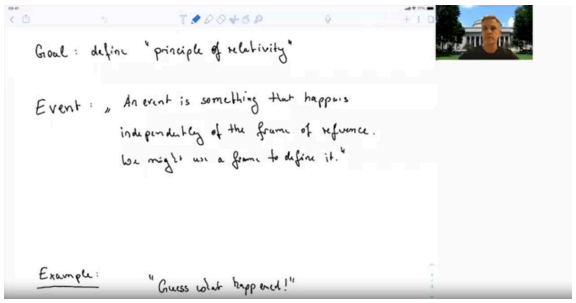
(d)



(e)



(f)



(g)



(h)

Figure 1.1 – Different types of lecture videos ranging from blackboard to slide-based presentations to more produced videos.

1.2 Challenges in Lecture Video Understanding

Lecture videos encompass a wide range of formats, shaped by pedagogical practices and evolving recording technologies. As illustrated in Figure 1.1, these formats have evolved from

raw, unstructured recordings of live classroom sessions to polished and intentionally produced presentations, such as TEDx talks. This progression reflects a broader shift toward enhancing the clarity, engagement, and accessibility of educational content. Each format carries distinct characteristics that influence how information is delivered, consumed, and analyzed, posing unique challenges for interpretation and automated processing.

Figures (a) and (b) show traditional blackboard-based lectures, where instructors write and explain the material as they go. This format supports a step-by-step development of ideas, reflecting the natural flow of a live classroom. At the same time, it poses challenges for automated analysis—such as recognizing handwritten content, handling occlusions caused by the instructor's movement, and recovering partially erased information.

Figure (c) captures a traditional lecture style where the presenter is filmed standing next to their slides. This format often features a fixed frame that juxtaposes the instructor with their visual aids. Analytical strategies for this format must account for the shared focus between the instructor's gestures, which reference specific slide content, and the extraction of static visual elements.

Slide-based presentations, as shown in Figures (d) and (f), reflect a shift toward pre-prepared, structured content, offering visual clarity that allows instructors to emphasize key points. Figure (f) builds on this format by incorporating the instructor's camera feed, typically displayed in a corner of the screen, creating a more engaging and personable experience. This approach allows students to connect with the instructor's expressions while maintaining focus on the slide content. Despite their visual organization, these formats demand techniques for analyzing static text, images, and diagrams, often in conjunction with contextual cues such as the instructor's speech or mouse movements.

Figure (e) represents a hybrid approach that integrates real-time interaction with enhanced visual production. Using a lightboard, the instructor writes on a transparent surface while maintaining direct eye contact with the audience. This setup merges the immediacy of blackboard teaching with the production quality of pre-recorded content, making it particularly effective for online and asynchronous learning.

Figure (g) introduces an approach frequently seen in online classes, combining a blackboard-style presentation with videoconferencing tools. In this format, the instructor uses digital writing tools to explain concepts, often while appearing in a corner of the screen through a live video feed. This setup merges the interactive and dynamic elements of blackboard teaching with the accessibility and flexibility of online learning. Analytical challenges here include segmenting the instructor's annotations, integrating the visual cues from the blackboard, and interpreting speech in tandem with written content.

Finally, Figure (h) shows highly produced, performance-oriented lectures, such as TEDx-style talks. These presentations rely on professional staging, lighting, and minimal visual aids, placing a strong emphasis on the speaker's delivery and audience engagement. While engaging, this format shifts the analytical focus toward speech processing and speaker movement, as visual content often plays a secondary role.

These formats reflect the progression of lecture delivery methods, from traditional classrooms to digital and hybrid environments. Their diversity highlights the need for flexible analytical techniques capable of addressing the distinct challenges each format presents, whether it involves extracting handwritten content, interpreting multimodal interactions, or analyzing speech in performance-driven presentations.

For students, the length and density of lecture videos often make it difficult to stay engaged and fully absorb the material. As (Lange and Costley 2020) observed, recurring issues such as pacing, clarity, video quality, lack of diverse media, and inconsistencies in presentation further complicate the learning experience. Unlike in a classroom, where questions can be asked and answered in real time, students watching lecture videos are left to navigate challenging concepts on their own. Without opportunities for interaction or tailored feedback, many learners face cognitive overload, which can limit their ability to focus and retain information.

Instructors, on the other hand, struggle to get useful insights from how students interact with lecture videos. Knowing how students watch and engage with the content is key to improving teaching strategies, but the tools to gather and analyze this kind of data are still limited. As (Giannakos et al. 2015) point out, although online video platforms generate valuable clickstream data, this information is rarely made accessible to instructors or researchers, limiting their ability to understand and improve video-assisted learning.

Accessibility is another big issue—many lecture videos don't have captions, transcripts, or other features that make them usable for people with disabilities, which reduces their inclusivity and impact.

1.3 Potential of Analysis in Lecture Videos

Analyzing lecture videos presents an opportunity to address their inherent challenges by unlocking meaningful insights and enhancing their usability for learners and educators. At the core of this potential is the ability to process and extract information from multiple dimensions of these videos, including text, audio, and visuals. By leveraging these modalities, advanced analytical techniques can enable transformative applications, such as:

- Content Summarization and Indexing: Automatically segmenting long lecture videos into coherent sections and generating searchable keywords to improve navigation and accessibility. (Adcock et al. 2010a, Biswas et al. 2015, Mahapatra et al. 2018)

- Interactivity Enhancements: Enabling interactive features like clickable transcripts, embedded quizzes, and hyperlinked references to foster engagement and active learning. (Kim et al. 2022a, Weng et al. 2022)
- Accessibility Improvements: Producing captions, transcripts, and translations to make lecture content more inclusive and usable for diverse audiences, including those with disabilities. (Peng et al. 2021a, 2021b; Sechayk et al. 2024)

Lecture videos often rely on visual elements—such as slides, diagrams, and presenter gestures—to convey key information, making visual analysis essential for effective content understanding. Accurately identifying and categorizing components like titles, text blocks, and illustrations is critical for improving indexing, navigation, and the overall learning experience. Chapter 3 explores these challenges in depth, particularly the difficulty of recognizing diverse visual elements within the dynamic context of a lecture.

Another important area is video segmentation (explored in Chapter 4), where techniques to delineate logical sections—such as topic transitions or shifts in presentation style—can greatly enhance the usability of long lecture recordings. Developing more precise segmentation methods can support learners in finding relevant information easier and faster.

1.4 The Need for Multimodal Analysis

While single-modality approaches (e.g., focusing solely on visual or textual data) can provide insights, they fall short of capturing the complexity of lecture videos, where multiple modalities interact simultaneously. Visual data from slides, auditory information from the instructor's voice, and textual content—whether from spoken words or displayed text—all contribute to the richness of the lecture experience. Addressing this interplay requires a multimodal approach that integrates these diverse streams of information to produce a better understanding.

Recent advancements in deep learning, particularly in transformer-based architectures (Vaswani 2017), have introduced a unified approach for processing multiple data modalities within a single model. This thesis builds on these developments to explore new techniques for multimodal analysis in lecture video understanding.

1.5 Objectives and Contributions of This Research

This thesis presents three main contributions. The preamble, marked as Chapter 2, provides context for the following chapters by reviewing the state of the art in lecture video analysis applications. It also covers recent advances in deep learning, with a focus on transformer-based models that enable multimodal processing. Chapter 3 introduces a tool for generating synthetic slide datasets with rich annotations, designed to support the development and evaluation of multimodal models. Chapter 4 proposes improved methods for lecture video segmentation by

enhancing the identification of coherent topic boundaries, facilitating more effective indexing and navigation. Chapter 5 presents a proof-of-concept application that aims to improve interactivity, accessibility, and understanding in lecture videos. It builds on our contributions while also addressing limitations in existing solutions within the domain.

Chapter 2

State of the Art

2.1 Applications of Lecture Video Analysis

As discussed in the Introduction, lecture videos have become an essential part of modern education, but their effectiveness is often limited by challenges such as long durations, difficulty navigating content, and accessibility barriers. Addressing these issues requires solutions that go beyond the traditional use of lecture videos as passive learning tools. Most existing efforts rely heavily on extracting both visual and audio elements from lecture videos—such as slide content, diagrams, presenter information, and speech transcripts—to drive meaningful applications.

In this section, we examine prior work in this domain, highlighting how researchers and developers have used automated analysis to enhance the lecture video experience. By exploring these applications, we aim to situate our work within the broader landscape of solutions designed to transform lecture videos into more effective educational resources.

2.1.1 Content Summarization and Indexing

When searching online for an explanation of "neural networks," learners are often presented with a long list of hour-long lecture videos. Identifying the specific moment that provides the needed information can become time-consuming. Lecture videos are an incredibly rich source of information, but their length and density can overwhelm learners, making it difficult to focus on specific topics or locate relevant content.

Content summarization and indexing address this challenge by breaking down videos into structured, meaningful segments. These techniques not only help learners navigate content more effectively but also make complex material easier to comprehend. Summarization distills key points, while indexing provides a roadmap to specific concepts, transforming lecture videos into accessible, learner-friendly resources.

One such solution is *TalkMiner* (Adcock et al. 2010a), a specialized search engine for lecture webcasts that indexes videos by extracting textual data from presentation slides and associated metadata such as titles and abstracts. The system begins by identifying distinct slide images within lecture videos using frame-differencing algorithms. These algorithms detect significant changes in frame content to isolate slide transitions while addressing challenges like speaker appearances, picture-in-picture setups, and room background. Spatial analysis is employed to focus on central regions of frames where slides are usually displayed, and speaker appearance modeling with support vector machines excludes frames dominated by non-slide content. After identifying slides, Optical Character Recognition (OCR) is applied to extract textual information, which is indexed alongside metadata. A custom retrieval framework enables keyword-based searches, linking query results to specific timestamps within the lecture videos.

Building on this, *LectYS* (Yoo et al. 2021), focuses on subtitles to produce time-stamped summary, this time, of YouTube lecture videos. Using Latent Semantic Analysis, the authors identify the most semantically relevant sentences from subtitles. Then, by providing a keyword, students can jump directly to the sections containing the keyword, bypassing irrelevant content and saving time.

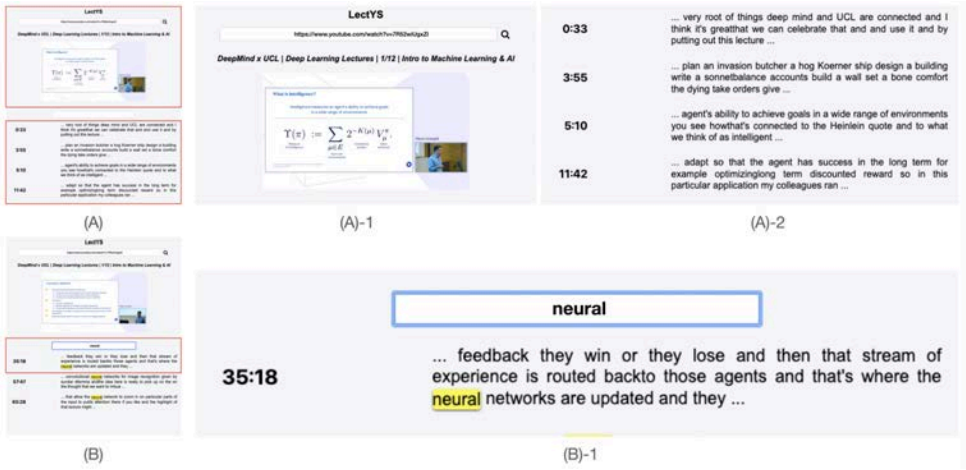


Figure 2.1 – Interface of LectYS (Yoo et al. 2021). LectYS let users search keywords in Youtube videos

Beyond indexing, techniques for content summarization aim to provide an overview of key points in a lecture. (Biswas et al. 2015) introduce a method for generating tables of contents (ToC) for slide-based lecture videos by combining visual saliency of slide elements and textual saliency from spoken transcripts. Visual saliency is quantified using features such as boldness, capitalization, location on the slide, and font size, which are extracted via OCR and processed using a Rank-SVM model trained on human-annotated data to assign weights to each feature. Spoken word saliency is computed using TextRank (Mihalcea and Tarau 2004), which measures the co-occurrence of terms in the transcript within a contextual window. The framework integrates these modalities by linking visual and textual elements through cosine similarity of contextual vectors, aligning synonyms, and resolving coreferences. The segmentation process is formalized as an optimization problem using a dynamic programming approach, where a cost function minimizes the commonality of salient words between adjacent segments while maximizing their distinctiveness.

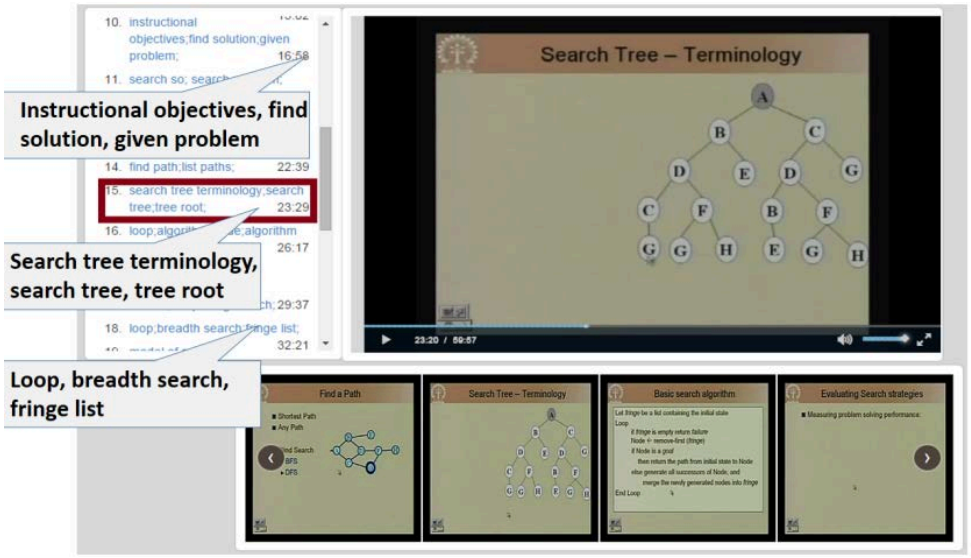


Figure 2.2 – Example of a Table of Content generated by (Biswas et al. 2015)

In the same vein, *VideoKen* (Mahapatra et al. 2018) also demonstrates the utility of multimodal analysis for summarization and indexing in educational videos by generating Multimedia Tables of Contents and phrase clouds. The MMToC integrates visual, audio, and textual modalities, leveraging Optical Character Recognition and Automatic Speech Recognition using off-the-shelf algorithms to extract text and compute their saliency in a similar way as (Biswas et al. 2015) from video frames and speech. However, they complement the idea by introducing phrase clouds to organize key concepts by topic, enabling users to interactively locate relevant content within the video timeline in a non-linear way by clicking on them. The phrase clouds are generated by ranking sentences from the transcript using TextRank (Mihalcea and Tarau 2004), then embeddings are computed with the Word2Vec (Mikolov et al. 2013) model. Finally, the system evaluates their significance based on their distance from predefined domain-specific dictionaries followed by clustering with DBSCAN. The platform also supports customizable playlists, further emphasizing its adaptability in creating focused learning modules from diverse video sources. Their interface can be seen in Figure 2.3.

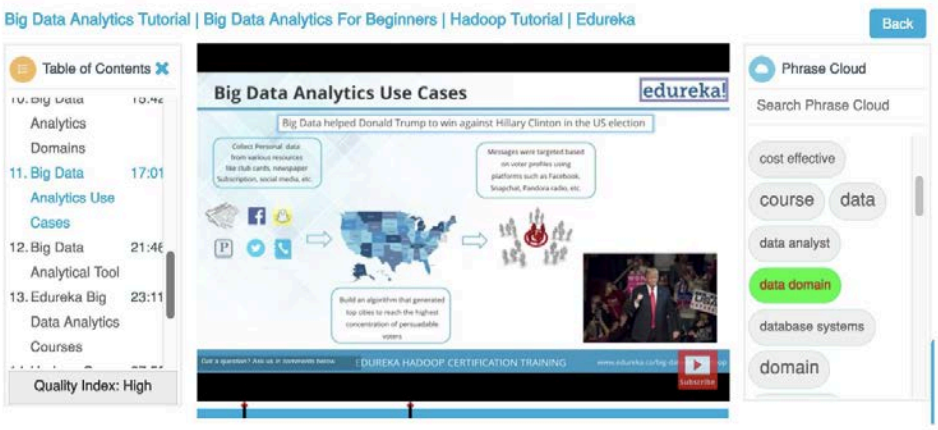


Figure 2.3 – System of (Mahapatra et al. 2018) that provides a table of contents (on the left) and phrase clouds (on the right).

(Zhao et al. 2019a) extend this concept by enriching the navigation experience with a rich visual interface for slide-based lecture videos by integrating multiple semantic and structural clues derived from slides, transcripts, and cursor movements. The system processes videos in three main steps: Slide extraction using edge-based transition detection, where a Canny edge detector identifies key transitions between frames; Content entity segmentation and classification, first by detecting connected components then employing OCR tools and a supervised learning framework to identify and classify slide elements such as text blocks, mathematical expressions, and diagrams; and Speech recognition and correspondence, using automatic speech recognition to get audio transcript and associate them with slide entities based on textual similarity, mouse movements, and slide progression. However, their content entity segmentation and classification fails on more complex slides with a lot of elements and when background is not uniform. The interface is displayed in Figure 2.4.

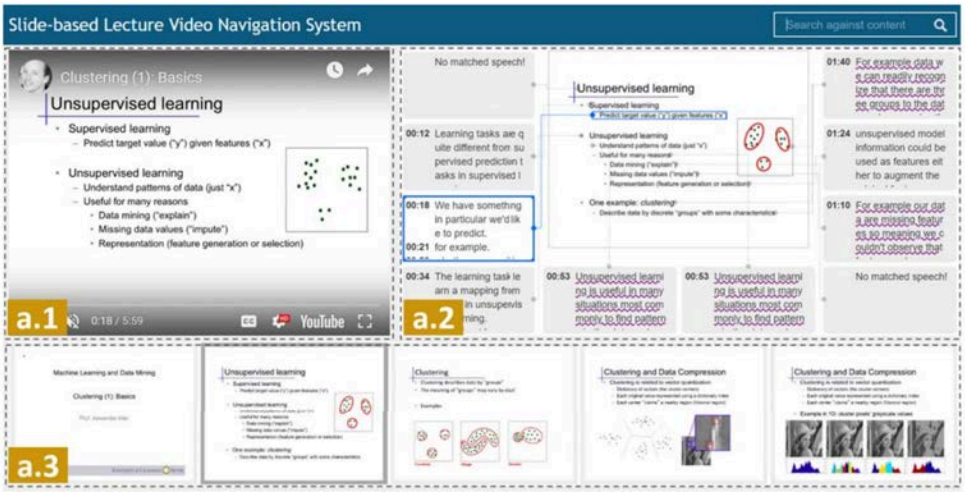


Figure 2.4 – System of (Zhao et al. 2019b) that let users search a keyword within the slides and the transcript. The sequence of slides, the transcript and the corresponding visual entities are displayed to the user to easily skim through the videos.

In contrast, (Rahman et al. 2020) take a simpler approach, summarizing lecture videos into a few key visuals, focusing on providing users with a quick and clear overview by selecting the most representative images from transition frames. Transition frames are identified by detecting significant scene changes in the video. Visual entities are extracted using a sliding window protocol that detects distinct visual regions based on pixel continuity. The extracted images are analyzed for similarity using a distance matrix, which quantifies dissimilarity between image pairs. Local similarity is calculated by matching unique keypoints detected via Scale-Invariant Feature Transform, while global similarity is assessed through affine transformations to measure alignment between images. The importance of each image is evaluated based on its size, information density (keypoints per unit area), and display duration in the video, combined into a weighted importance score. A heuristic graph-based algorithm then selects a subset of representative images by minimizing the distance between images outside the summary and their closest representative within the summary, while prioritizing higher importance scores.

(Koka et al. 2020) achieved a similar work but this time, to select representative keywords from slides by determining their importance. Keyword importance is computed using a weighted combination of term frequency, inverse document frequency (IDF), font size, time on screen, domain relevance, and rare word detection. Term frequency and IDF identify significant terms specific to a segment, while font size and screen duration emphasize prominent text. Domain dictionaries ensure contextual relevance, and rare word detection filters out unlikely candidates. These factors are combined into a normalized score to rank and select the most representative keywords for each lecture video segment. These two works has been implemented in one interface displayed in Figure 2.5 where an example of selected representative images and keywords is shown.

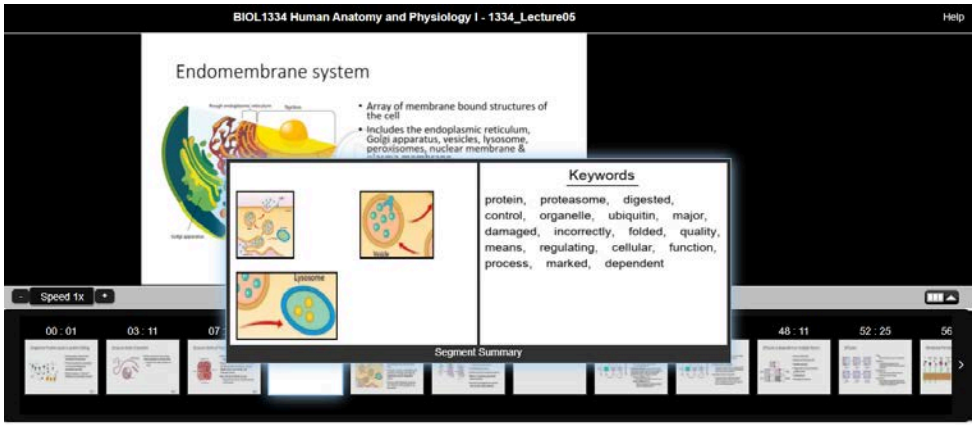


Figure 2.5 – Visual index of representative images and keywords from (Rahman et al. 2020) and (Koka et al. 2020)

Different from the previous works, *Lecture2Note* (Xu et al. 2019) takes a different approach by transforming slide-based lecture videos directly into structured notes by aligning visual entities with their corresponding speech content. Visual entities (text, formulas, graphs) are first extracted using binarization, morphological operations, and Run-Length Smoothing Algorithm. Then the system classifies these entities by text, formula or graph via a Support Vector Machine trained on features such as aspect ratio, letter ratio, and baseline fluctuation. Semantic alignment is achieved using Word Mover's Distance to measure similarity between word embeddings from pre-trained models (Bojanowski et al. 2017). Then, key visual entities are paired with the speech. The notes are organized using a Treemaps-based layout, allocating space based on entity importance derived from speech duration and textual density.

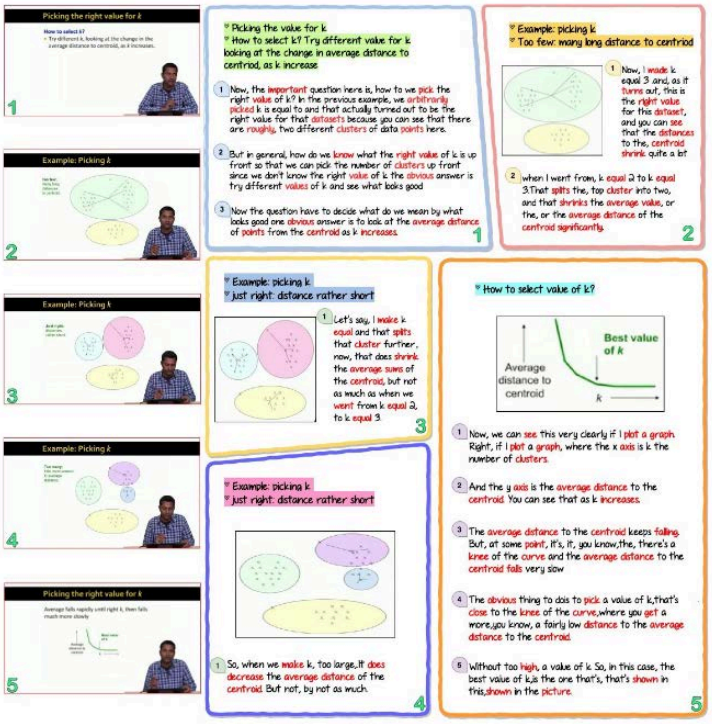


Figure 2.6 – Left: The original video. Right: Example of notes generated from the video. (Xu et al. 2019) detects the visual entities (text and images) from the slide and generates notes using those entities and the transcript.

While the focus so far has been on applications for slide-based video lectures, blackboard-based lectures also represent a significant area of research due to their distinct challenges. Unlike slides, blackboards involve dynamic handwritten content, often requiring specialized methods to process and summarize. The main goals is to summarize

For such lecture, two approaches stand out: *Automated Whiteboard Lecture Video Summarization by Content Region Detection and Representation* ((Kota et al. 2021)) and *FCN-LectureNet* ((Davila et al. 2021)).

The first employs a region-based framework that segments video frames into distinct handwritten content areas, relying on techniques like multi-scale histograms of gradients and deep metric learning for feature representation. These methods account for occlusions caused by speaker movements and gradual content changes. Weighted spatio-temporal conflict minimization is used to segment videos into topically coherent sections, ensuring that the generated summaries are both concise and representative. The key output includes both "key object" summaries and keyframes, providing flexibility in summarization granularity.

FCN-LectureNet, on the other hand, uses a fully convolutional network (FCN) to extract handwritten content as binary images, bypassing the need for complex preprocessing. The FCN comprises three branches: background estimation, text mask prediction, and binarization. Unique handwritten components are tracked over time to construct a spatial-temporal index,

which informs both temporal segmentation and summary generation. FCN-LectureNet emphasizes high recall and precision in handwritten content extraction, ensuring that all key content is preserved in the summaries.

2.1.2 Interactivity and User Experience

Watching a lecture video can often feel like a passive experience, with limited ways to engage, ask questions, or explore related concepts. This lack of interactivity can hinder understanding and make learning less effective. To address this, research has focused on improving interactivity and user experience through features like in-video quizzes, clickable annotations, and interactive transcripts. One such example is the system described in Redesigning the Online Video Lecture Player to Promote Active Learning (Walk et al. 2020). This custom video player includes features specifically designed for lecture videos, such as a keyword search function, allowing students to quickly locate specific topics or segments. It also tracks user interactions—such as play, pause, and seek events—providing instructors with data on how students engage with the material and highlighting sections that might be challenging. The integration of quizzes within the video further supports active learning by encouraging students to reflect on the content as they watch.

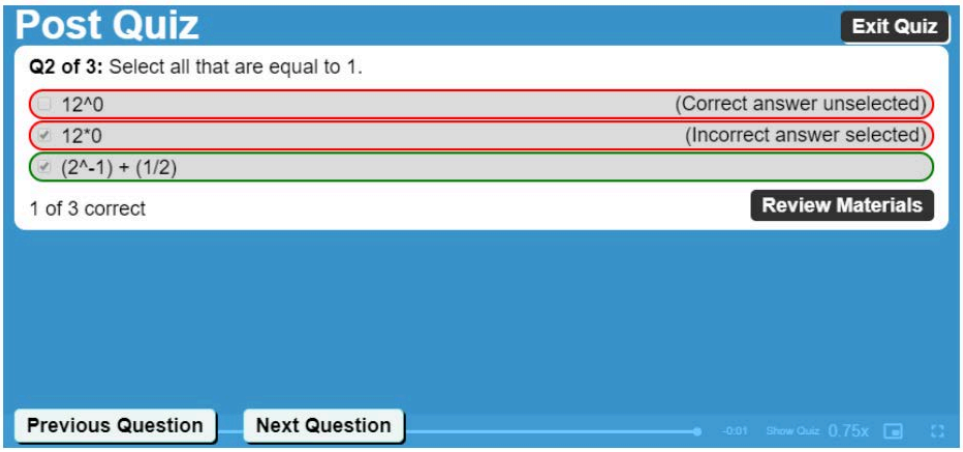


Figure 2.7 – A sample quiz question and answer feedback from (Walk et al. 2020). A quiz can be integrated in the player to promote active learning.

(Weng et al. 2022) propose *HiVideo*, a hierarchical browsing interface designed to enhance the interactivity and usability of educational videos, also accessible on mobile devices thanks to a responsive design. By leveraging a combination of deep learning-based document layout analysis and frame comparison algorithms, the system extracts and synchronizes slide content with video playback. This approach enables users to navigate non-linearly through video content by clicking on specific slide elements, which are linked to corresponding timestamps. The interface includes three layers: a poster layer for high-level summaries, a slide layer for detailed textual exploration, and a video layer for synchronized playback seen in Figure 2.8. The slide detection

relies on frame hashing techniques, while textual content is extracted using OCR and speech recognition technologies.

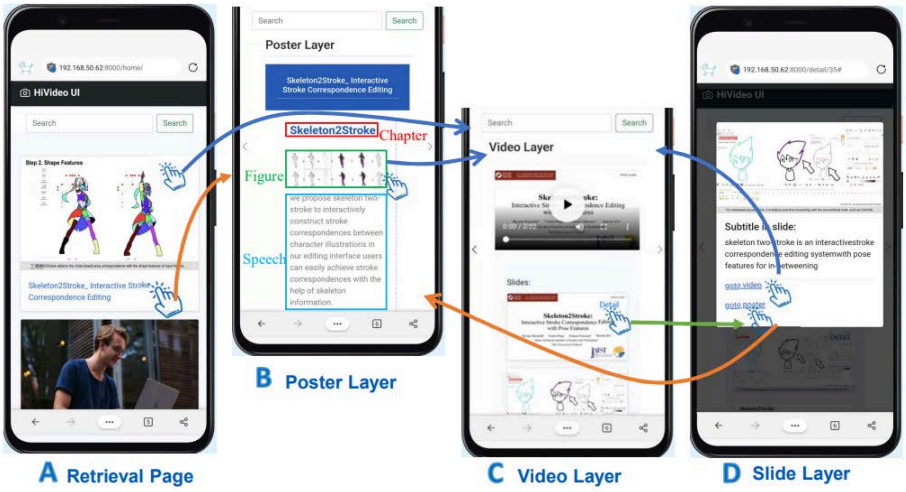


Figure 2.8 – HiVideo (Weng et al. 2022) interface with retrieval page (A), poster layer (B), video layer (C) and slide layer (D).

Adapting slide-based lectures to mobile devices, this is what (Kim et al. 2022a) present in *FitVid: Towards Development of Responsive and Fluid Video Content Adaptation*, which focuses on adapting slide-based lecture videos for different screen sizes. FitVid detects and extracts visual elements, such as text and images from slides thanks to an object detection model trained on their dataset. Then, their system aligns the visual entities with audio narration by computing embeddings similarity. This allows to adapt the visuals to the narration dynamically. It also enables the flexible resizing, repositioning, and adjustment of the visual content to fit various screen sizes and user preferences. Additional features, such as dark mode, template switching, and direct manipulation of content, make it easier for learners to tailor the presentation to their needs.

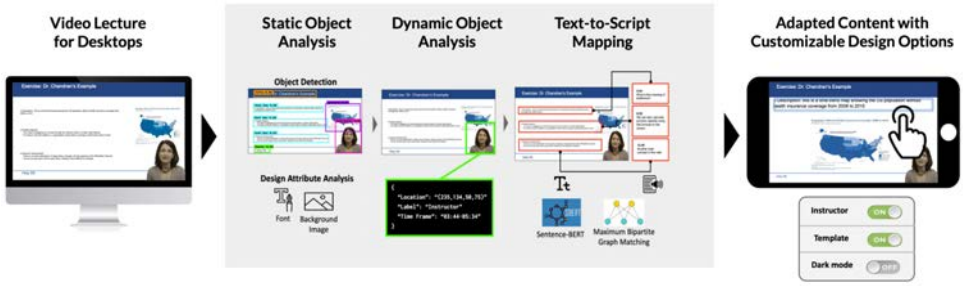


Figure 2.9 – System of (Kim et al. 2022a). Their approach allows slide-based videos to be tailored to different device sizes and provides customizable options to enhance user experience.

2.1.3 Accessibility

Accessibility in educational videos focuses on making learning materials more inclusive for individuals with disabilities, attention challenges, or language barriers. Features like captions, audio descriptions, and adjustable playback speeds ensure that content is available to a broader audience, promoting equal access to educational opportunities.

For instance, (Peng et al. 2021a) present *Slidecho: Flexible Non-Visual Exploration of Presentation Videos* to enable non-visual access to slide content by extracting and aligning slides with the presenter's speech. The system offers audio notifications about slide changes and highlights undescribed slide elements, providing visually impaired learners with a structured and accessible way to engage with presentation videos.

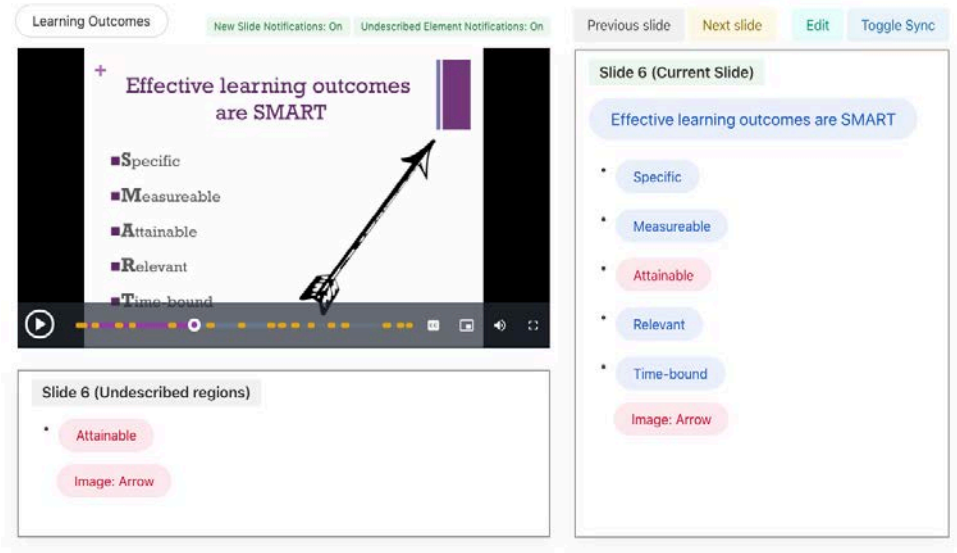


Figure 2.10 – Slidecho (Peng et al. 2021b) highlights visual elements that are not described (undescribed regions)

Additionally, the *Say It All* (Peng et al. 2021b) system provides both real-time and post-presentation feedback, this time, to presenters, enabling them to improve the accessibility of their slide-based presentations. The system analyzes slide elements such as text, images, and videos through a Document Object Model (DOM) parser and applies Optical Character Recognition (OCR) for text embedded in visuals. Real-time transcription aligns spoken words with slide elements, using word-stem matching and bounding-box highlights to guide presenters in covering all slide content. Post-presentation feedback identifies gaps and offers actionable suggestions, such as simplifying visuals or adding more detailed descriptions, empowering presenters to refine their slides for greater inclusivity.

SlideGestalt (Peng et al. 2023) enhances slide deck accessibility for blind and visually impaired users by automatically extracting hierarchical structures. Using the Google Slides API, it annotates slide elements (text, images, tables) and computes visual and textual correspondences

through visual and content embeddings derived from pre-trained on document models. Slide clustering is achieved using cosine distances and an unsupervised sequence alignment model, identifying sections and subsections based on dividers, build slides, and topic splits. High-level structure is described using slide titles or auto-generated captions via the Google Vision API, while element reading order is optimized by proximity to the top-left corner

It is important to note that the three previous works uses slides presentations from Google Slides and uses their platform and metadata to extract visual entities and additional information (such as hierarchy), which is easier than detecting directly from the video, making difficult to generalize their methods.

Another notable tool is *SmartLearn* (Sechayk et al. 2024), a system designed to improve accessibility in slide-based e-learning videos, addressing issues such as pacing, navigation, and visual focus. It incorporates temporal adjustments, OCR-speech alignment, frame-activity detection, and semantic segmentation to create an intuitive learning experience. Temporal adjustments regulate video playback speed by aligning it with a desired words-per-second rate, calculated from transcribed speech. OCR-speech alignment synchronizes text extracted from slides with spoken narration by matching common words in sentences, enabling interaction between visual and audio content. Frame-activity detection identifies key slide interactions, like pointer movements and annotations, emphasizing important visual elements. Semantic segmentation uses slide transitions and language models to divide videos into chapters, generating sections with descriptive titles.



Figure 2.11 – SmartLearn (Sechayk et al. 2024) interface. They propose adjustement to playback speed, OCR-speech cross highlighting and a temporal segmentation with titles.

The systems discussed here illustrate how advances in image and video analysis have tackled challenges in summarization, accessibility, interactivity, and usability. These advances rely on a range of techniques, including multimodal analysis, OCR, semantic analysis, entity detection, and automatic speech recognition (ASR). Deep learning is central to many of the techniques discussed, enabling accurate and efficient solutions across a range of applications. A recurring challenge, however, is the reliance on detecting visual elements in slide-based videos—a process that often struggles with inconsistencies in slide design or quality. This limitation stems from a larger issue: the lack of diverse datasets tailored specifically to slides. To address this, our first contribution detailled in Chapter 3 focuses on generating synthetic slides with rich annotations to tackle this challenge.

Before presenting our work, it is important to look at how deep learning has evolved to tackle challenges in vision, audio, and text. The next section highlights key milestones that have shaped the techniques discussed so far.

2.2 Modalities in Deep Learning

Lecture videos are inherently multimodal, combining information from various sources such as speech, visuals (images and handwritten blackboards) and text (transcripts and slide content). Each modality offers unique insights, and deep learning has become instrumental in extracting and integrating these features effectively. This section provides an overview of deep learning strategies in the domains of computer vision, speech/audio processing, and natural language understanding.

2.2.1 Vision

In the context of lecture video analysis, computer vision tasks such as image classification, object detection, segmentation, and tracking serve as building blocks for transforming raw lecture recordings into dynamic, accessible, and interactive learning resources. These tasks answer fundamental questions: *What is this content? Where is it located? How does it change over time?*

For instance, image classification allows systems to identify and categorize content like slides, diagrams, or equations, helping students quickly locate and review key concepts during revision. However, knowing "what" something is often isn't enough. Object detection extends this capability by not only recognizing visual elements but also pinpointing their position within a video frame—whether it's a diagram, a graph, a title or the lecturer's position. Segmentation further refines this understanding by delineating the boundaries of these elements at the pixel level, allowing systems to extract precise slide content or isolate specific structural details. Finally, visual tracking adds a temporal dimension by following the instructor's gestures or

monitoring movement across video frames, helping students focus on points of emphasis in the lecture.

Achieving these tasks relies on significant progress in deep learning, particularly for image processing. Prior to what we could call the "deep learning era", visual analysis depended on manually crafted features—handpicked rules to describe edges, shapes, or textures. However, these techniques were limited in their ability to generalize to complex data like lecture videos. The introduction of convolutional neural networks (CNNs), as formalized in LeNet-5 (LeCun et al. 1998) marked a turning point in visual analysis. CNNs automate feature extraction through a layered architecture. Early layers learn to detect simple patterns such as edges, while deeper layers learn to capture more intricate structures, such as in our applications, diagrams, equations, or even the lecturer's gestures.

With the increase in computer power over the years, a major breakthrough came with AlexNet (Krizhevsky et al. 2012), a 8-layer CNN that demonstrated the power of training large networks on massive datasets. AlexNet employed the Rectified Linear Unit (ReLU) activation function to introduce non-linearity, enabling faster convergence during training compared to traditional activation functions like the sigmoid or tanh activation functions. By leveraging GPU acceleration, AlexNet achieved unprecedented accuracy on the ImageNet (Deng et al. 2009) challenge—a benchmark for image classification. This innovation means systems can classify more complex visual content and with better accuracy.

Building on the success of AlexNet, subsequent architectures such as VGG (Simonyan and Zisserman 2015) increased network depth by stacking multiple small (3×3) convolutional filters. This deeper design enabled the model to learn more expressive feature representations, albeit at a significant computational cost. Later, ResNet (He et al. 2016) addressed the challenges associated with very deep networks, particularly the vanishing gradient problem—a phenomenon where gradients become too small to effectively update weights in early layers during backpropagation meaning the model would not learn. Residual connections enable the output of a layer to be added directly to the output of a deeper layer, effectively allowing the network to learn residual functions instead of direct mappings. This architecture creates shortcut paths through which gradients can flow more easily during backpropagation, mitigating the vanishing gradient problem. As a result, residual connections make it feasible to train significantly deeper networks—sometimes consisting of over a hundred layers—without the degradation in performance that traditionally accompanied increased depth.

While classification identifies what visual elements are present, object detection answers where they are located. Faster R-CNN (Ren et al. 2015) introduced a two-stage approach: first, it generates regions of interest using a Region Proposal Network (RPN), and then it classifies and refines these regions. This method is particularly effective for detecting smaller objects like sub-

diagrams or icons in slides but comes at a computational cost. To address this, the YOLO (You Only Look Once) (Redmon et al. 2016, Redmon and Farhadi 2018, Jocher et al. 2023) family introduced a single-stage architecture that predicts bounding boxes and class probabilities in a single pass through the network.

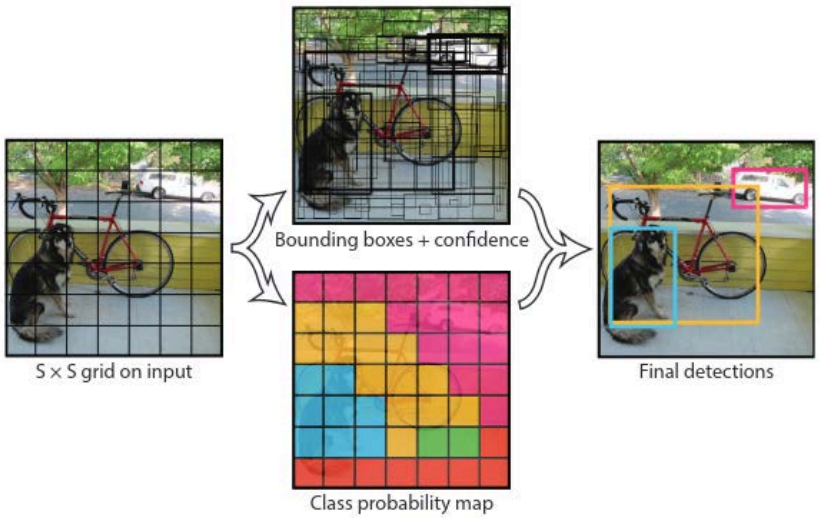


Figure 2.12 – YOLO models detection as a regression problem. It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. (Redmon et al. 2016)

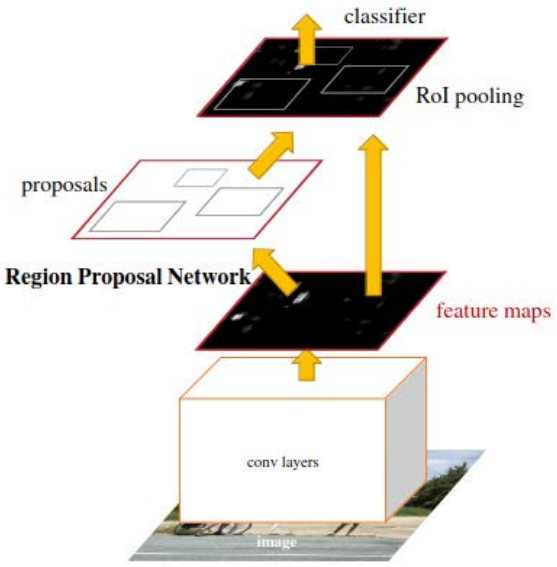


Figure 2.13 – Faster R-CNN is a two-stage network for object detection. First, it generates regions of interest and then classifies and refines these regions. (Ren et al. 2015)

For tasks requiring even greater precision, such as isolating slide regions or fine content boundaries, image segmentation techniques come into play. Unlike detection, which localizes objects with bounding boxes, segmentation delineates exact object shapes at the pixel level. A notable

advancement in this area is Mask R-CNN (He et al. 2017), which extends traditional detection frameworks by incorporating a mask prediction branch. This allows models to not only detect objects but also generate precise pixel-level masks, making it particularly useful for extracting slide content, diagrams, and structural details in lecture videos.

Recent developments such as EfficientNet (Tan and Le 2019) and ConvNext (Liu et al. 2022a) have further pushed the boundaries of deep learning in vision. EfficientNet optimizes model size by scaling its depth, width, and resolution in a balanced manner, making it lightweight yet powerful—ideal for large-scale video analysis. ConvNext, on the other hand, integrates design ideas from Vision Transformers (ViTs)—a new architecture inspired by natural language processing. By reinterpreting CNNs through transformer principles, ConvNext achieves competitive performance, bridging the gap between traditional convolutional models and emerging transformer-based approaches.

2.2.2 Audio

When analyzing lecture videos, audio processing plays a key role in understanding spoken content, detecting acoustic events, and generating accurate transcriptions. These tasks require deep learning models capable of capturing the dynamic and sequential nature of audio signals, which evolve continuously over time. Early models, such as recurrent neural networks (RNNs), were designed to capture temporal dependencies by maintaining a hidden state that processes inputs step by step. However, RNNs faced a significant limitation: the vanishing gradient problem, where gradients diminish as they propagate through long sequences. This challenge makes it difficult for these models to learn relationships spanning extended time steps, such as connecting concepts introduced at the start of a lecture to those discussed near the end.

To address this limitation, long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997) introduced memory cells and gating mechanisms—input, forget, and output gates—that regulate the flow of information. These mechanisms enable LSTMs to retain relevant information across long sequences and discard irrelevant details, making them highly effective for tasks like speech transcription. A streamlined alternative, the gated recurrent unit (GRU) (Chung et al. 2014), simplifies the architecture by combining the input and forget gates into a single update gate, reducing computational cost while maintaining similar performance. GRUs are particularly well-suited for efficiently processing lengthy audio recordings, such as lecture transcriptions.

Most modern audio processing models, including RNNs, LSTMs, and GRUs, rely on mel spectrograms as input features to represent audio signals. A spectrogram maps sound by displaying time on one axis and frequency on the other, showing how different frequency components of the audio evolve over time. Specifically, mel spectrograms use the mel scale,

which aligns with human auditory perception by giving finer resolution to lower frequencies and coarser resolution to higher ones.

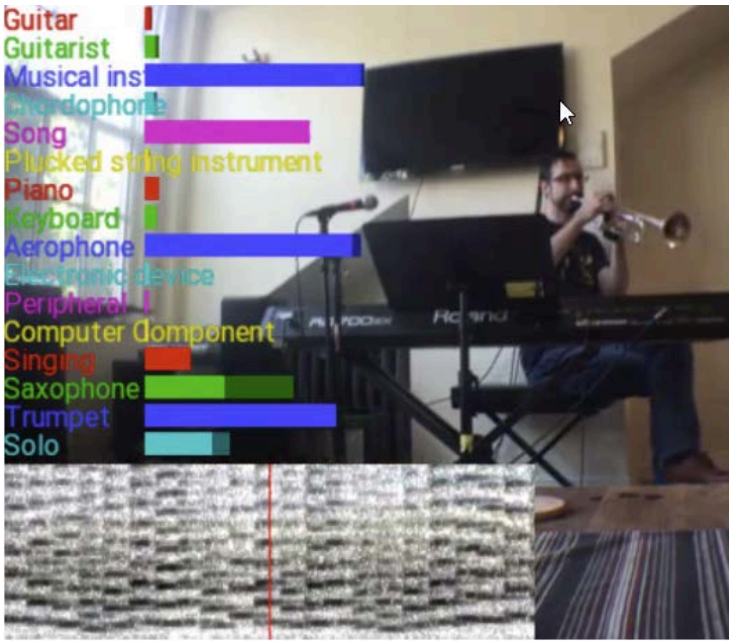


Figure 2.14 – A CNN is used on the spectrogram (bottom-left) to classify from what instrument the sound is from (Hershey et al. 2017)

In addition to being used as feature inputs for sequential models, spectrograms can also be treated as images and processed directly by convolutional neural networks (Piczak 2015, Hershey et al. 2017). In this approach, spectrograms are fed into CNNs, which classify the audio based on visual patterns. This method is particularly effective for tasks like speaker identification, where unique frequency patterns associated with an individual's voice are visible in the spectrogram, or acoustic event detection, where the model learns to recognize distinct sound events like background noise, music, or applause.

A significant leap in audio processing came with the introduction of attention mechanisms in sequence-to-sequence models. Traditional approaches struggled to align input spectrogram features with their corresponding transcriptions, as spoken words often vary in pace and duration. The *Listen, Attend, and Spell* (LAS) model (Chan et al. 2016) overcame this by dynamically focusing on the most relevant parts of the spectrogram during decoding, improving both alignment and transcription accuracy.

2.2.3 Text

Text processing plays a central role in extracting information from transcripts, summarizing content, and analyzing textual elements like on-screen slides or captions. Like audio, text is inherently sequential, where the meaning of each word depends on its surrounding context. Capturing these relationships efficiently posed a significant challenge for early machine learning

models, which relied on handcrafted features or statistical methods that struggled with complex language patterns.

The advent of recurrent neural networks, and their improved variants such as LSTM and GRUs provided a breakthrough for handling sequential dependencies. These models excelled at capturing relationships between words across sentences, enabling early successes in tasks like language modeling, text generation, and machine translation (Cho et al. 2014), where long-term dependencies are critical. For instance, RNNs laid the foundation for sequence-to-sequence (seq2seq) methods, a paradigm introduced in Sequence to Sequence Learning with Neural Networks (Sutskever et al. 2014). Seq2seq models process input and output sequences of differing lengths, such as translating an English sentence into French, by encoding the input into a latent representation and then decoding it step-by-step into the target language. While powerful, these methods struggled to capture alignments between input and output sequences, leading to challenges in translation accuracy for long or complex sentences.

This limitation was addressed by the introduction of attention mechanisms, like for the audio modality, as proposed in *Neural Machine Translation by Jointly Learning to Align and Translate* (Bahdanau et al. 2015). Attention allowed models to focus on specific parts of the input sequence when generating each word in the output, improving translation quality. For example, when translating a lecture transcript, the model could attend to the relevant parts of the source sentence to generate coherent translations of technical phrases or terminologies.

While sequential models tackled dependencies effectively, they required a step-by-step processing approach, making them computationally expensive and difficult to parallelize for long text sequences. To overcome these limitations, word embeddings emerged as a complementary innovation for representing textual data in deep learning. Prior to embeddings, words were treated as discrete, unrelated symbols, limiting a model's ability to understand meaning. Models like Word2vec (Mikolov et al. 2013) changed this by learning dense vector representations that capture semantic relationships between words.

Word2vec introduced two key approaches: the skip-gram model, which predicts nearby words given a target word, and the continuous bag-of-words (CBOW) model, which predicts a target word based on its surrounding context. For example, skip-gram might take the word "lecture" and learn to associate it with frequent neighbors like "video" or "slides," while CBOW might infer "neural" from the surrounding words "deep" and "learning." These embeddings enabled models to understand analogies such as "king - man + woman ≈ queen" by capturing relationships in the vector space, allowing deeper semantic reasoning. Techniques like GloVe extended this idea by incorporating global co-occurrence statistics, ensuring word relationships were preserved across entire corpora.

Despite these advances, RNN-based models still struggled to handle the demands of large-scale text processing efficiently. Attention mechanisms offered an initial solution to this bottleneck, but the true paradigm shift came with the introduction of the Transformer architecture (Vaswani 2017). By removing sequential dependencies altogether and relying solely on self-attention, Transformers enabled models to process entire sequences in parallel, capturing both local and global dependencies with efficiency.

2.3 Transformers: A Unifying Architecture

The introduction of transformers mredefined the landscape of deep learning for sequence modeling by addressing the inefficiencies of prior architectures. While RNNs and LSTMs laid the groundwork for processing text and audio, their reliance on step-by-step processing made them inefficient for long sequences and large-scale data. In the paper *Attention is All You Need*, (Vaswani 2017) addressed these challenges by replacing sequential computations with the self-attention mechanism, an approach that computes relationships across all elements in an input sequence simultaneously.

Unlike RNNs, where context is built progressively, transformers consider the entire sequence at once. For example, in the sentence "The cat sat on the mat," the word "mat" can directly "attend" to both "sat" and "cat" without needing to traverse intermediate tokens. This ability to dynamically assign relevance enables transformers to capture both global context and long-term dependencies efficiently, while their parallelizable design eliminates the bottlenecks of traditional models. These advantages unlocked the ability to scale to larger datasets, longer inputs, and more complex tasks.

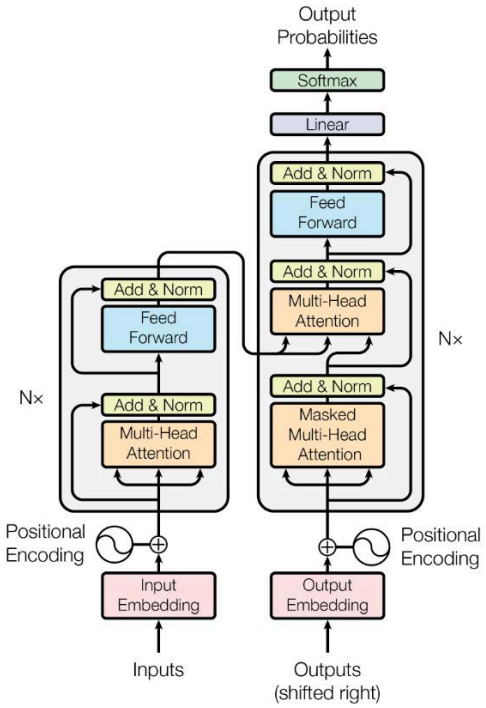


Figure 2.15 – The architecture of a Transformer

At the architectural level, transformers are built on encoder-decoder stacks that are both modular and scalable. The encoder processes input sequences through layers of self-attention and feed-forward networks, creating rich contextual representations of the data. In tasks requiring generation, the decoder integrates these representations, attending to relevant parts of the encoder outputs while generating new tokens. To compensate for the absence of inherent sequence order, transformers use positional encodings, which inject information about token positions, allowing them to model sequences effectively despite their parallel structure.

Among the most impactful applications of transformers are the development of BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2019) and GPT (Generative Pre-trained Transformer) (Brown et al. 2020a), two models that showcase the architecture's adaptability to Neural Language Processing tasks. BERT is designed primarily for language understanding tasks. Unlike traditional left-to-right or right-to-left models, BERT uses bidirectional attention, allowing it to consider both preceding and succeeding tokens simultaneously. This approach enables BERT to capture nuanced relationships between words, making it effective for tasks like classification, named entity recognition, and semantic search. BERT can also generate sentence representations (embeddings) by aggregating token-level embeddings, which are useful for comparing the similarity of sentences. For instance, given the sentences "The teacher explained the equation in detail" and "The instructor provided a detailed explanation of the formula", BERT can encode each sentence into a vector representation that captures its meaning, allowing the model to measure how semantically similar the sentences are, even if they use different wording.

In contrast, GPT is designed for text generation and excels in tasks requiring fluency and creativity. GPT employs unidirectional attention, where each token attends only to previous tokens, generating text one word at a time. This autoregressive approach ensures that outputs are coherent and contextually consistent. For example, GPT can take a prompt like "Write a summary of this lecture video transcript," and generate well-structured paragraphs that align with the given context. While this makes GPT ideal for tasks like summarization, storytelling, and text completion, it is less suited for understanding-focused tasks, where bidirectional context is essential.

Transformers did not stop at text. In audio processing, both the Conformer (Peng et al. 2021c) and Whisper (Radford et al. 2023a) models leverage spectrograms but adopt fundamentally different architectures to handle ASR tasks. The Conformer employs a hybrid approach, combining self-attention for capturing long-range dependencies with convolutional layers that model local features like phonemes and transitions between sounds. This design balances the strengths of transformers and CNNs.

In contrast, Whisper relies entirely on a transformer-based architecture. It processes spectrograms directly through multiple layers of self-attention, capturing both local and global dependencies within a unified framework. Additionally, Whisper's training leverages a massive, diverse dataset encompassing multilingual and noisy audio, enabling it to generalize robustly across a wide range of transcription and translation tasks.

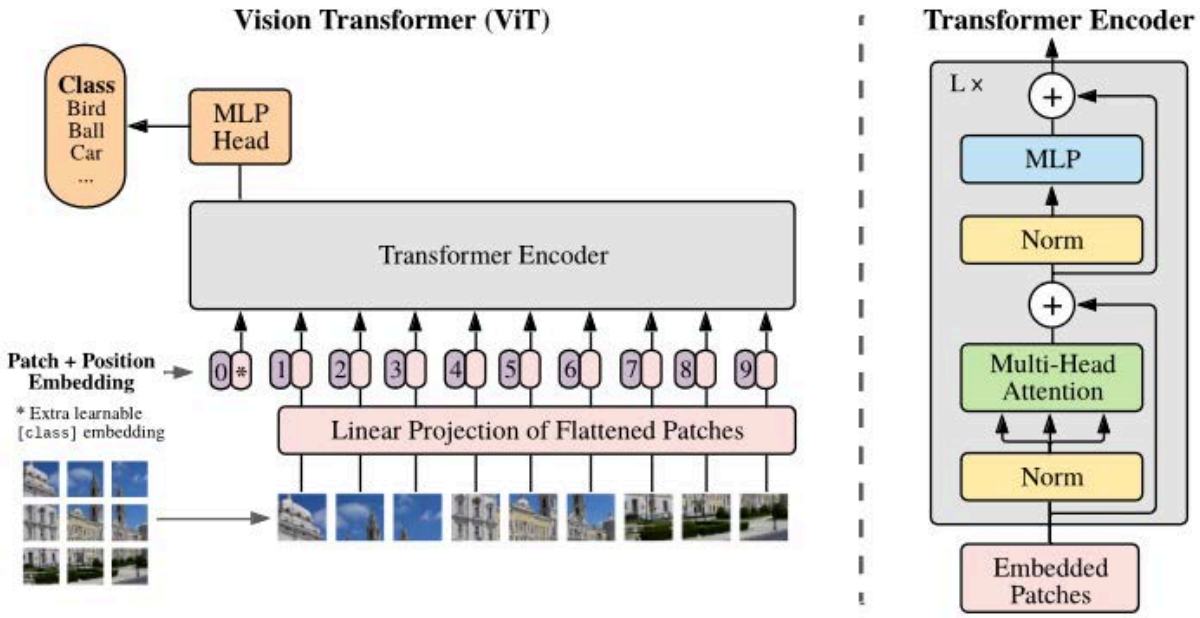


Figure 2.16 – The architecture of a Vision Transformer: Images are divided into patches to form a sequence. Positional embeddings are added to the patch sequence, akin to text processing, to preserve spatial relationships.

In the visual domain, transformers redefined image processing through the introduction of Vision Transformers (ViTs) (Dosovitskiy et al. 2021). Unlike convolutional neural networks (CNNs), which focus on local patterns, ViTs treat images as sequences of small patches and apply self-attention to model global relationships shown in Figure 2.16. With this progress, models like DINO (Caron et al. 2021) and CLIP (Radford et al. 2021) have emerged as foundational models with the ability to learn rich, generalizable representations.

DINO (Caron et al. 2021) uses self-supervised learning to train vision transformers without requiring labeled data. It employs self-distillation, a process where the model learns to produce consistent feature representations across different augmented views of the same image. This approach enables DINO to capture both global and local structures. Combined with the scalability of transformers and access to massive datasets, DINO achieves state-of-the-art performance in tasks such as unsupervised object segmentation, classification and semantic understanding.

In contrast, CLIP (Radford et al. 2021) aims to bridge the gap between vision and langauge by aligning image and text embeddings through cross-modal learning. It achieves this by training a CNN-based vision encoder alongside a transformer-based text encoder on large-scale image-text pairs. This alignment enables CLIP to perform zero-shot image classification, where visual content can be recognized using textual prompts. For example, given a slide containing a chart, CLIP can associate the image with concepts like “graph” or “data visualization” without additional training.

The strength of transformers lies in their ability to handle massive amounts of data efficiently, a capability that opened the door to the creation of large language models.

Architecture	Core Mechanism	Primary Strengths	Typical Modalities	Example Applications
CNN	Convolution, Pooling	Spatial hierarchies, Local patterns, Translation invariance	Vision, Audio (Spectrograms)	Image Classification, Object Detection
RNN/LSTM/GRU	Recurrence, Gating	Temporal sequences, Sequential dependencies	Audio, Text, Time Series	Speech Recognition, Machine Translation
Transformer	Self-Attention	Long-range dependencies, Parallel processing	Text, Vision, Audio	NLP Tasks (LLMs), ViT, Audio Generation

Table 2.1 – Summary of the different neural architectures. Transformer architecture can efficiently treat all modalities.

2.4 Multimodal Large Language Models

Built on the foundation of the transformer architecture, large language models (LLMs) have revolutionized natural language processing, achieving groundbreaking results across a wide range of tasks. By leveraging vast datasets comprising trillions of tokens and billions of para-

meters, these models have set a new benchmark for what machines can achieve in understanding and generating human language. For instance, GPT-3 (Brown et al. 2020b) demonstrated its capacity to perform complex tasks like summarization, translation, and code synthesis—tasks that were previously infeasible without significant task-specific fine-tuning—establishing a clear leap over earlier language models. From generating coherent responses and summarizing documents to answering complex queries and synthesizing code, LLMs have redefined the scope of AI-driven text-based applications.

What makes LLMs truly groundbreaking is their ability to generalize across tasks with minimal or no task-specific training. Extensive pretraining on diverse, unstructured datasets—including books, web pages, technical papers, and conversational data—allows these models to internalize intricate patterns of grammar, semantics, and reasoning. This capability has enabled emergent behaviors such as zero-shot and few-shot learning, where models like GPT-3 (Brown et al. 2020a) adapt to new tasks simply by interpreting prompts or seeing a few examples at inference time. Furthermore, techniques like reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) refine their outputs to align with human preferences, enhancing their usability and precision for specific applications.

The rise of conversational assistants exemplifies the transformative impact of LLMs. First appearing in 2022, these systems democratized the use of AI and LLMs, making them accessible to a wide audience through intuitive interfaces and practical applications.

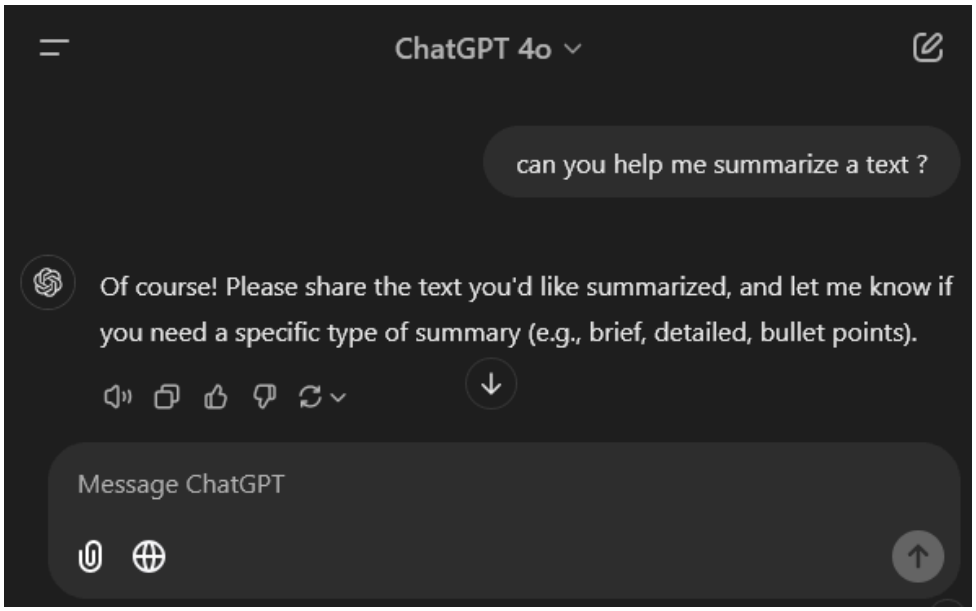


Figure 2.17 – Interface of ChatGPT 4, a conversational assistant powered by the Large Language Model GPT-4

By fine-tuning general-purpose models on question-and-answer pairs and conversational datasets, these systems are tailored to focus on interactive dialogue. Unlike general-purpose LLMs, conversational assistants like ChatGPT, Gemini, LLaMA (Touvron et al. 2023, OpenAI

et al. 2024, DeepMind et al. 2024) are fine-tuned to handle multi-turn interactions, maintain context, and align closely with user intent. By enabling interactive, back-and-forth exchanges, these systems allow users to receive highly tailored responses and even gain new perspectives through dynamic dialogue. Additionally, these LLMs can be augmented with tools that enhance their capabilities, such as retrieving information from text documents, conducting web searches, or executing code snippets.

The evolution of LLMs has recently extended beyond text, with a growing focus on multimodal capabilities. These next-generation models integrate text with other data types—images, audio, and even video—enabling them to "see, hear, and describe the world" in ways that approximate human-like understanding (Bommasani et al. 2022).

Multimodal LLMs adapt the principles of transformers to process non-text modalities by mapping them into a shared latent space. This unified representation enables cross-modal reasoning, where relationships between visual, auditory, and textual data are understood holistically like in CLIP (Radford et al. 2021). Training these models on large-scale multimodal datasets, such as LAION-5B (Schuhmann et al. 2024), which contains billions of image-text pairs, ensures that they learn meaningful associations across modalities, bridging the gap between text and vision.

Prominent examples include GPT-4, Gemini and LLaMA 3, which integrate vision and audio capabilities alongside text (Touvron et al. 2023, OpenAI et al. 2024, DeepMind et al. 2024). This allows users to interact with images and audio, through natural language queries, automatic speech recognition, analyzing content, generating descriptions, and answering questions based on contextual cues.

The practical applications of multimodal LLMs are diverse and transformative. These models excel in tasks like image captioning (BLIP (Li et al. 2022)), where they produce detailed and coherent textual descriptions of visual content. Visual Question Answering (VQA), once considered a challenging task when introduced in 2015 (Antol et al. 2015), has become significantly easier with multimodal LLMs. In the audio domain, multimodal LLMs not only transcribe spoken language but identify speakers, and engage in interactive conversations with users through realistic text-to-speech models (Li et al. 2023, OpenAI et al. 2024).

Since their emergence as conversational assistants in 2022, these models have consistently achieved significant milestones. In 2023, they demonstrated exceptional performance on code benchmarks (Chen et al. 2021a) and introduced multimodal capabilities, integrating text, vision, and audio seamlessly. By 2024, LLMs further expanded their functionalities, understanding voice commands with greater precision, and introducing advanced reasoning abilities (Huang and Chang 2023).

Despite their promise, LLMs are not without limitations. A recurring challenge is hallucination, where models generate outputs that are plausible but factually incorrect. In the survey on hallucination from (Ji et al. 2023), the authors say that "hallucination is of concern because they appear fluent and can therefore be misleading". For example, in a medical context, an LLM might confidently suggest an unverified treatment plan or misinterpret diagnostic data, leading to potentially harmful consequences. In educational settings, hallucinations can result in the propagation of misinformation, such as generating historical events that never occurred. This issue is particularly problematic in applications requiring high reliability, such as medical imaging or educational content. Hallucinations highlight the need for careful evaluation, fine-tuning, and constraints to ensure models produce accurate and trustworthy results.

The implications of multimodal LLMs go far beyond conventional AI applications. Now that they are able to "see" and "talk", they are enabling innovations in accessibility, such as tools that describe visual content for visually impaired users or transcribe and summarize audio for better navigation. As LLMs evolve into multimodal systems, they represent a crucial step toward more integrated and intelligent AI. However, their high computational cost and susceptibility to errors like hallucination underscore the importance of exploring lighter and more efficient alternatives for specific tasks.

In the next chapter, we will address these challenges by contributing two key solutions. First, in Chapter 3, we generate synthetic slides enriched with custom annotations—such as textual descriptions, object detection labels, and segmentation data—to overcome data scarcity in lecture video analysis. These synthetic datasets provide customized slide images for training without the need for extensive manual labeling. Second, in Chapter 4, we propose an approach leveraging lightweight embedding models across modalities to segment lecture videos into coherent topics. This method enables improved indexing and navigation, transforming lecture recordings into structured and accessible resources without relying on computationally heavy multimodal LLMs.

Chapter 3

Synthetic Slides Generation for Robust Slide Analysis

To improve the understanding of lecture videos, an important component is the visual semantic analysis of slide content, which plays a central role in applications such as summarization, content indexing, and accessibility enhancement. A major obstacle in this area is the scarcity and poor documentation of high-quality slide datasets, which hampers the development and training of robust models. To address this, we propose the design of a synthetic slide generator.

In this chapter, we present the methodology for building this tool, its capacity to generate diverse and richly annotated synthetic slide datasets, and its potential to support more effective training and evaluation of slide analysis models.

3.1 Introduction

Presentations in sectors such as education and research are increasingly being recorded and shared online, particularly since the pandemic of the early 2020s. This trend fueled a renewed interest in innovative applications for processing and interacting with recorded digital presentations, such as summarization (Xu et al. 2019, Yoo et al. 2021), indexing (Mukhopadhyay and Smith 1999, Adcock et al. 2010b, Kim et al. 2014, Zhao et al. 2019a), and enhanced browsing (Mukhopadhyay and Smith 1999, Adcock et al. 2010b, Kim et al. 2014, Zhao et al. 2019a). Slides are important visual aids for a presentation that often drive the structure and key points of the content of a presentation. Extracting the semantics from the slides, therefore, is an important fundamental task that enables these downstream applications. Much of the slides available online, however, are in image or video format, making computational analysis and understanding of the slides a non-trivial problem.

Recent efforts in slide analysis have pivoted to data-driven, learning-based, methods (Kim et al. 2022b, Tanaka et al. 2023). For such methods to be effective, there needs to be a large corpus of data with high-quality annotations. However, existing slide datasets (Haurilet et al. 2019a, 2019b; Kim et al. 2022b, Tanaka et al. 2023) often either suffer from inconsistencies and inaccuracies, or lack the essential annotations (Araujo et al. 2015) necessary for the development of effective analysis algorithms. Beyond these issues, there are additional concerns that slow down progress in this domain. Firstly, the process of accurately annotating slide datasets is labor-intensive and expensive, requiring significant expertise and time to ensure high-quality annotations. Secondly, this challenge is compounded by the scarcity of comprehensive slide datasets available in the public domain, limiting the diversity and scope of training data for analysis algorithms. Finally, there is often an imbalance in the type of elements present on the slides within these datasets. For example, some classes of visual or textual elements may be overrepresented while others are underrepresented, leading to skewed learning outcomes and limiting the algorithms' ability to generalize across different slide layouts and content types.

To address these challenges, we introduce a tool developed for generating synthetic slide datasets. This first tackles the issue of labor-intensive annotations. Generating content with annotation eliminates the costly and time-consuming nature of manual annotation, ensuring high-quality, consistent annotations across the dataset. Second, it mitigates the scarcity of comprehensive slide datasets by generating a large and diverse array of realistic slides. Finally, it addresses the imbalance in slide elements by allowing for customizable element distribution in generated slides. This ensures a balanced representation of both visual and textual elements. In summary, our tool's design not only imitates real-world presentations but also methodically overcomes the existing limitations of slide datasets, thereby facilitating the development of slide analysis algorithms.

In this chapter, we present the development, capabilities, and potential impact of our tool. Our focus is on its contribution to overcoming the current limitations in slide dataset availability and quality, thereby paving the way for future advancements in digital presentation analysis. **We show that by augmenting the existing dataset with slides generated from our tool, we can improve the mAP50 up to 13% on object detection models.**

3.2 Related Work

Studies in slide dataset creation have primarily focused on collating slides from existing sources, with an emphasis on real-world diversity. However, these datasets often face challenges in terms of size, annotation accuracy, consistency, and imbalance. WiSe (Haurilet et al. 2019b) and SPaSe (Haurilet et al. 2019a) datasets, while offering segmentation masks across 25 diverse classes, are limited by their small size, containing only 2000 images – a quantity insufficient for comprehensive analysis. FitVid (Kim et al. 2022b) introduces a more subject-diverse object detection dataset over 12 classes, yet it comprises only 5527 images. In contrast, Slideshare-1M (Araujo et al. 2015), despite being the largest slide collection, lacks annotations, because it was made for content retrieval. Meanwhile, SlideVQA (Tanaka et al. 2023) stands out as the most extensive annotated slide dataset available, with 52,480 slides across 9 classes designed for object detection, showcasing a step forward in annotated slide data availability. Despite this, it faces two notable issues: a disparity between text and visual content, and the presence of inconsistencies as shown in Figure 3.1.

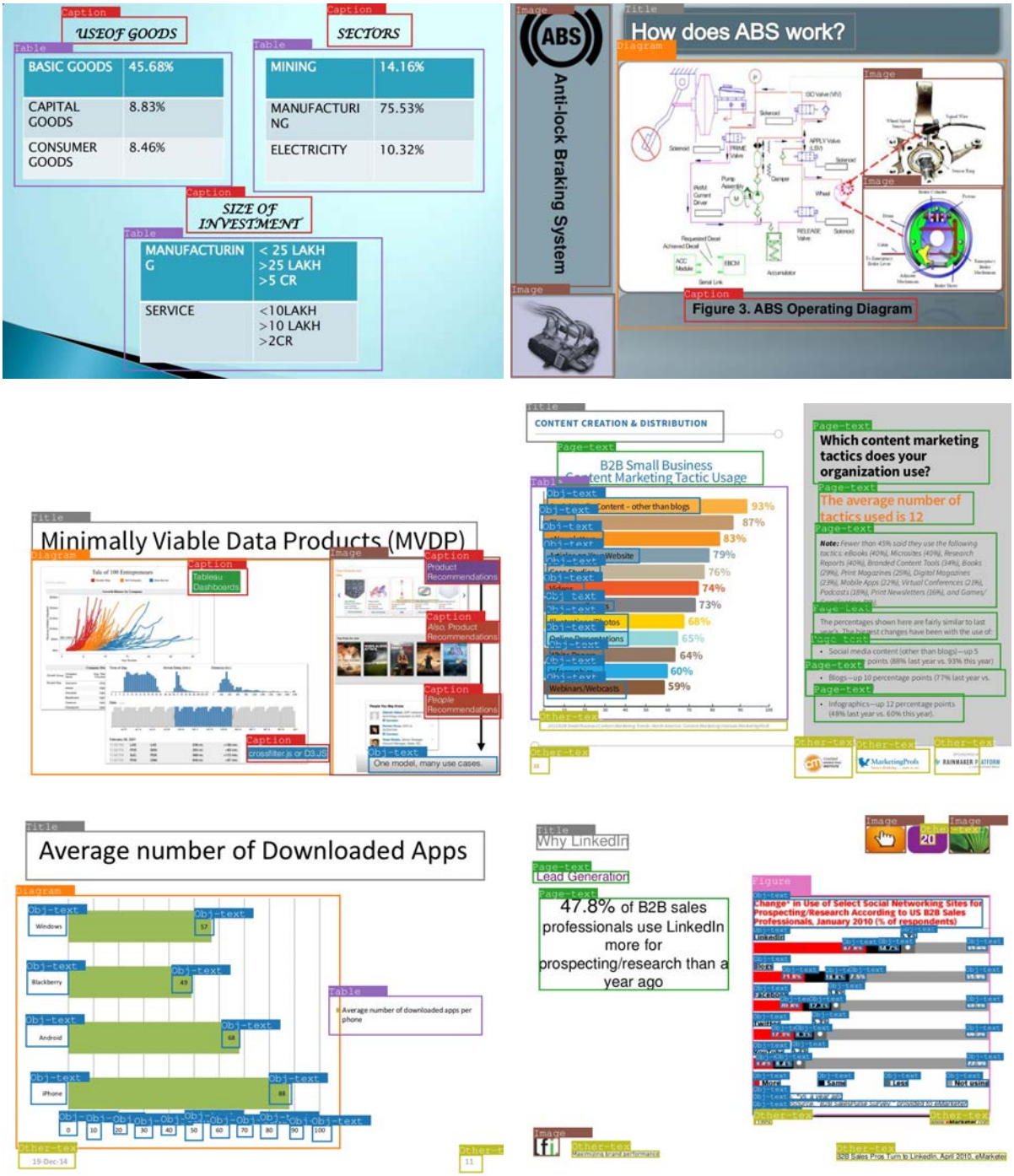


Figure 3.1 – SlideVQA contains some inconsistencies and errors. The Obj-Text class which represents all the text seen on any graphical elements (Image, Diagram, Table, Figure) is not always annotated. Moreover, there is some confusion between the class Figure and Diagram shown in the last two slides. Even though they are both horizontal charts, one is labeled as Diagram and in the other, as Figure. This affects training performance and evaluation reliability.

Document layout analysis has recently garnered increased attention in the machine learning community, a development largely fueled by the introduction of more popular benchmarks and

high-quality datasets such as DocBank (Li et al. 2020a), DocLayNet (Pfitzmann et al. 2022), and PubLayNet (Zhong et al. 2019). These datasets have been instrumental in advancing the field, offering comprehensive and diverse collections of document layouts that facilitate the training and testing of sophisticated layout analysis algorithms. However, despite these advancements, a significant gap remains in the specific domain of slide layout analysis. Unlike standard documents, slides have distinct characteristics: they are typically more visual, presented in landscape format, and contain less structured text. Furthermore, the classification needs for slide elements differ markedly from those in traditional document layouts. For instance, slides often require specific classes for elements like bullet points, titles, and visual aids, which are not commonly accounted for in general document layout datasets. The existing datasets, while robust for general document analysis, fall short of addressing the unique demands of slide presentations and potential applications.

Specialized datasets focused on document elements like tables (Li et al. 2020b), charts (Masry et al. 2022), diagrams (Kembhavi et al. 2016), plots ((Methani et al. 2020)), equations (Deng et al. 2017) and handwriting (Liwicki and Bunke 2005) have been developed to encourage better analysis and recognition. Given that slide presentations often incorporate such elements, leveraging these specific datasets within our tool presents an excellent opportunity to refine and broaden the analysis capabilities of models trained using generated slides.

Historically, synthetic datasets have been approached with caution in machine learning applications due to their tendency to diverge from real-world scenarios. This divergence often results in a performance gap when algorithms trained on synthetic data are applied to actual tasks. In contrast to this trend, some researchers have employed advanced techniques like Generative Adversarial Networks (GANs) to create more realistic synthetic datasets (Capobianco and Marinai 2017, Blanc-Beyne et al. 2021, Ferreira et al. 2021). These methods involve training neural networks to generate data that closely mimic real-world scenarios, a technique that has shown promise in fields such as image and document generation but which cannot be easily used to create reliable annotations. Others have utilized photorealistic open-world computer games to generate large datasets, offering reliable pixel-level semantic annotations for enhanced accuracy in various applications (Richter et al. 2016).

Furthermore, another approach in the generation of synthetic datasets for document layout analysis has involved creating layouts first by using a generative model and then populating them with content (Pisaneschi et al. 2023). This method targets article layouts, where the focus is on arranging text and graphical elements in a manner consistent with traditional document formats, thereby creating synthetic datasets that resemble the structure of real-world documents. However, these techniques are predominantly tailored to the needs of standard document layouts, such as articles and reports, and do not directly address the unique requirements of

slide presentations, such as having more visual content such as charts, graphs, and images, as well as the necessity to accommodate diverse layouts. The semantic organization within slides, how information is structured and prioritized, differs significantly from traditional documents, necessitating tools, and datasets specifically designed to understand and generate these visually oriented formats.

3.2.1 Existing datasets inconsistencies

In slide analysis, two of the most recent datasets, FitVid and SlideVQA, show notable inconsistencies. These issues highlight a broader problem in the field: the lack of standardized annotation practices. First, despite being a more recent effort, the SlideVQA dataset deviates from previous datasets in its labeling conventions. This lack of consensus in annotation practices complicates the task of uniformly evaluating slide content. As illustrated in Figure 3.2, the SlideVQA dataset contains several inconsistencies:

- In images (a) and (b), both slides originate from the same presentation but are annotated differently. Logos at the bottom are labeled as "Image" in one instance and "Other-Text" in another. The figure is correctly labeled in the left image but misclassified as "Table" in the right. Additionally, some "Obj-Text" elements are omitted.
- Images (c) and (d) display further inconsistencies. "Obj-Text" elements are incorrectly labeled as "Other-Text" in both images. In image (d), a table is labeled as an "Image," and no "Obj-Text" elements are identified, contrary to the dataset authors' definitions.

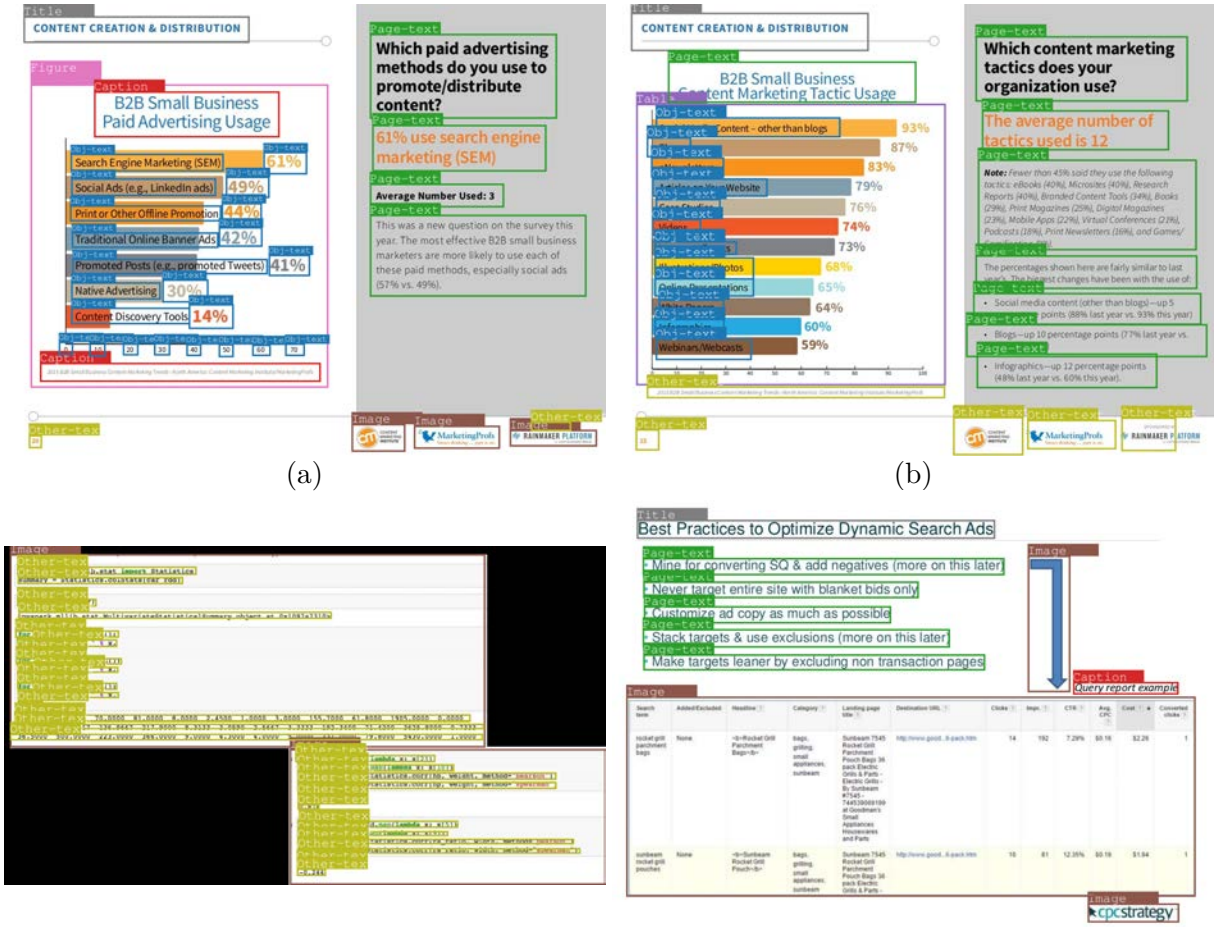


Figure 3.2 – More inconsistencies in SlideVQA.

Similarly, as shown in Figure 3.3, the FitVid dataset also suffers from inconsistencies and class confusion:

- The "Figure" class, meant to represent mathematical expressions, is sometimes absent. In image (c), mathematical expressions are labeled as "text box," which is misleading.
- In image (a), bullet points are collectively labeled as one "figure," despite containing minimal mathematical expressions.
- In image (d), a table is labeled as an "Image," and in image (e), text boxes are missing.
- Image (h) shows an entire text block labeled as "Diagram," conflicting with image (b), where a similar diagram is labeled as "Image."
- Additionally, bullet points are inconsistently annotated—sometimes grouped into a single box (image (a)) and sometimes annotated separately (image (d)).

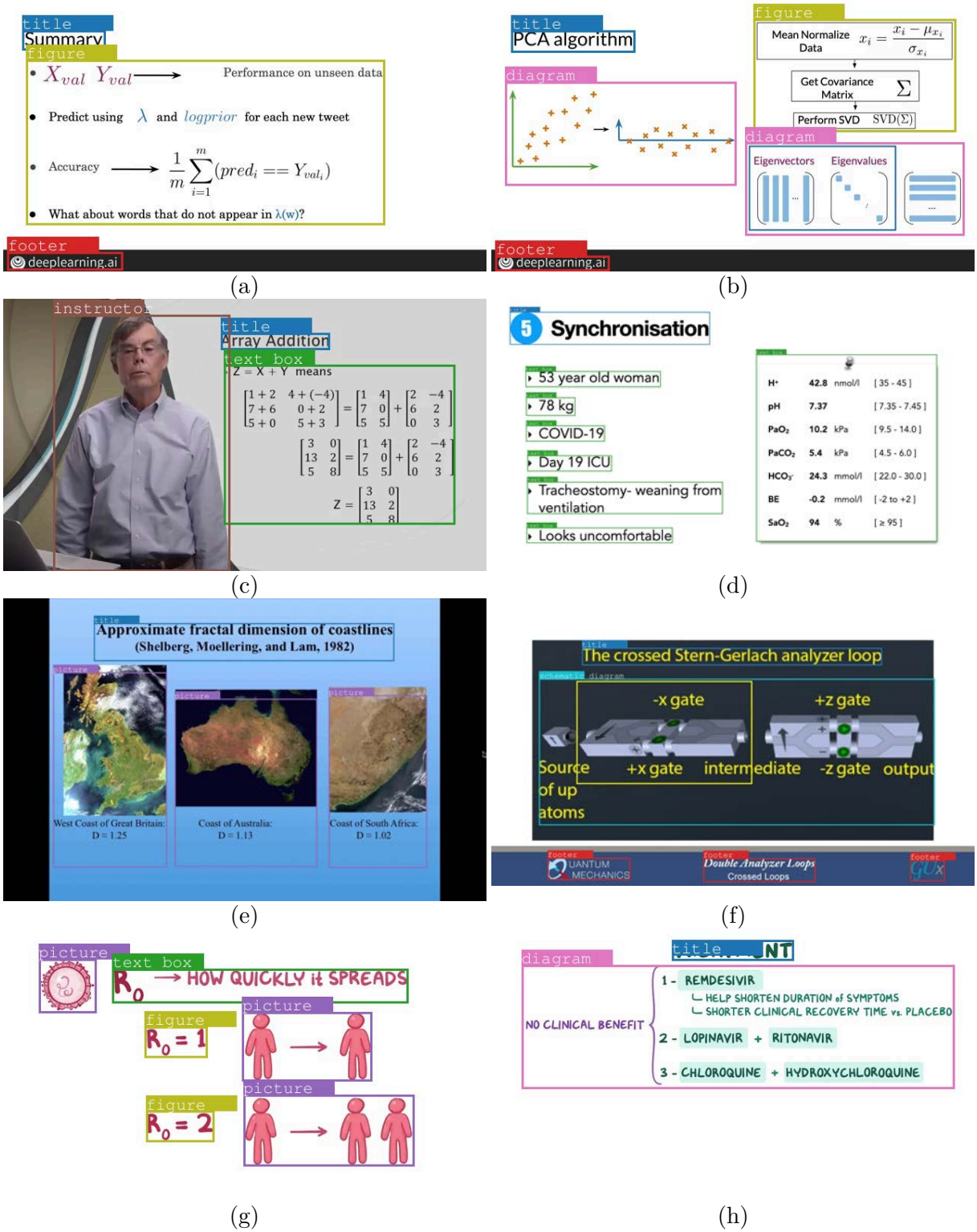


Figure 3.3 – Inconstencies and confusions for classes in the FitVid Dataset.

These inconsistencies undermine the reliability of training and evaluation, especially when working with smaller datasets. Moreover, the lack of a standardized classification scheme limits rigorous evaluation and cross-dataset comparability, presenting a significant obstacle to progress in slide analysis research.

To address these challenges and mitigate the limitations of existing datasets, we have developed a pipeline for generating synthetic slides. We provide a detailed analysis of this pipeline of generating synthetic slides in the next section.

3.3 Pipeline

We design a tool for generating extensive, labeled datasets for training learning-based slide layout analysis methods. Its primary function is to facilitate the production of a wide array of annotated slide data. This tool operates through a set of flexible rules that ensure the generation of coherent, well-structured, and readable presentations. These rules can be dynamically adjusted by users, allowing for the creation of customized layouts and styles to suit specific requirements.

The tool can be broken into four components, each responsible for generating a different element: content, layout, style, and annotation. Figure 3.4 shows the whole pipeline.

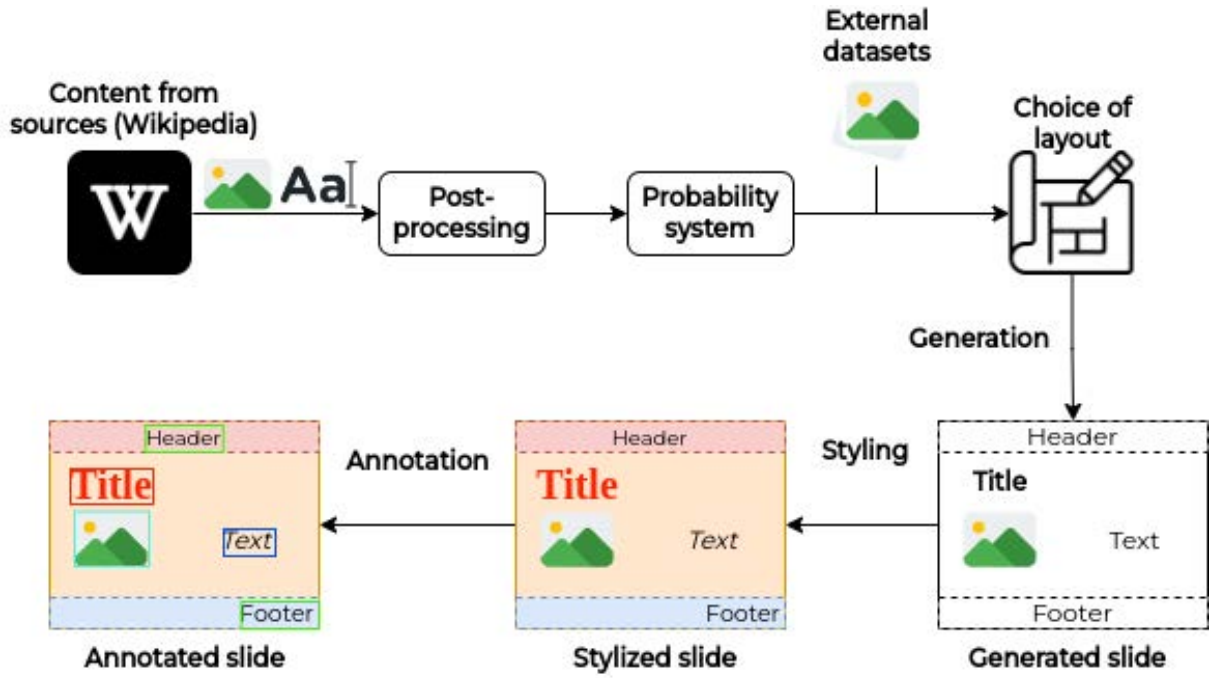


Figure 3.4 – Pipeline of our system to generate synthetic slide

Content

The content component of the pipeline is integral to generating the textual and visual elements of the slides, with primary reliance on Wikipedia articles for source material in our current implementation (the source material could come from anywhere). Each section and subsection of an article is adeptly converted into an individual slide, aligning with the original content's structure. To adapt the typically lengthy and detailed text from Wikipedia articles to a slide-friendly format, we employed Mistral-7B (Jiang et al. 2023), a Large Language Model, for

summarization. This LLM was tasked with distilling the textual elements into bullet points and short sentences, which are more suitable for the concise and direct nature of slide presentations.

However, given that Wikipedia articles are predominantly text-based and slides necessitate a strong visual component, we introduce the possibility of supplementing them with various graphical elements from other public datasets. This addition includes charts (ChartQA (Masry et al. 2022)), plots (PlotQA (Methani et al. 2020)), diagrams (AI2D (Kembhavi et al. 2016)), tables (TableBank (Li et al. 2020b)), equations (im2latex-100k (Deng et al. 2017)), handwriting (IAM-OnDB (Liwicki and Bunke 2005)), logos (Logodet-3k (Wang et al. 2022)) and more. While incorporating these diverse visual elements could introduce a degree of incoherence with the original text, it is a deliberate choice aimed at enhancing the overall visual diversity of the slides. This step is beneficial for the primary objective of the tool - to improve the performance of slide layout analysis algorithms. We show in our experiments that the use of this additional material helps to improve object detection performance in several classes.

By expanding the range of visual content in the slides, we provide a richer and more challenging dataset for training these algorithms, ultimately contributing to their enhanced accuracy and effectiveness in real-world applications.

Layout

The layout component is crucial in determining the arrangement and presentation of content within each slide. This component is designed to emulate traditional layouts commonly found in popular presentation tools like Google Slides and PowerPoint. It assesses the quantity and type of content elements - be it text, images, or graphical data - and selects the most suitable layout for each slide. The decision-making process is guided by principles of design and readability, ensuring that the final output is not only visually appealing but also easy to comprehend. To actualize these layouts, we utilize Marp (2024), a versatile tool that enables the creation of slides using Markdown, HTML, and CSS. Marp's flexibility and simplicity allow for efficient translation of the chosen layouts into polished slides. Based on the content, we choose a template tailored to the content, and we generate a markdown file that contains instructions for Marp to translate into a slide in HTML format. Templates are written in HTML and CSS and contain code to create layouts with columns, headers, footers, and different positions for elements. This approach ensures that we can produce slides that are aesthetically consistent with standard presentation formats, thereby making the generated datasets ideal for enhancing slide layout analysis algorithms. Figure 3.5 shows some examples of layouts we can generate.

A probability system also plays a part in determining the composition of each slide. This system allows for control over the occurrence probabilities of various elements within the slides. Users can adjust the generation process by specifying the weights for different classes and determining their inclusion in a slide in a greedy way. The system plays a role both in the

content phase, where it determines the likelihood of incorporating external content, and in the layout phase, deciding whether to include specific classes like titles, headers, and footers. This feature is particularly beneficial when aiming to balance an existing dataset or when focusing on enhancing specific classes that may be underrepresented or inadequately portrayed in the dataset. By adjusting these probabilities, we can tailor the layout of the slides to address specific needs or deficiencies in the dataset. For example, if the initial analysis indicates that slides with diagrams are less frequent, the system can be configured to generate more slides with diagrams. This level of customization in the layout component not only adds versatility to the tool, but also ensures that the resulting dataset is well-rounded and accurately reflective of diverse presentation scenarios. Such targeted adjustments are instrumental in fine-tuning the dataset for more effective training and evaluation of slide layout analysis models.

Style

The style component is responsible for the aesthetic diversity of the slides. This component operates by randomly selecting various stylistic parameters to ensure that each slide is distinct from the others, but can also be modified to target certain styles of slides. These parameters include a wide range of elements such as colors for different text sections (like paragraphs, titles, headers, and footers), font types, text sizes, the degree of text boldness, and more. Additionally, it also encompasses choices for background styles and colors. This is done through the modification of the CSS in the markdown files read by Marp.

Randomization in selecting these stylistic features is key to producing a dataset with high variability. This diversity is not merely cosmetic; it is needed for the robustness of the slide layout analysis algorithms. By exposing these algorithms to diverse styles, we challenge and enhance their ability to accurately analyze and interpret slides across a broad spectrum of designs. In doing so, the style component contributes to the generation of a comprehensive and varied dataset. Figure 3.6 shows some outputs of our system with different layouts and styles, and Figure 3.7 shows the same but for the same source material.

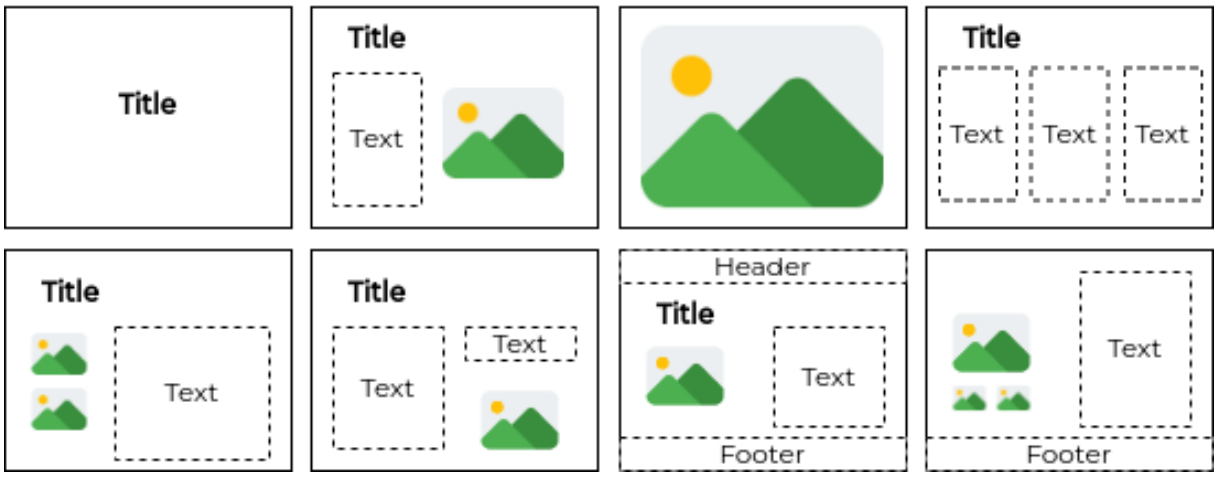


Figure 3.5 – Examples of different layouts: different number of columns, display of header and footer, display of the title. The number of elements taken from the content can be adjusted to choose between more visual or more textual output.

While our system represents the first attempt at creating a synthetic slide dataset for slide layout analysis, it is important to acknowledge its limitations in replicating the full spectrum of slide styles. The tool is not designed to mimic every possible type of slide, as the variety and complexity of presentation styles are vast and often context-specific. However, our goal is to cover the most common and widely used slide styles. By focusing on these prevalent formats, we aim to provide a comprehensive and representative dataset that reflects the majority of real-world scenarios. This strategic approach allows us to optimize the tool's effectiveness in training and improving slide layout analysis algorithms, while realistically addressing the practical constraints of such a generative system.

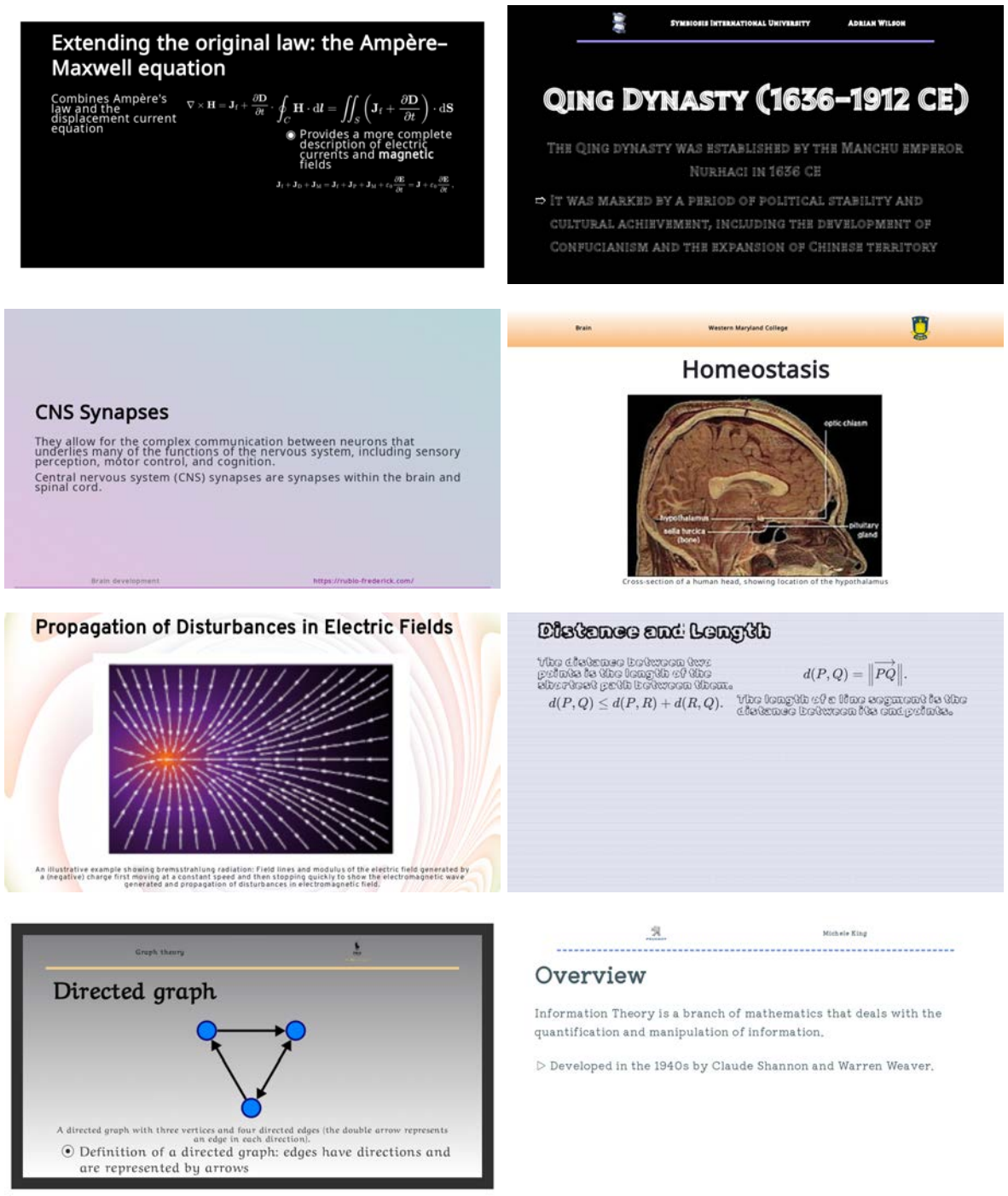


Figure 3.6 – Examples of generated slides. Different styles and layouts are shown. Colors, backgrounds, font size, and font style are chosen randomly to increase the diversity of the generation.

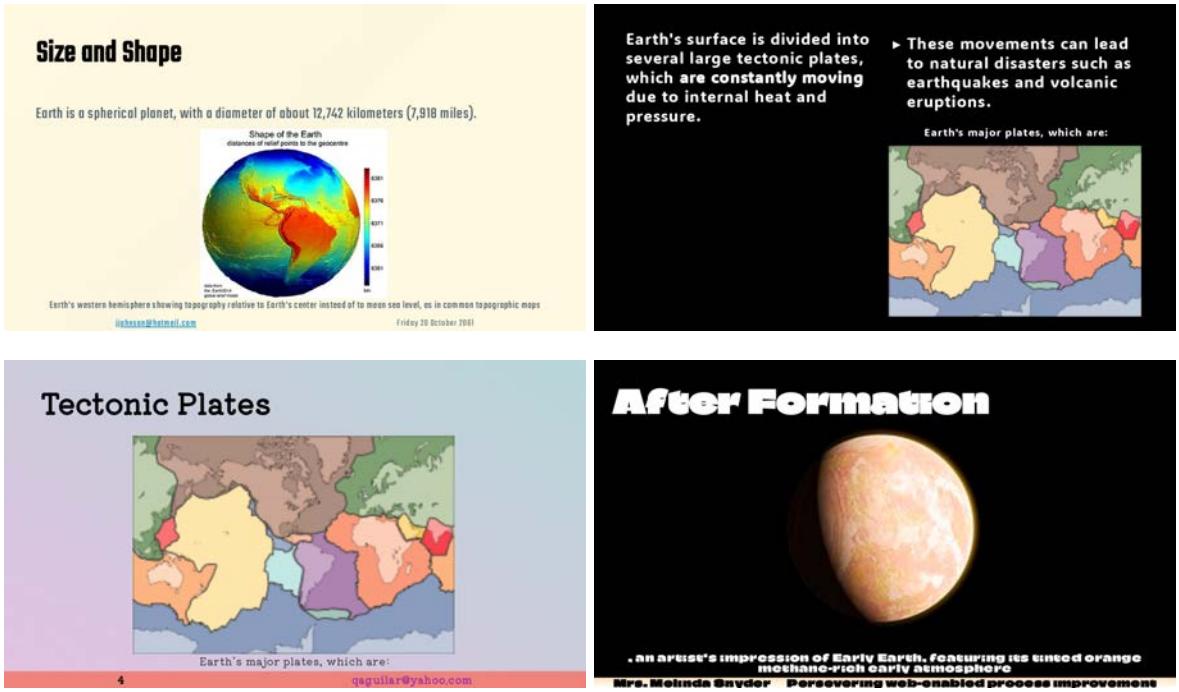


Figure 3.7 – Different styles of slides from the same source (Wikipedia: Earth). We can see diverse fonts and backgrounds, various font sizes and colors, the presence of a footer or not, and different numbers of elements.

Annotations

We generate annotations through the processing of the HTML generated by Marp for each slide. We employ JavaScript to access each text element within the Document Object Model (DOM) and obtain precise bounding boxes for these elements. To enhance accuracy, we modify the HTML by adding span tags around paragraphs and words, enabling us to capture more detailed bounding boxes around the text at a paragraph and word level. This approach ensures precise demarcation of text elements, critical for effective layout analysis.

In parallel, segmentation masks are created by isolating individual elements on the slides with CSS and computing the image difference. This method is particularly effective for identifying and delineating graphical elements within the slides.

For the classification of text elements, we use two information. Primarily, we rely on HTML tags to determine their classes (e.g., 'p' for general text, 'li' for bullet points, 'h1' for titles, etc.). Additionally, we incorporate class information derived during the layout computation phase, further refining the accuracy of our text annotations.

Graphical elements, on the other hand, are annotated based on the classifications available from the labeled images we use from existing datasets. By incorporating these pre-labeled images into our slides, we ensure that each graphical element is accurately categorized, significantly enhancing the reliability of our annotations. We also apply OCR to graphical elements to

extract bounding boxes for any text they contain, enhancing text recognition across diverse slide components.

This comprehensive approach to annotation, combining DOM manipulation, image processing, and existing dataset labels, ensures that we not only create visually varied slides but also provide richly annotated data.

The tool can be customized to generate custom classes as needed as shown in Figure 3.8 where we generate different classes depending on the dataset we want to extend.

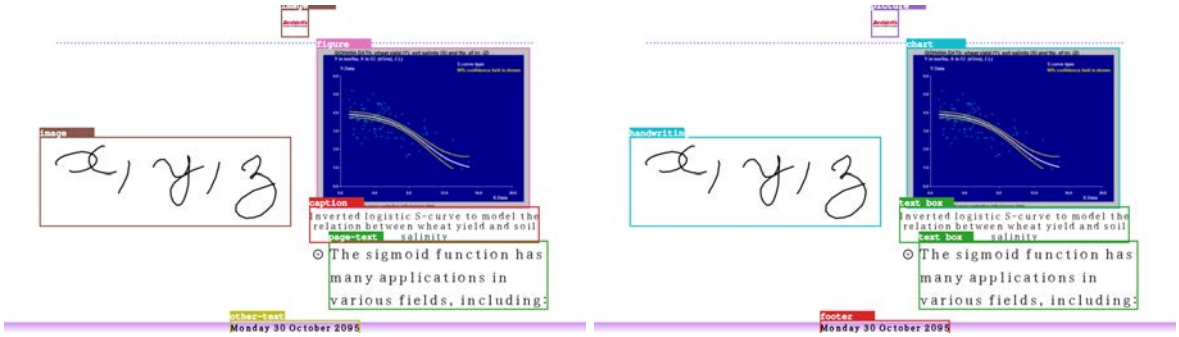


Figure 3.8 – Example of generated slides for SlideVQA annotations (Left) and Fitvid annotations (Right). Handwriting, header, footer, and equation, called Figure in FitVid, do not exist in SlideVQA.

3.4 Experiments

To demonstrate our approach's effectiveness, we evaluate object detection performance using two models and use our generated synthetic dataset to extend two existing datasets.

Datasets

We selected the two largest annotated datasets available. The FitVid dataset, featuring 5,527 images, covers a wide array of 12 classes, including Title, Text Box, Picture, Chart, Figure, Diagram, Table, Schematic Diagram, Header, Footer, Handwriting, and Instructor. Given that there is no test dataset, we create a split by carefully separating slides from the same presentation. We obtain a train set with 4,421 slides and a test set with 1,106 slides. Similarly, the SlideVQA dataset brings together a vast collection of 52,480 slides, categorized into 9 distinct classes: Title, Page-Text, Obj-Text (text on graphical elements), Caption, Other-Text, Diagram, Table, Image, and Figure. The authors provide a train of 37,023 images, a validation set of 5,839 images, and a test set of 7,727 images.

Models

We selected FasterRCNN (Ren et al. 2015) with a ResNet50FPN (He et al. 2015, Lin et al. 2017) backbone and YoloV8 (Jocher et al. 2023) as our models, noting that FasterRCNN tends to be more sensitive to class imbalance, while YoloV8 appears to show less sensitivity, probably

due to the use of focal loss. We trained FasterRCNN with a batch size of 16 using the SGD optimizer, starting with a learning rate of 0.005 and a momentum of 0.95, coupled with a Cosine Annealing Learning Rate scheduler. Data augmentation techniques included color jittering, random gamma adjustments, and brightness and contrast modulation, alongside blurring. For YoloV8, we used a batch size of 64 and adhered to standard settings but changed the input resolution to 1280 and omitted mosaic and flip augmentations. We trained both models for 50 epochs with an early stopping (patience of 5 epochs).

Experiments

Dataset completion. Utilizing our system, we generate 25,000 synthetic slides for each dataset, enriching them with classes underrepresented in existing data, such as Diagram, Table, Plot, Handwriting, and Equation, by incorporating additional datasets for balance. We specifically target the underrepresented classes with the probability system while trying to fit the original dataset by adjusting the layout and style components. For example, in SlideVQA, to imitate the Obj-Text class, we run publicly available OCR Tesseract (Kay 2007) on top of all the graphical elements. To imitate captions, we include a randomly generated smaller text close to the top or the bottom of a graphical element. We now compare the performance of each model trained on each dataset with and without added synthetic slides, evaluating their respective test datasets.

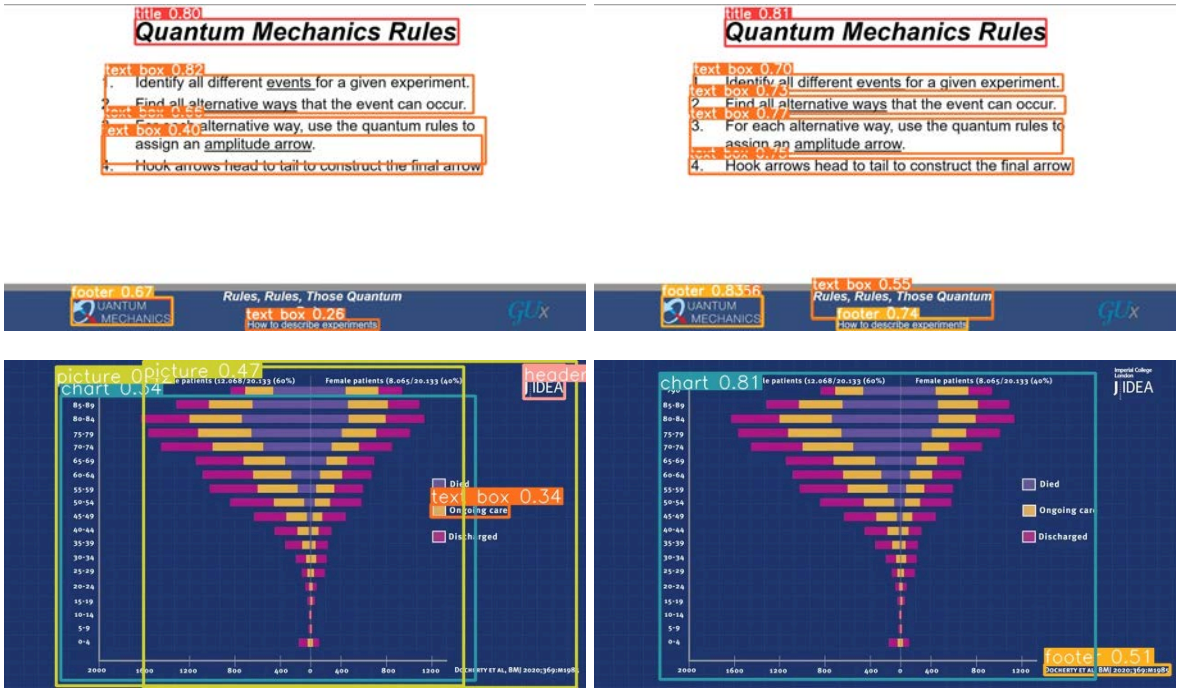


Figure 3.9 – Predictions on FitVid test dataset with our models. **Left**: Inference with YOLOv8 trained without added synthetic slides. **Right**: Inference with YOLOv8 trained with added synthetic slides. The model trained with synthetic data from our approach demonstrates higher precision in box predictions, correlating with the marked improvement in mAP scores.

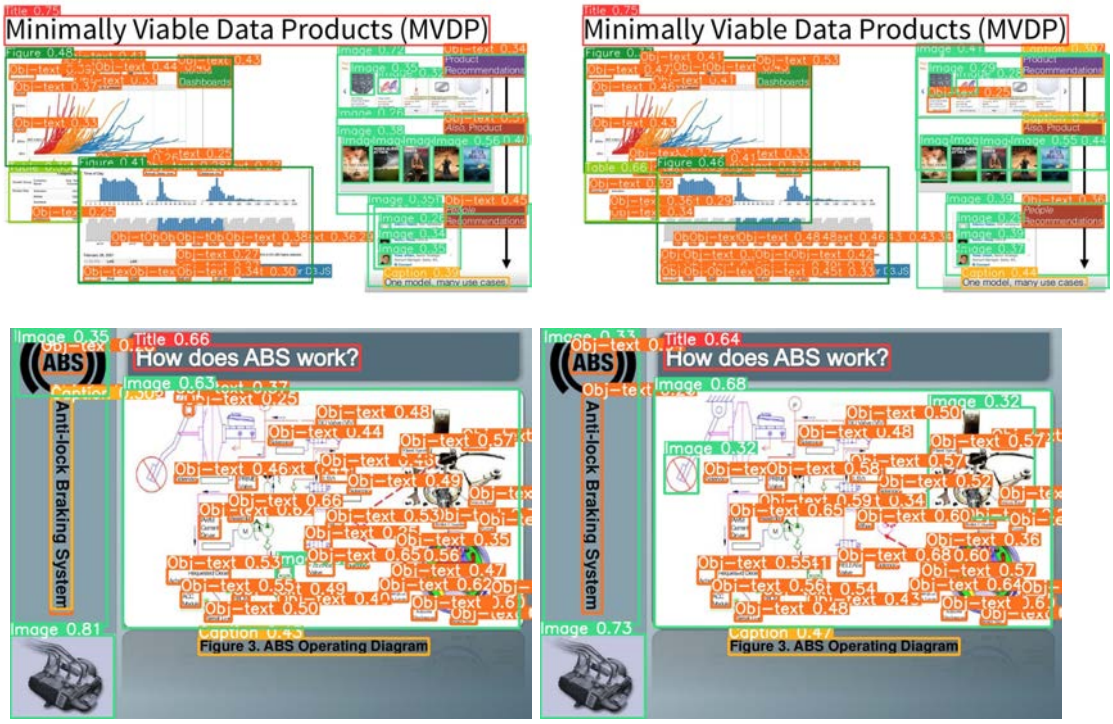


Figure 3.10 – Predictions on SlideVQA test dataset with our models. **Left**: Inference with YOLOv8 trained without added synthetic slides. **Right**: Inference with YOLOv8 trained with added synthetic slides. We can see that the model trained with synthetic data contradicts the ground-truth, despite looking accurate. For example, here, we predict more Obj-Text with added synthetic slides, but the ground truth in Figure 3.1 does not contain these elements.

Dataset	FasterRCNN			YoloV8		
	Original mAP50	Mix mAP50	Improve-ment	Original mAP50	Mix mAP50	Improve-ment
SlideVQA	0.641	**0.645**	0.59%	0.682	**0.685**	0.44%
FitVid	0.485	**0.530**	9.22%	0.513	**0.581**	13.26%

Table 3.1 – mAP50 of FasterRCNN and YoloV8 on SlideVQA test set and FitVid test set, training with and without the slides generated by our system

Our first results, displayed in Table 3.1, reveal that mixing generated data into training can lead to superior performance. With synthetic slides, SlideVQA observes a slight improvement of the mAP50 by 0.59% and 0.44% with FasterRCNN and YoloV8 respectively. Conversely, FitVid benefits significantly from the synthetic slides, showing an improvement of the mAP50 by 9.22% and 13.26% for FasterRCNN and YoloV8, respectively. This gain is illustrated in Figure 3.9. These outcomes underscore the importance of large datasets for achieving optimal performance, as evidenced by the more pronounced improvements in FitVid compared to SlideVQA when training with an extended and larger dataset.

The marginal gains on SlideVQA suggest its extensive size may inherently limit performance improvements. However, we think that the dataset inconsistencies likely skew results. For example, we get a lower mAP50 (-0.3%) for the Obj-Text class on our model trained with generated slides than the one trained solely with the original dataset. Despite this, our predictions, as highlighted in the images from Figure 3.10, show that we predict Obj-Text elements that should be correct based on how this class was defined. We also predict two Figures and a Table, which seem correct. However, as showcased in Figure 3.1, these Obj-Text elements do not exist in the ground truth, and the whole left part of the slide is labeled as Diagram. Such cases happen in other images, as highlighted by Figure 3.1.

Our first experiment shows that extending a dataset with generated data can improve performance, especially on smaller datasets.

Ground Truth Data ratio. Furthermore, we conduct an additional experiment where we vary the ratio of real to synthetic slides from SlideVQA to see the impact of our tool on different dataset sizes.

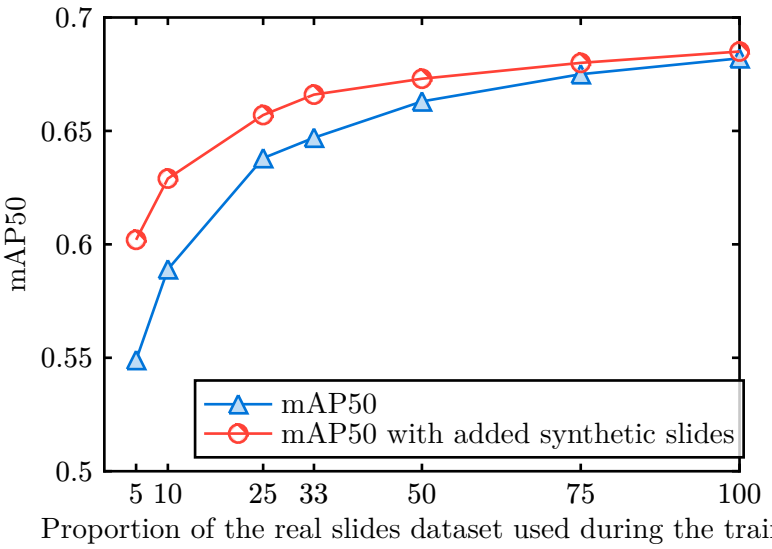


Figure 3.11 – Comparison of mAP50 obtained after training YoloV8 with and without added 25,000 generated slides. Testing on SlideVQA test dataset. The results show a direct correlation between dataset size and the impact from adding synthetic slides: smaller datasets see greater benefits from the inclusion of synthetic slides.

The findings from our second experiment, illustrated in Figure 3.11, reveal that smaller datasets experience more significant enhancements when trained with synthetic slides. Our performance only decreases by 2% when we train with 50% of the SlideVQA dataset augmented with synthetic slides, compared to training with the entire SlideVQA dataset.

This highlights our tool's potential to significantly reduce the necessity for manually labeled slides in training object detection models, while potentially improving

performance. This efficiency suggests a strategic approach in dataset annotation, emphasizing quality and consistency over sheer volume.

Balancing dataset. We also study whether balancing the datasets affects the performance of FasterRCNN. To achieve this, we created additional synthetic datasets comprising 25,000 slides each, deliberately designed to match the imbalance of the original datasets. The new sets were then combined with the original SlideVQA and FitVid datasets to observe the effects of skewed class distribution. We then train a FasterRCNN on the imbalanced datasets.

Dataset	Original	Imbalanced	Balanced
SlideVQA	0.641	0.637	**0,645**
FitVid	0.511	0.522	**0.530**

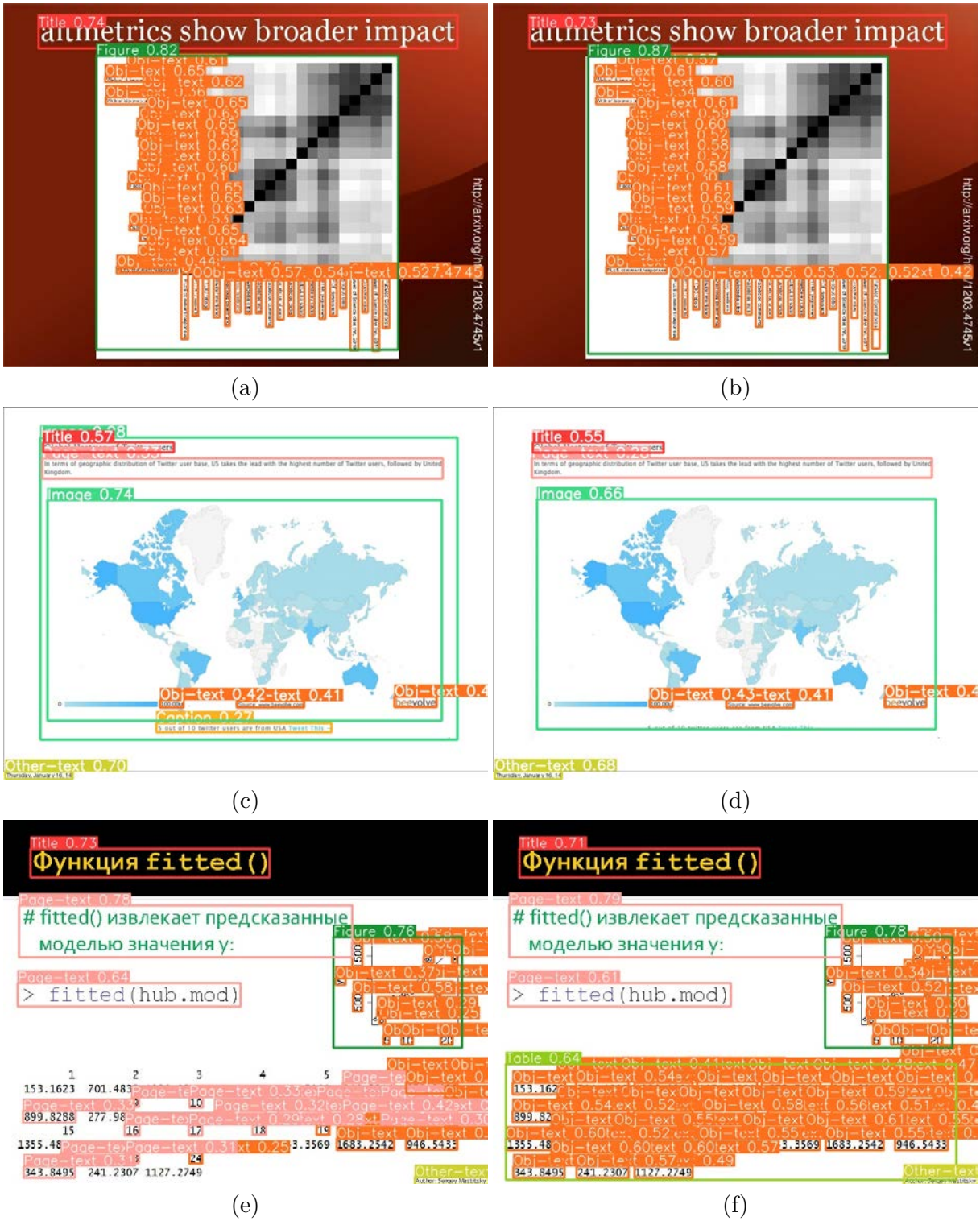
Table 3.2 – Comparison of mAP50 obtained after training FasterRCNN on original dataset, balanced dataset and imbalanced dataset. The balanced and imbalanced dataset were created after adding generated slides with different distribution. The distribution was chosen to balance or not the original dataset.

The results shown in Table 3.2 reveal that training on balanced datasets give a slightly better performance on FasterRCNN. For SlideVQA, the model trained on the imbalanced dataset has a mAP50 of 0.637 which is slightly lower than training only on the original SlideVQA dataset. The FasterRCNN trained on the more balanced dataset, however, displays a mAP50 of 0.645 which is slightly better. For FitVid, FasterRCNN gets better performance with added synthetic slides but the balanced dataset gets the best performance. This suggests that balancing the dataset has a positive impact on performance, while an imbalanced dataset can lead to worse performance.

This result underscores the value of the probability system in enabling dataset balancing.

3.5 Qualitative results

We show qualitative results on some slides from SlideShare-1M in Figure 3.12 and Figure 31. In Figure 3.12, on the images (c) and (d), our best model is more accurate but we lose the caption element. On images (e) and (f), our model predicts more Obj-Text elements and we detect the table.



(a) (b)

(c) (d)

(e) (f)

Figure 3.12 – Predictions on Slideshare-1M. Left: Predictions with YoloV8 only trained on SlideVQA. Right: Predictions with YOLOv8 trained on SlideVQA and added synthetic data.

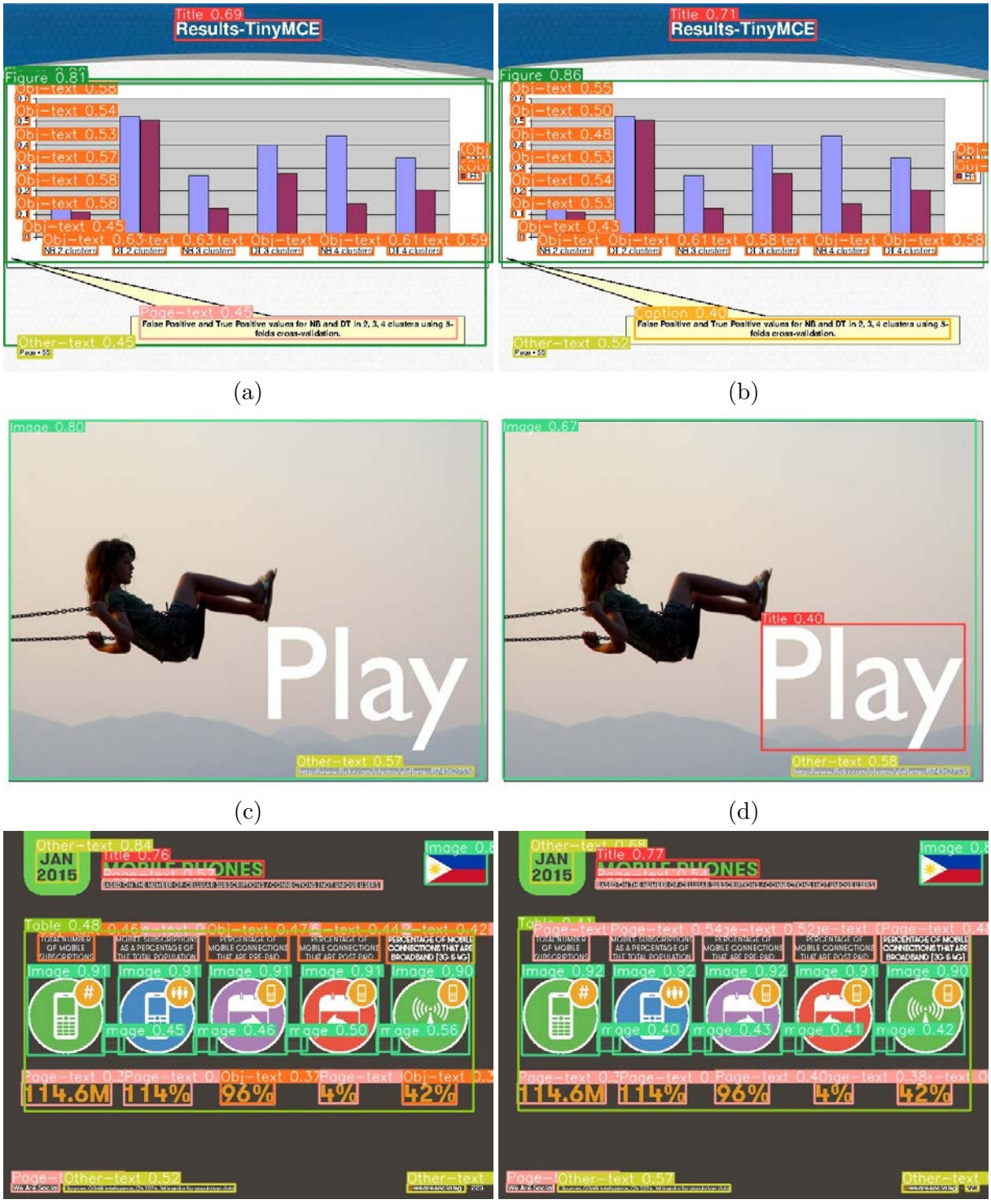


Figure 3.13 – More predictions on Slideshare-1M. Left: Predictions with YoloV8 only trained on SlideVQA. Right: Predictions with YOLOv8 trained on SlideVQA and added synthetic data.

In Figure 3.13, on images (a) and (b), our best model is more accurate in terms of boxes and we detect a caption wheras the other model only detects a Page-Text. On the images (c) and (d), our model detects a correct Title whereas the model trained without generated slides does not.

In the images (e) and (f), our model only detects Page-Text whereas the other model detects Page-Text and Obj-Text elements, showing that our model outputs more consistent labels. Both models detected a Table. These qualitative results show that our system improves the reliability and quality of the predictions, even on external datasets.

3.6 Other types of annotations

Although not extensively studied, our tool also possesses the ability to generate other types of annotations such as pixel-level segmentation masks and slide to Markdown. We show some qualitative results for both type of annotations as there is no good datasets to evaluate against.

3.6.1 Segmentation masks

Our tool can generate both instance and segmentation masks as shown in Figure 3.14 and Figure 3.15 for any classes like bounding boxes.

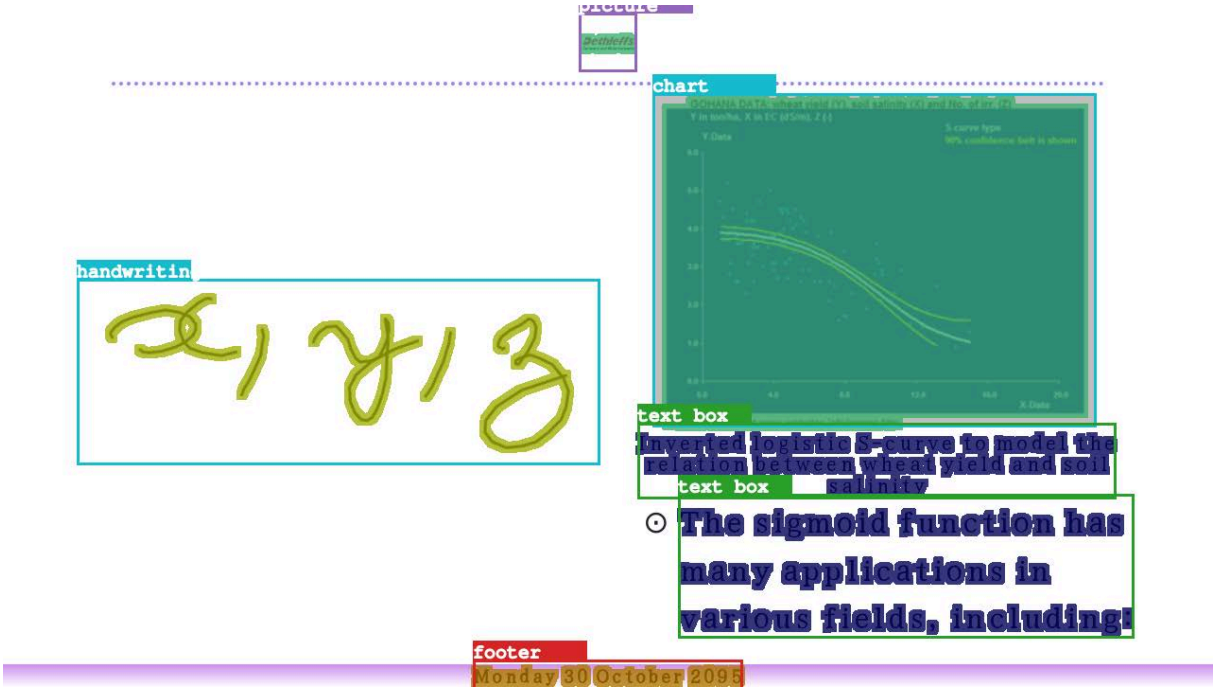


Figure 3.14 – Generation of instance segmentation masks

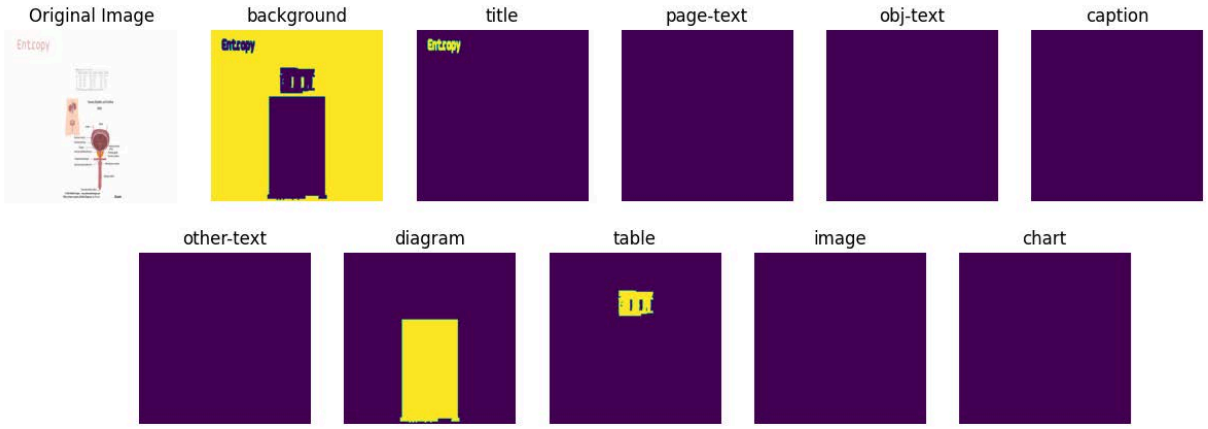


Figure 3.15 – Generation of semantic segmentation masks

3.6.2 Slide to Markdown

We also explored the generation of structured text from slide images using our synthetic dataset. Specifically, we trained a Pix2Struct model (Lee et al. 2023) to convert slide images into Markdown-formatted text, as illustrated in Figure 3.16. While no formal evaluation was conducted, this experiment demonstrates the potential of our dataset for training models capable of structured content extraction.

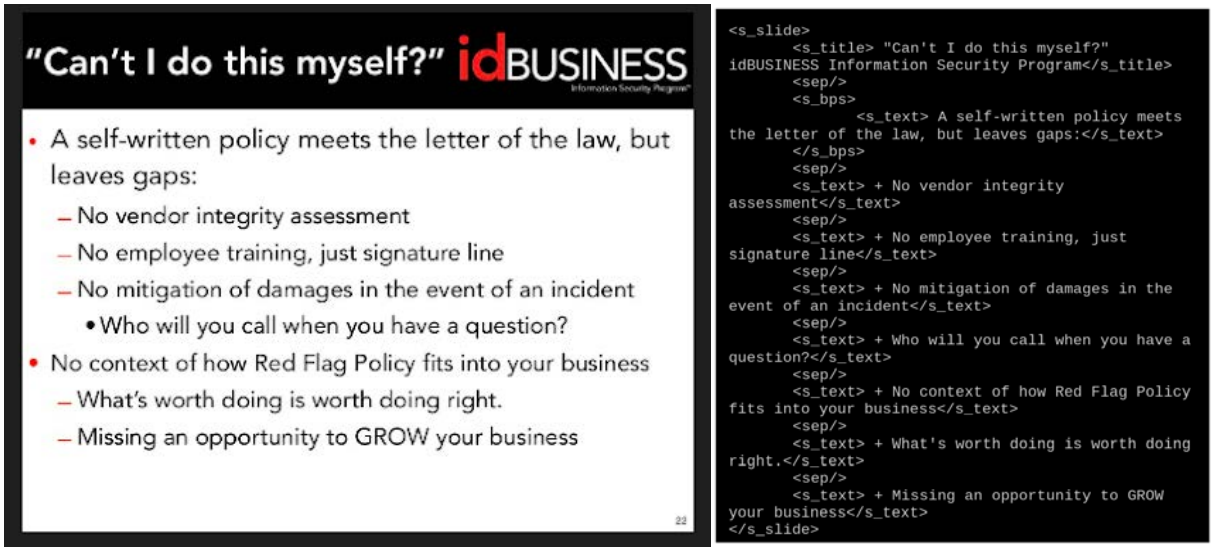


Figure 3.16 – Generation of Markdown text corresponding to the image

3.7 Discussion and Future Work

We demonstrated that integrating synthetic data generation enhances existing datasets while minimizing the need for extensive manual labeling. Prioritizing high-quality annotations and supplementing them with generated data emerges as a strategic approach to boost model performance. However, challenges persist, notably the inconsistencies within and between datasets, particularly in how ambiguous classes such as diagrams are annotated, and the lack

of standardized classes for dataset creation. Our tool offers a solution by accommodating these discrepancies, enabling researchers to tailor datasets to specific needs and applications, marking a step toward more standardized and effective dataset creation for object detection models. This adaptability paves the way for more focused and customized research, enhancing the overall quality and applicability of object detection within diverse contexts. In order to further enhance our tool's capabilities, we see potential in refining the generation of specific classes, such as captions, headers, and footers and expanding the diversity of layouts and styles to better mirror real-world presentations. Another potential use of our tool is to enhance presentations' clarity and adaptability by recreating the original slide, aiming to improve accessibility for diverse audiences.

Summary

The increasing amount of slide presentations in various sectors has amplified the need for effective slide layout and semantic analysis. However, we found that current slide datasets contain inconsistencies, mislabels, and incomplete annotations. Using them as a basis for developing deep learning-based slide analysis models could lead to models that are not robust and suboptimal. Addressing these challenges, we introduce a tool for creating synthetic slide datasets that imitate real-world presentations with different types of annotations: boxes, segmentation masks and text. This tool overcomes the drawbacks of existing datasets by allowing users to create balanced, diverse, and accurately annotated slide data. We demonstrate its efficacy in enhancing slide layout analysis algorithms, focusing on its capability to improve dataset quality and object detection performance. Our code and a demo can be found at this https://travisseng.github.io/slidecraft.

Chapter 4

Temporal Video Lecture Segmentation

Lecture videos are a rich source of educational content, offering unique opportunities for various tasks such as transcription, summarization, and content retrieval. Among these, temporal video segmentation enables the organization of long, unstructured recordings into meaningful segments based on topics or themes. This task is important for educational videos, where clear segmentation facilitates improved navigation, indexing, and summarization, making it easier for learners and educators to access specific content.

In this chapter, we focus on temporal segmentation as it not only addresses the challenges of educational video content—such as varying presentation styles and long durations—but also lays the groundwork for subsequent tasks in video analysis.

4.1 Introduction

With the growth of online education and video-based learning, understanding and organizing the content of educational videos has become increasingly important. Temporal video segmentation, a long-standing problem in the signal processing community (Koprinska and Carrato 2001), plays a role in enabling this understanding by dividing videos into meaningful segments. Traditionally, this task has been approached as shot detection, which focuses on identifying salient cuts within the video. While this method works well for certain types of videos, such as movies, it is less suited to educational videos, where the structure and content can vary significantly.

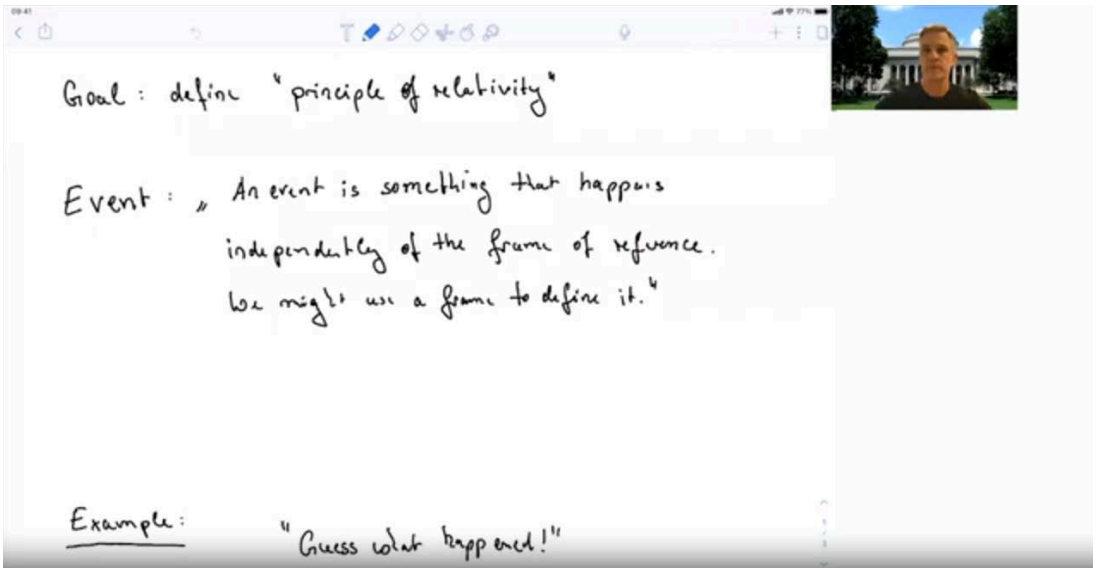
For instance, an in-class recording may have no visible cuts but still consist of multiple semantic segments corresponding to different topics. Conversely, a video composed entirely of slides might contain numerous cuts, many of which correspond to the same overarching topic. These challenges highlight the need for segmentation approaches tailored specifically to the unique characteristics of educational videos.

Lecture video segmentation presents unique challenges due to the reliance on multimodal cues, including text, audio, and visual features, as well as the natural structure of lecture videos. Unlike other types of content, lecture videos are tightly tied to the progression of the discussion, where the instructional material (such as slides or blackboard content) and the lecturer's speech revolve around the same topic and gradually shift to new concepts. Existing approaches often utilize supervised learning but face significant limitations, including the need for extensive annotated datasets and poor generalization across diverse lecture formats. Unsupervised methods, which leverage multimodal signals without requiring labels, offer a promising alternative.

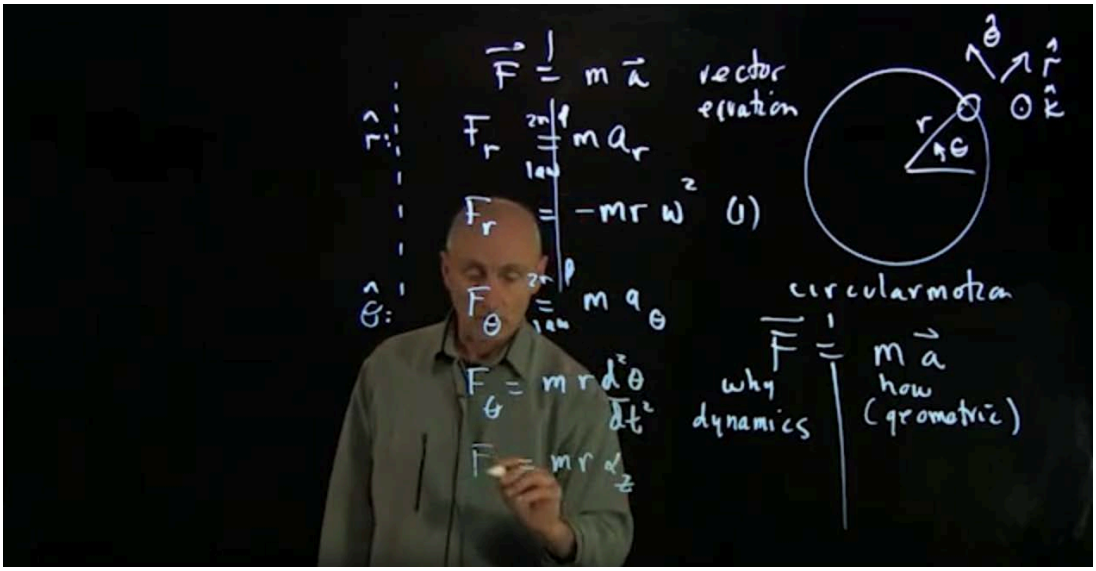
In this chapter, we introduce a lecture video segmentation approach that builds upon the unsupervised state-of-the-art algorithm by (Gupta et al. 2023) to better handle different types of lecture video. To extend their approach, we asked the following question to uncover the features

it uses: **What features are the most relevant? Can we add other useful information? Do some types of lectures benefit more from specific features?**

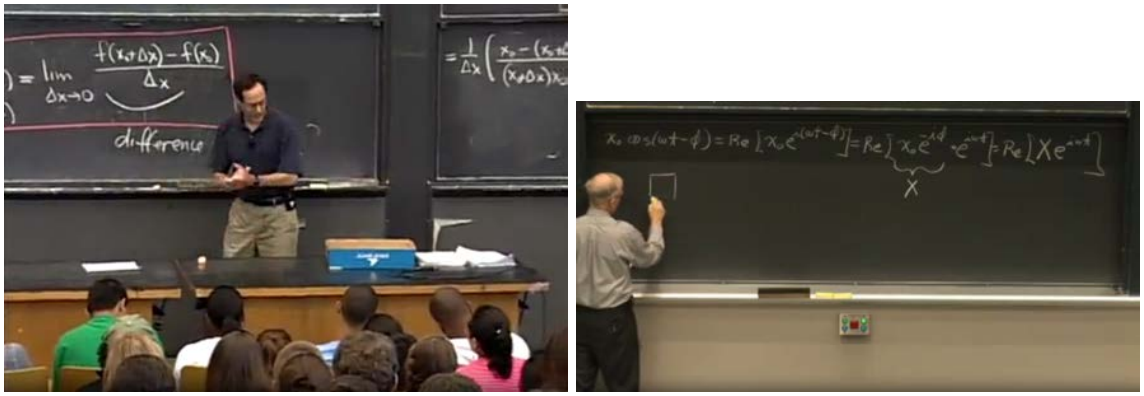
In this chapter, we aim to address these questions, delving into an analysis that extends the understanding of modality contributions in lecture video segmentation through experimentation with new datasets.



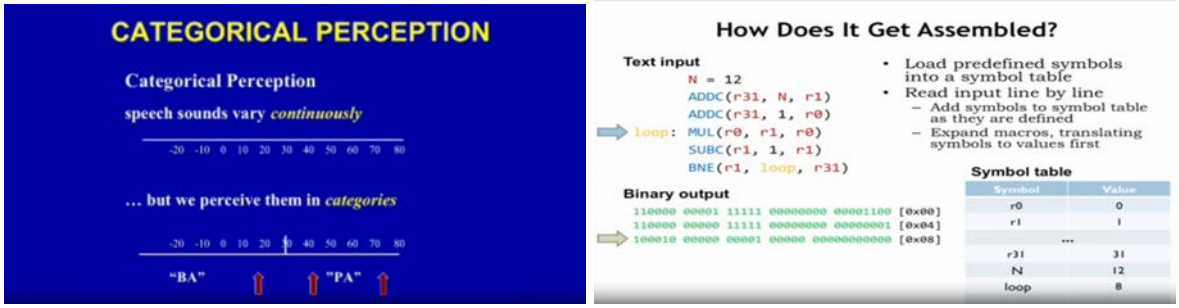
Computer screen video



Transparent screen video



Blackboard videos



Slides video

Figure 4.1 – Different types of lecture videos in the dataset from AVLectures Dataset (Gupta et al. 2023).

Our proposed method begins by dividing the lecture video into smaller segments of 10–15 seconds. For each segment, we extract embeddings from both the visuals (OCR, image content) and the audio (transcripts) to align them. The aligned embeddings are combined to represent the content of each segment and then temporally clustered, with transition information, to group segments based on their content and structure, resulting in a final segmentation of the lecture video. We also introduce two datasets to further study the impact of features extracted from lectures: image, OCR, transcript, and video.

Our contributions include: (1) extending the evaluation of AVLectures to new datasets, (2) introducing enhancements to multimodal feature alignment and clustering depending on the lecture type, and (3) demonstrating improved performance across diverse lecture video types.

The remainder of this chapter is organized as follows: Chapter 4.2 reviews existing work, Chapter 4.3 reviews existing and new datasets, Chapter 4.4 describes our approach, and Chapter 4.5 presents comprehensive evaluations and ablation studies.

4.2 Related Work

4.2.1 Text Segmentation

Text segmentation is a valuable tool for dividing lecture transcripts into coherent segments based on topics or subtopics, providing a baseline for segmenting the corresponding video content. In the context of lecture video segmentation, text segmentation serves as an effective approximation for analyzing the speech component, which plays a central role in conveying the discussion's structure and flow. By identifying topic boundaries within the transcript, these methods offer an initial framework that can later be refined using additional modalities such as visual and audio features. Text segmentation itself is a well-studied task in natural language processing (NLP), with approaches ranging from supervised models trained on labeled datasets to unsupervised methods that rely on semantic and lexical features.

Supervised approaches to topic segmentation frame the task as a binary classification problem, predicting whether a sentence represents a topic boundary. Traditional methods relied on feature engineering, while modern techniques utilize neural architectures to encode text representations.

For instance, (Koshorek et al. 2018) employed a bidirectional LSTM-based model for sentence-level predictions, demonstrating competitive performance on datasets like Wiki-727k. Building on this, (Glava s and Somasundaran 2020) proposed a Transformer-based architecture with a dual-level modeling approach—at the token and sentence levels—and an auxiliary coherence objective to better capture contextual information. These methods have shown strong results but depend heavily on labeled training data, which can limit their application in new domains.

Unsupervised methods address the challenge of limited labeled data by leveraging statistical and semantic properties of text. Techniques in this category include approaches based on lexical cohesion, such as TextTiling (Hearst 1997), which segments text by analyzing patterns of cohesion across word distributions. This idea is further extended by methods like LCseg (Galley et al. 2003), which incorporate lexical chains to more effectively identify topic boundaries. Another common approach involves topic modeling, where algorithms like TopicTiling adapt the principles of TextTiling by utilizing latent topics derived from models such as Latent Dirichlet Allocation (LDA) to detect shifts in thematic content. Graph-based methods, exemplified by GraphSeg (Glavaš et al. 2016), represent sentences as nodes in a graph with semantic relationships forming the edges, enabling the clustering of nodes into coherent topic segments. More recent advances focus on semantic similarity, leveraging embeddings from models like BERT to compute the coherence between sentences or text blocks and identifying topic boundaries where semantic consistency decreases significantly (Solbiati et al. 2021). While unsupervised methods

are flexible and domain-agnostic, they often fall short of the precision achieved by supervised models in scenarios where high-quality annotated data is available.

Commonly used datasets for topic segmentation include Wiki-727k (Koshorek et al. 2018), CHOI (Choi 2000), and QMSum (Zhong et al. 2021), which offer a mix of synthetic and real-world text. These datasets have supported the development of both supervised and unsupervised models across domains such as news articles, Wikipedia entries, and meeting transcripts.

These methods in text segmentation, particularly those leveraging semantic similarity, serve as a foundation for addressing segmentation challenges in multimodal contexts, such as video.

4.2.2 Temporal video segmentation

Temporal video segmentation extends segmentation tasks into the domain of video content, aiming to group consecutive frames into coherent segments to facilitate indexing and summarization. A large body of work is devoted to the segmentations of movie-like videos, in which the problem is often broken down into two-level detection: shot detection followed by scene detection. In movie editing, a shot represents a continuum of frames between two splices and can last from less than a second to several minutes (sequence shots). In the literature, shot detection is often cast as splice detection, which can be performed either by looking for large color changes between two consecutive frames (Cotsaces et al. 2006), or by training a binary classifier (Souček and Lokoč 2020). Detecting scenes requires grouping neighboring shots, which form a semantic unit. This step is usually harder, as the shots constituting a scene may look and sound very different. A standard approach to this problem is to extract features at the shot level and group shots with similar features. (Rao et al. 2020) uses several models to extract features representing characters, place, action, etc., and uses dynamic programming to compute an optimal grouping of shots into scenes. (Chen et al. 2021b) extracts shot visual features from uniformly sampled key frames, as well as audio features from a spectrogram representation of the audio, and learns inter-shot similarity using contrastive learning in a self-supervised fashion. Using similar shot features, (Liu et al. 2022b) uses a correlation network and predicts whether two consecutive shots belong to the same scene or form a transition between scenes.

There are many other variants of the temporal video segmentation problem, including video action segmentation, in which each segment should also be labeled with the action performed (Chen et al. 2020, Gao et al. 2021). Video Procedure Segmentation is similar in essence but does not need to label the segments with predefined labels. (Ji et al. 2022)

4.2.3 Lecture video segmentation

Lecture video segmentation is yet another variant of the video segmentation task. Even if the problem can be framed similarly, lecture videos are very different in practice from other videos.

First, lectures videos are diverse and can range from simple commented slides to a sequence shot of in-class recordings. Second, the underlying structure of the video is much more dependent on the audio (more specifically, the spoken content) than on the visual cues. The most important visual element of lecture videos is often the text, either on a slide or a blackboard. Some methods (Che et al. 2013, Das and Das 2020) only look at text extracted from the slides using Optical Character Recognition (OCR) to perform segmentation. (Gandhi et al. 2015) further relies on the visual appearance of texts (e.g., large, bold, underlined) on the slides to predict their saliency and thus their importance. They then use word saliency to estimate the probability of a slide being a topic transition slide. (Jeong et al. 2012) look at the specific problem of recorded slide video lectures, and use SIFT to match consecutive frames that display the same slide. They segment the video based on when the speaker switches to the next slide.

Most of the work uses a transcript of what is said during the video, either provided (as a subtitle file) or extracted automatically using Automated Speech Recognition (ASR). (Das and Das 2019) extract sentence descriptors and learn to classify transitioning sentences from Wikipedia. (Galanopoulos and Mezaris 2019) uses Word2Vec descriptors to represent a window of words from the transcript, and create segments when neighboring windows are particularly dissimilar. (Tuna et al. 2015) use the same idea but consider both transcript and OCR text as features. (Shah et al. 2014) extract the visual features (color histograms) and text features, and train a support vector machine to detect transitions. They later extended their work (Shah et al. 2015) by using additional features extracted from Wikipedia to improve their results.

Some works use audio cues along with the transcript. (Soares and Barrére 2018) perform spectral clustering on audio features, and later add (Soares and Barrére 2019) the transcript modality using Word2Vec to improve their method. (Chand and Oğul 2021) have a similar approach but use a genetic algorithm to generate the segmentation. (Dimitsas and Leidner 2023) detect silences to over-segment the video, and extract keywords from the transcript to compute the similarity between segments.

These prior works on lecture video segmentation lacked a consistent and comprehensive evaluation dataset. However, the work of (Gupta et al. 2023), which serves as our baseline, is an unsupervised approach for lecture video segmentation and provides an evaluation dataset. Their method aligns embeddings from multiple modalities—image, video, OCR, and transcript —using pretrained encoders such as BERT (for text) and ResNet/ResNeXt (for visual features). These embeddings are projected into a shared space and optimized with a max-margin ranking loss. The approach uses self-supervised pretraining on MIT OpenCourseWare videos, followed by finetuning on the evaluation dataset for alignment, requiring no annotations. Their findings show that aligning these multimodal features improves segmentation performance compared to using standalone features, especially after finetuning.

4.3 Datasets

To the best of our knowledge, there is only one annotated dataset for temporal segmentation in video lectures: the AVLectures dataset, which contains a mix of all types of video as seen in Figure 4.1. In this work, we expand the resources available for evaluation and analysis by introducing two new datasets. The first, VideoLectures Slide Dataset, is focused on video lectures with slide-based content. The second, a more diverse dataset drawn from Yale OpenCourses, centers on lectures mostly in social sciences, offering broader topical variation. Together, these additions provide a more comprehensive foundation for evaluating segmentation performance across different content domains.

AVLectures Dataset. We utilize the AVLectures Evaluation Dataset (Gupta et al. 2023) as our primary evaluation dataset. It includes 15 labeled courses from MIT OpenCourseWare, totaling 350 lecture videos, with 93 slide-based lectures and 247 non-slide-based lectures. To our knowledge, this dataset was the most recent and comprehensive available for lecture video segmentation. It covers a range of subjects—including Physics, Economics, Mathematics, and Engineering—and includes different video types, such as computer screen, blackboard, slides, and mixed formats (see Figure 4.1). Specifically, it comprises 157 blackboard videos, 93 slide-based videos, 8 videos with a mix of blackboard and slides, and 12 videos mirroring a computer screen. Each lecture video has ground truth segmentation based on human-annotated YouTube chapters, when available, or by concatenating shorter clips of the same lecture, **which introduce segmentation bias as it can be easily detected with shot boundary detection** and differs from real chapters that can happen without transition. Half the videos from AVLectures exhibit this issue. To ensure fair evaluation and mitigate such bias, we introduce two additional datasets. These datasets include unedited lecture videos with manually curated segment boundaries.

VideoLectures Slides Dataset. In this dataset, we focus on slide-based presentations, where speakers often outline the structure of their talks, as illustrated in Figure 4.2. Using these outlines as references, we listen to the speech to identify topic transitions and annotate the segments accordingly. The dataset comprises 50 slide-based lecture videos with segments. These videos are sourced from VideoLectures.NET and YouTube, spanning topics such as Mathematics, Physics, and Computer Science, with an average video length of 49 minutes. The dataset includes video content, slides, and segment transitions, both within each slide and between chapters.

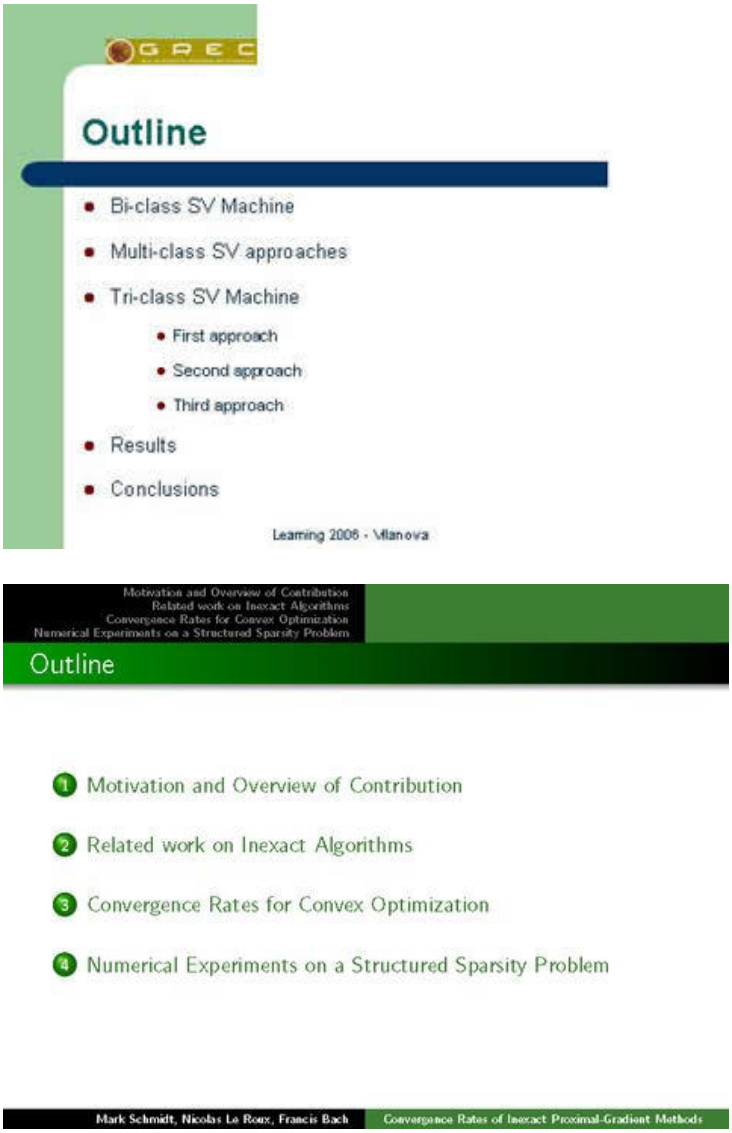


Figure 4.4 – Outlines from slide-based videos.

Yale Dataset. We also propose a new dataset of 920 lecture videos from Yale OpenCourses, which we refer to as the Yale Dataset. This collection encompasses one hour long videos on average on subjects such as History, Literature, Philosophy, Economics, and various other scientific fields, representing a broader range of lecture topics and formats detailed in Figure 4.5. Especially, most of the talks are only focusing on the presenter and his speech. As shown in Figure 4.8, the dataset also includes diverse video types—blackboard videos, presenter-focused videos, and slide-based videos where both the screen and the presenter are visible—introducing a layer of complexity in preprocessing. The distribution of the video types is shown in Figure 4.6. Some videos incorporate edited cuts to alternate camera angles, often to include audience perspectives.

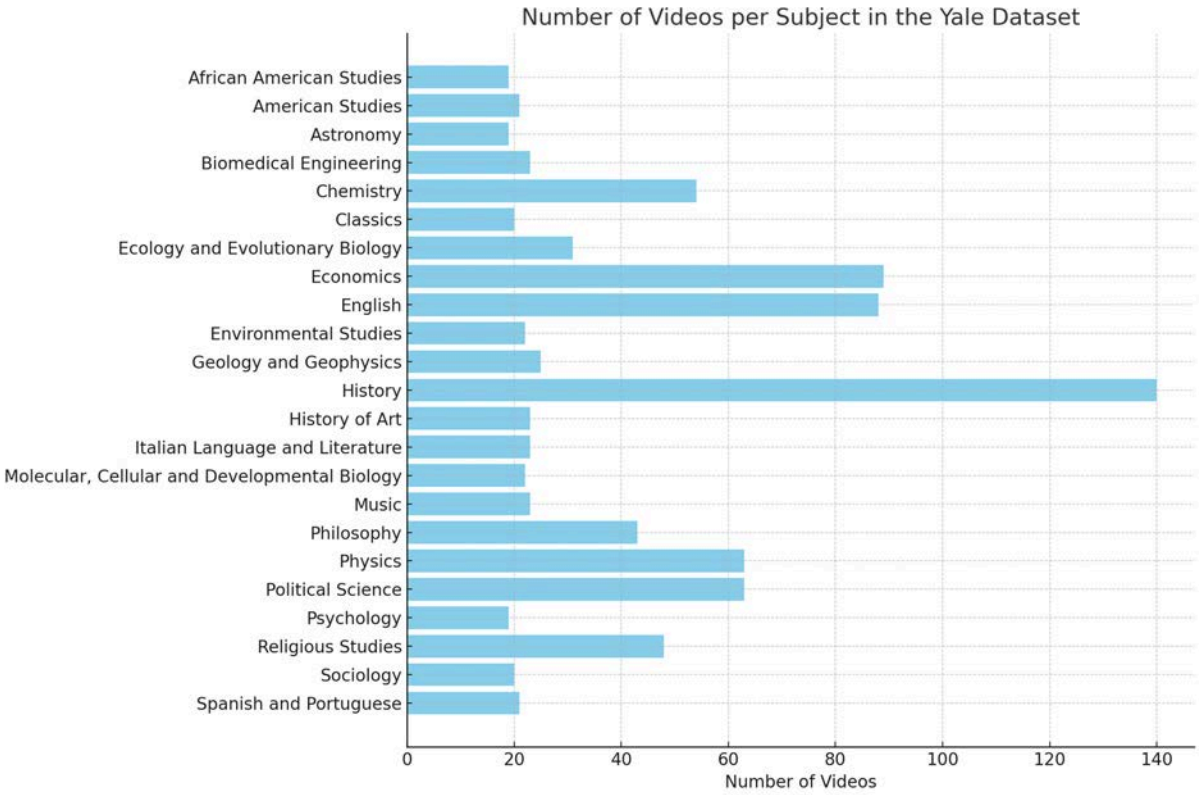


Figure 4.5 – Number of videos per subject in Yale Dataset (920 videos)

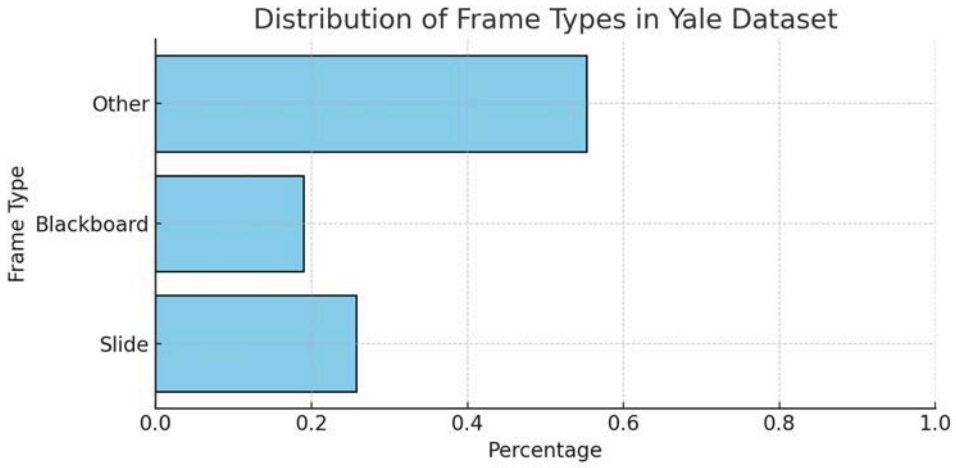
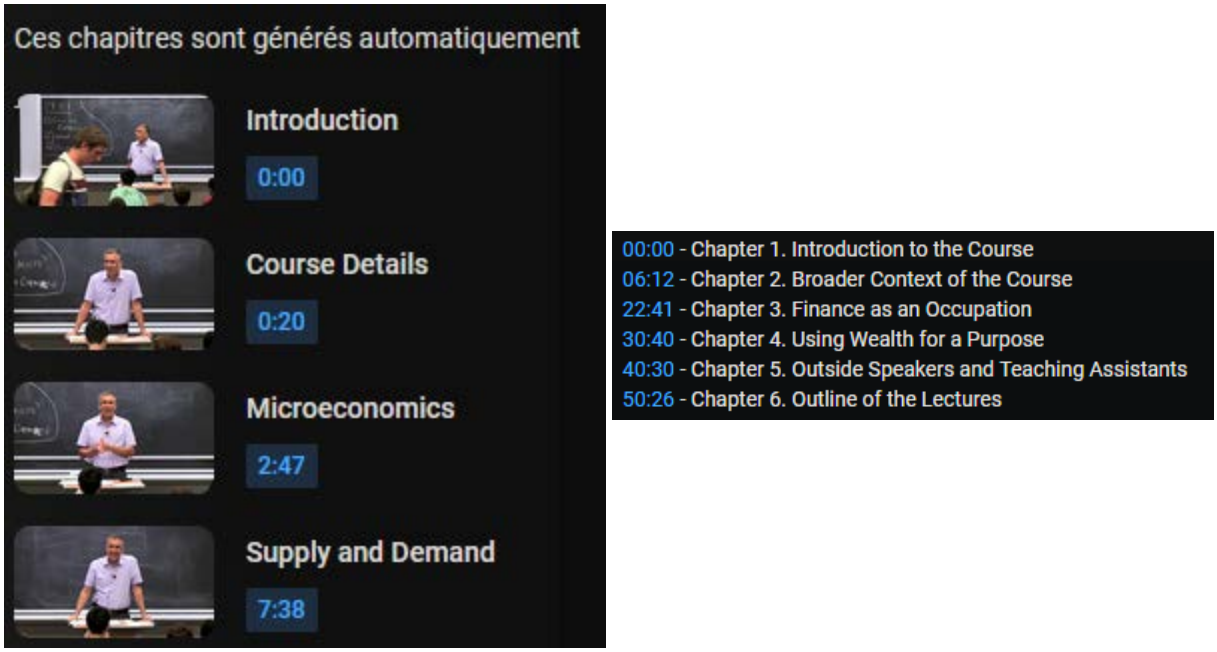


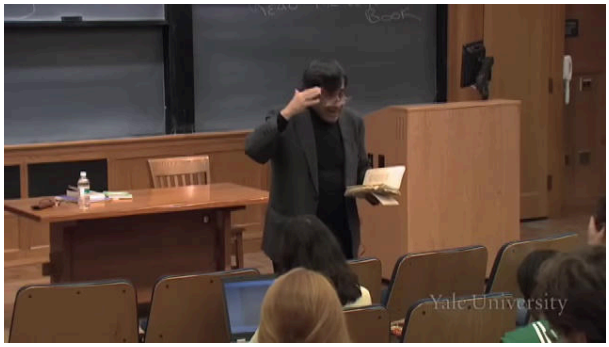
Figure 4.6 – Distribution of frame types in Yale Dataset

Each video in the Yale Dataset is accompanied by segment annotations from YouTube, obtained through direct scraping. YouTube offers a feature for video chapters, which may be automatically or manually added by the video authors. To ensure high-quality annotations, we filtered for manually-created chapters by analyzing the presence of timestamped titles in the video descriptions. We developed a YouTube scraper to collect only those videos with manual timestamps in their descriptions, resulting in 950 well-annotated videos. By analyzing the chapter titles, we further distinguished human-generated chapters (with coherent titles) from automatically generated ones. An example of this distinction is shown in Figure 4.7.

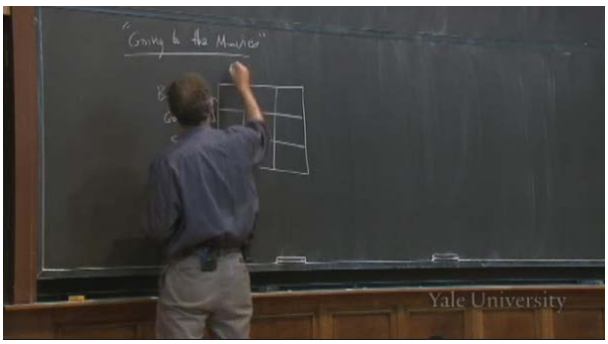


Automatic chapters Manual chapters

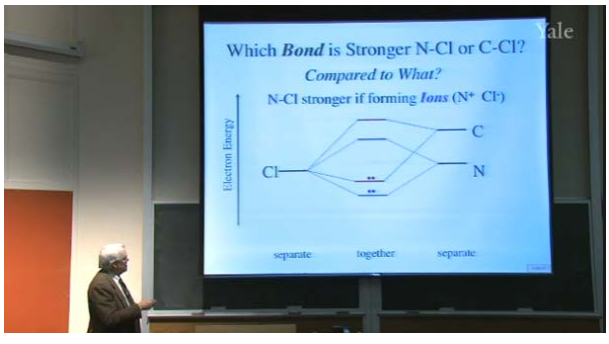
Figure 4.7 – An example of an automatic chapter made by Youtube and manual chapters added by the author of a video. The automatic chapter does not appear in the description. We can also see the difference of quality in the titles: the manual chapters are more detailled, suggesting that it is human made.



ITAL 310: Dante in Translation



ECON 159: Game Theory



CHEM 125b: Freshman Organic Chemistry II



PHIL 181: Philosophy and the Science of Human Nature

Figure 4.8 – Screenshots of videos from Yale Dataset. Blackboard, presenter-focused and slide-based video are present in this dataset.

4.4 System Overview

In this section, we present our system in detail, also presented in Figure 4.9.

4.4.1 Preprocessing

Given a lecture video, we run a preprocessing step to generate essential information for subsequent stages of our system. Throughout this process, transcription, Optical Character Recognition (OCR), transition and semantic elements are directly derived from the video content.

Scene and shot detection

Our process begins with detecting transitions within the video using content-aware scene detection techniques (Castellano 2018). These transitions include build-up slides, new slides, cuts, or camera switches. We do not classify the type of transition; we simply identify the boundaries.

Shot classification

Lectures can be delivered in various formats, including slide-based presentations projected onto a screen, handwritten content on a whiteboard or blackboard, or purely through speech. Video recordings of these lectures may capture the classroom environment, including the screen or board and the speaker, the computer screen displaying the slides, or even just the presenter. Among these formats, slide-based lecture videos tend to be more structured and focused. The transition to a new slide often indicates a shift to a new topic or concept, highlighting the importance of distinguishing between different lecture formats—such as blackboard-based, slide-based, or others—in our approach.

We thus begin by categorizing each segment as slide-based, blackboard or other (no slide or no blackboard). This classification is achieved by extracting frames from the video at one-second intervals to classify these frames. We use the object detection model from Chapter 3 to detect the elements in the frame. As highlighted in the previous chapter, the model can detect persons, handwriting, text boxes, titles, images and even more. We consider the confidence, size, position and number of these detected elements to determine whether or not there is a visible slide or not on a video frame. This is especially important on frame with a presenter that can move in front of the slide.

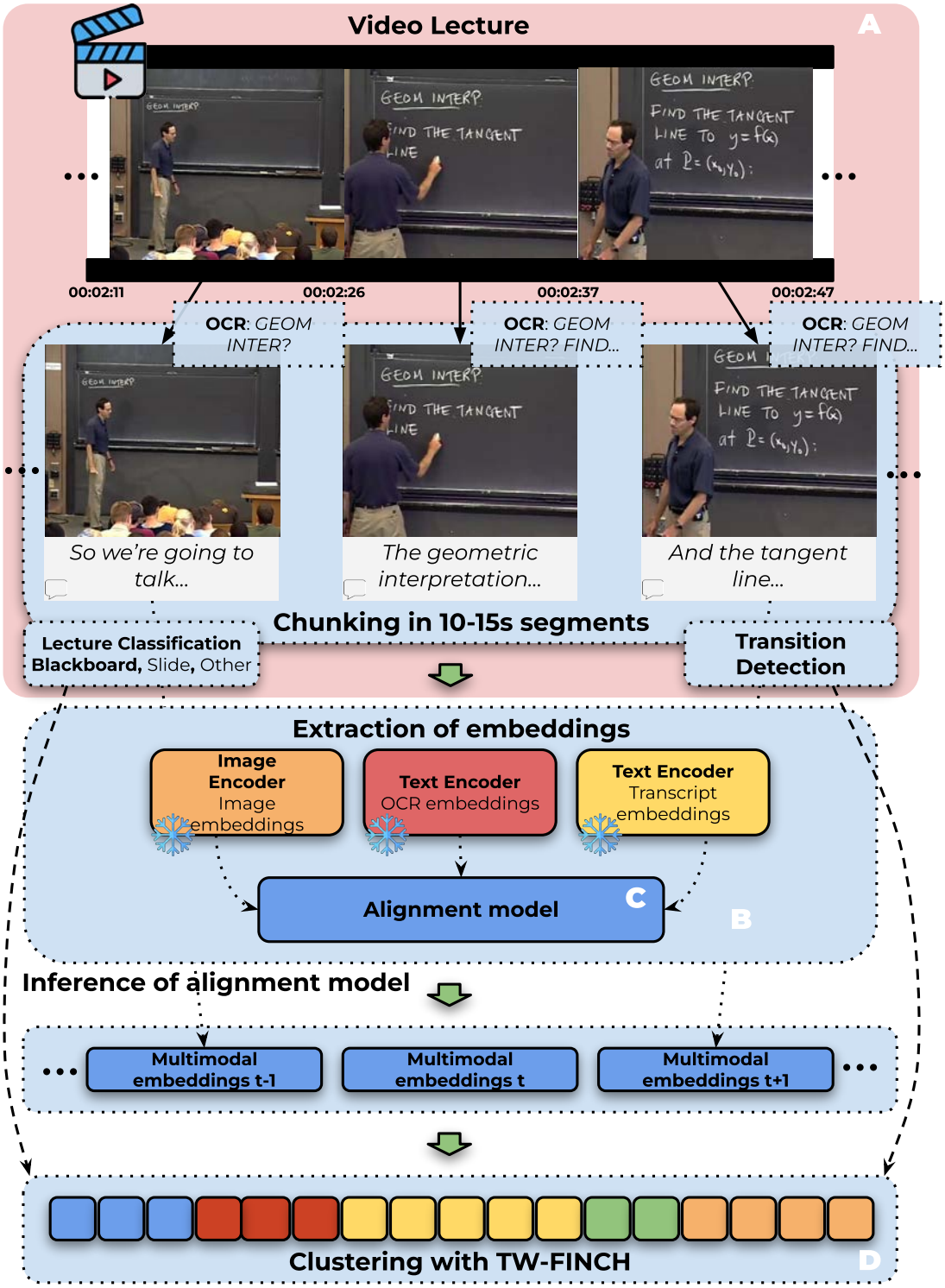


Figure 4.9 – Overview of our system: The video is processed by chunking it into 10-15 second segments, with transcripts generated using an off-the-shelf algorithm. Embeddings are extracted from the visuals (images, OCR) and the transcript. These embeddings are lightly aligned using a shallow network. Finally, the aligned multi-modal embeddings are clustered using the temporal clustering method TW-FINCH with additional features such as transition cuts and pauses.

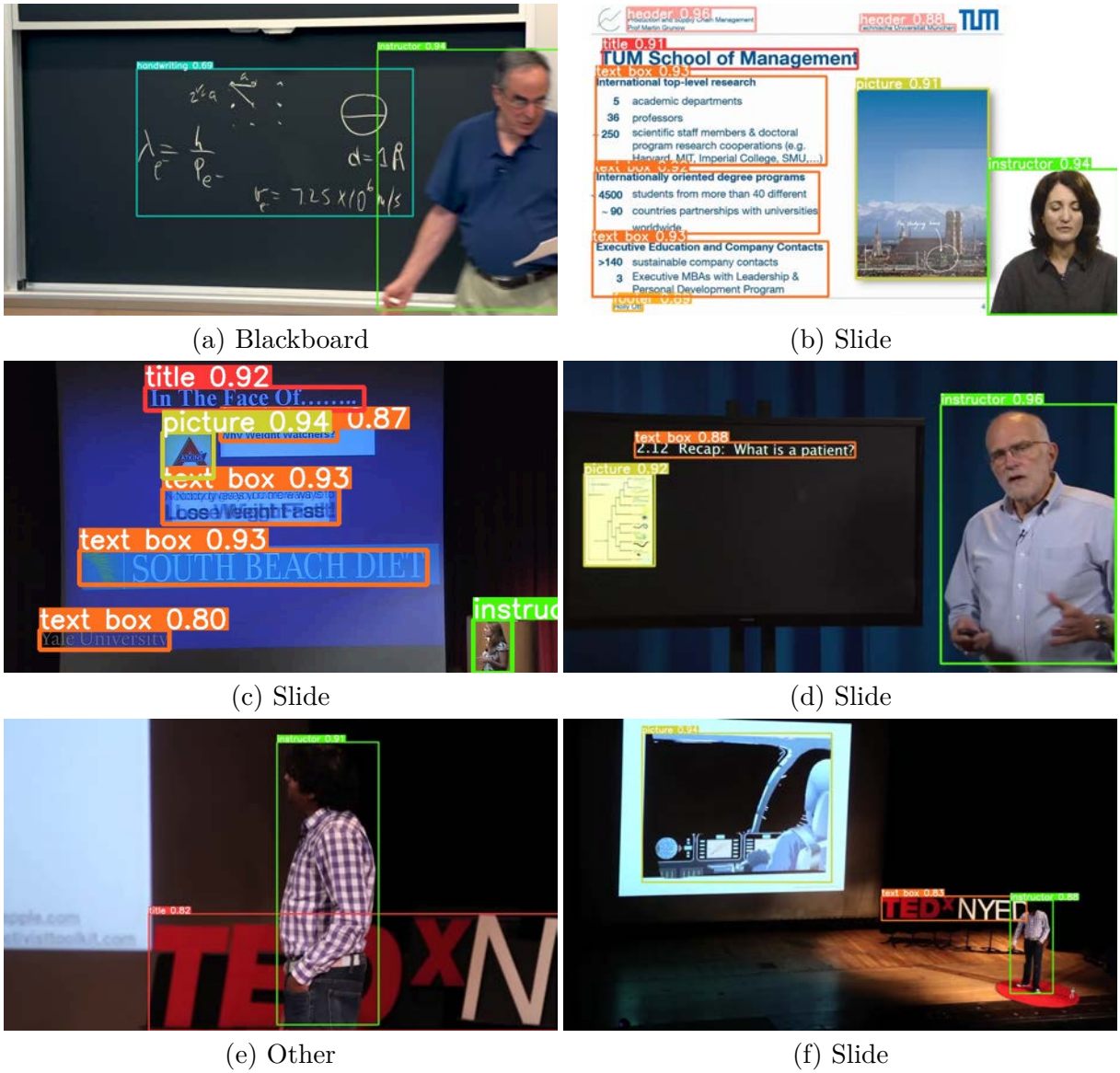
Frame type	Criteria	Example
Slide	• At least one title, image, or text box is present • No presenter detected in the middle of the frame in front of slide elements	Figure 4.10 (b, c, d,f)
Blackboard	• Handwriting is detected	Figure 4.10 (a)
Others	• None of the above	Figure 4.10 (e)

Table 4.1 – Criteria to classify a frame

Table 4.1 shows how our algorithm classifies a frame. A frame is classified as a slide frame if there is no presenter in the middle of the frame and if there is at least one title, one image (note that if the detected image is the whole image itself then we do not count it because it could be a false positive) or one text box. For example, if there is a person detected in the middle of the frame like in Figure 4.10 (e), then we consider it is a non-slide frame. If there is a title or text boxes that are not occluded, then we classify it as a slide frame. We can also classify a frame as a blackboard frame or not if we detect handwriting like in Figure 4.10 (a) but we did not evaluate this class. Examples are displayed in Figure 4.10. We evaluate the slide classification algorithm on an evaluation dataset containing 18 Youtube videos given by (Zhao et al. 2018). The results are shown in Table 4.2.

	Talkminer	Zhao et al	Ours
Precision	81.6	92.6	97.7
Recall	74.7	86.3	97.8
F1	74.2	88.6	97.7

Table 4.2 – Performance of different methods on slide classification on a frame (contains slide or does not contain slide). Average values over 18 videos from (Zhao et al. 2018) evaluation dataset.



(a) Blackboard (b) Slide

(c) Slide (d) Slide

(e) Other (f) Slide

Figure 4.10 – Results of the object detection model trained in last chapter. The number of detected objects, their size, their position and their confidence are used to classify the frame as slide, blackboard or other.

We begin by classifying each frame of the video into one of three categories: blackboard, slide, or other. Our algorithm processes the frames and assigns them to the appropriate category based on visual content.

Slide Element Detection and OCR.

For each segment identified as containing slides, we retrieve the detected objects from our detection model used for slide classification in the previous section. This enables to extract titles, texts, images, and their classes, such as diagrams or tables. For textual elements, TesseractOCR (Kay 2007) is used to extract the text.

4.4.2 Chunking

From a video, we divide the transcript into chunks containing sentences of a minimum duration of 10 seconds while making sure that we do not split a sentence. (Gupta et al. 2023) results show that 10-second segments lead to the best results as it can capture enough semantics but it is not too long to avoid finding the boundaries. The segments are then categorized as **slide**, **blackboard**, or **other** depending on their dominant frame type.

4.4.3 Audio Transcription and Matching.

The audio component of the video is transcribed using Whisper (Radford et al. 2023a), generating a transcript with timestamps for each word. This transcript will be used to match spoken sentences with their corresponding video segments. The text embedding model **all-mpnet-base-v2** (Song et al. 2020) is used on every transcript chunk to get the transcript embeddings (768-dimensional vectors).

4.4.4 Multimodal embeddings

From each chunk, image embeddings and OCR embeddings are extracted. For OCR embeddings, we also use the same model **all-mpnet-base-v2** (Song et al. 2020) to get a 768-dimensional vector. For image embeddings, we extract the last frame of the segment and pass it to a pretrained on ImageNet-1k **DINOv2** (Oquab et al.) model to obtain a 768-dimensional vector.

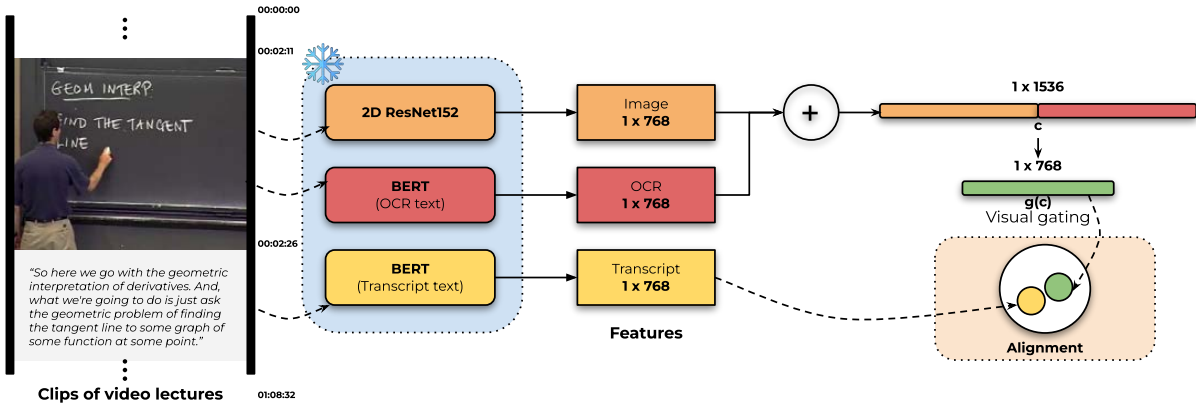


Figure 4.11 – The embedding alignment process: visual (image), OCR, and transcript features are extracted using DINO and BERT models, respectively. These embeddings are concatenated and gated through visual alignment mechanisms to produce a unified representation for clustering. The transcript embeddings are not modified.

4.4.5 Embeddings alignment

Our experiments reveal that incorporating video embeddings increases model complexity without yielding significant performance improvements. Similarly, we found that using a shallow network, rather than a more complex model, better preserves embedding quality and improves clustering performance. Additionally, we experimented with and without projecting the transcript embeddings, as aligning them was observed to potentially compromise their robustness, as demonstrated in the analysis of the Yale Dataset. We project the visual embeddings to the same dimension as the transcript embedding (768) and align them with the original transcript embeddings as shown in Figure 4.11, preserving the robustness of the textual features while ensuring effective multimodal integration.

We follow a similar approach of (Gupta et al. 2023). We use the extracted features from pretrained models. All visual features (Image and OCR) are concatenated into a single embedding. We then learn a projection that incorporates non-linear context gating as follows:

$$g(c) = (W_1^c * c + b_1^c) \cdot \sigma(W_2^c(W_1^c * c + b_1^c) + b_2^c)$$

where W_1^c, W_2^c are learnable weight matrices, and b_1^c, b_2^c are learnable biases. Here, $\cdot$ denotes element-wise multiplication, and σ is an element-wise sigmoid activation function. Then, we normalize each embedding.

Unlike (Gupta et al. 2023), we train our model using a self-supervised approach with the same approach as (Radford et al. 2021), using the **InfoNCE loss** (Oord et al. 2019). This loss encourages alignment between visual features and their corresponding transcript features, while separating unrelated pairs. We also adopt the same hyperparameters. The model is trained for **30 epochs** with a batch size of **1024**, Adam optimizer and a learning rate of **1e-4**.

4.4.6 Clustering

At this stage, we start with an over-segmented video that has aligned multimodal embeddings for each segment. Preliminary experiments on the datasets detailled in Chapter 4.5.3 suggest that certain types of lectures benefit more from specific features. Table 4.3 shows the optimal combination of features derived from our observation. For example, for SLIDE and BLACKBOARD videos, combining the transcript with aligned visual embeddings (by averaging) improves performance, whereas for OTHER videos, relying solely on the transcript yields better results. Then, we cluster these embeddings to group segments that belong to the same topic to get the final segmentation. In addition to these embeddings, we leverage **transition cuts** to further improve clustering performance.

Type of video	Transcript	Visual (Image/OCR)	Transition
Slide	✓	✓	✓
Blackboard	✓	✓	–
Other	✓	–	–

Table 4.3 – Features used according to the type of lecture. Experiments on Yale Dataset show these combinations of features lead to the best performance.

To achieve this, we adapt the TW-FINCH hierarchical clustering algorithm (Sarfraz et al. 2021a), the same clustering method employed in AVLectures. TW-FINCH is a recursive clustering algorithm that constructs clusters by creating a graph based on distance relationships between segments. It identifies connected components in this graph as initial clusters, which are then iteratively merged by averaging their features and timestamps until the target number of clusters is reached.

Our modification primarily involves altering the way distances are calculated. In the original TW-FINCH algorithm, the distance metric is a product of the cosine distance between embeddings and the L1 distance between frame positions. In our version, we replace the frame-based distance with the **L1 distance between segment positions**. Furthermore, we incorporate additional distance specific to transitions.

To use transition information, we create a vector where each element corresponds to a segment, with the vector's length equal to the total number of segments. For each segment, we increment the value at its corresponding position by one if a transition occurs at the start of that segment. This encoding allows transitions to be represented as part of the segment distance calculation, enriching the clustering process with temporal and structural cues derived from the video content.

We obtain the formulation of the distance:

Let $E_{\text{modality},i}$ be the modality embedding of the segment i.

Let cluster(i) be the cluster after shot detection of the segment i and α the weight to control the importance of slide transitions.

$$\text{TotalDistance}(i,j) = \text{Distance}_{\text{embeddings}}(i,j) * \left(\text{Distance}_{\text{temporal}}(i,j) + \alpha * \text{Distance}_{\text{transitions}}(i,j)\right)$$

$$\text{Distance}_{\text{temporal}}(i,j) = |j-i|$$

$$\text{Distance}_{\text{embeddings}}(i,j) = \frac{E_{\text{aligned features},i} \cdot E_{\text{aligned features},j}}{\|E_{\text{aligned features},i}\| \|E_{\text{aligned features},j}\|}$$

$$\text{Distance}_{\text{transitions}}(i,j) = |\text{cluster(j)-cluster(i)}|$$

4.4.7 Summary

Aspect	Original AVLectures	Our Approach	Experiments
Clustering Algorithm	TW-FINCH with frame-based distance (cosine × L1 distance of segment positions)	TW-FINCH with segment-based distance (cosine × L1 distance of segment positions)	
Transition Information	Not incorporated	Incorporates transition information (slide changes)	Chapter 4.6.3
Alignment	Visual and transcript embeddings are **projected into a shared space** then concatenated to get final embeddings (dimension 4096)	**Transcript embeddings remain unprojected** to preserve robustness; visual embeddings are projected for alignment. Embeddings are averaged (final embeddings of dimension 768)	Table 10
Loss	Max Margin loss	InfoNCE loss (Oord et al. 2019)	Chapter 4.6.1
Datasets	Evaluated on AVLectures dataset	Evaluated on AVLectures, **VideoLecture Slide datasets and Yale datasets**	
Modality used	Utilizes image, OCR, transcript, and **video embeddings**	Utilizes image, OCR, and transcript embeddings	Chapter 4.6.2
Models Used	Pretrained encoders (BERT for text, ResNet/ResNeXt for visuals)	Same encoders with additional integration of DINO	

Table 4.4 – Summary of differences between the original AVLectures approach and our proposed method.

The Table 4.4 provides a summary of the changes we made over the original algorithm from (Gupta et al. 2023) to improve performance.

4.5 Experiments

We now present the details of our evaluation protocol.

4.5.1 Experiment Setup

For a given video, we output a hierarchical clustering of the segments. We can choose the number of clusters in TW-FINCH, as it will continue to merge close clusters until we reach the desired number of clusters. For fair comparison, we set the number of clusters to the ground truth like AVLectures.

The best version of our approach runs with the same parameters outlined in Chapter 4.4 and performs clustering using aligned multimodal embeddings (from OCR embeddings, image embeddings and transcripts embeddings) and transition cuts. More details and experiments on the configuration will be discussed later.

4.5.2 Evaluation Metrics

To ensure consistency and comparability, we adopt the evaluation metrics used by (Gupta et al. 2023), which include a comprehensive set of measures for clustering and segmentation

tasks. For clustering, we calculate Normalized Mutual Information (NMI), while segmentation performance is evaluated using F1-score, Intersection over Union (IoU), Mean over Frames (MoF), and Boundary Score @ k (BS@k). The BS@k metric quantifies the alignment between predicted boundaries and ground truth boundaries within a specified k-second interval, offering a precise measure of boundary localization accuracy. For fair comparison with (Gupta et al. 2023), we set $k = 30$. Predicted segment labels are sampled every second for these metrics.

4.5.3 Findings on Yale Dataset

We first conducted experiments with the original AVLectures algorithm on our largest dataset, the Yale Dataset, to assess the impact of different modalities on clustering performance. To begin, we tested clustering using each non-aligned modality independently, trying transcript embeddings, OCR embeddings, and image embeddings alone. Then, we experimented with using aligned embeddings that combine multiple modalities.

Method	**Feature modality**			**Metrics (%)**				
	visual	**textual**	**learned**	**NMI↑**	**BS@30↑**	**MOF↑**	**IOU↑**	**F1↑**
Naïve (Equal Splits)	-	-	-	70.5	27.7	71.2	**69.1**	57.1
Original transcript embeddings only	-	✓	-	**72.9**	**41.1**	**72.5**	67.6	**57.6**
Aligned transcript embeddings only	-	-	✓	71.9	40.6	71.4	66.5	56.9
Image embeddings only	✓	-	-	68.1	40.5	66.2	50.4	56.3
OCR embeddings only	✓	-	-	67.2	39.7	65.3	48.1	55.3
Multimodal	-	-	✓	71,6	40.7	70.9	65.7	55.4

Table 4.5 – Lecture video segmentation performance on Yale dataset. The best performance is obtained using original text embeddings.

Table 4.5 indicates that aligning embeddings does not produce the best results on this dataset. Using embeddings from the transcript alone proves to be the most effective approach. This outcome likely stems from the fact that many videos in the Yale Dataset make minimal use of slides or visual, as seen in our dataset composition analysis. The sparse presence of slides reduces the relevance of visual information, resulting in lower performance when aligning multimodal embeddings, as there is insufficient visual content to support meaningful alignment. Even when using only the aligned text embeddings, we observed a slight decline in clustering performance, suggesting that alignment doesn't improve performance.

Thus, aligning transcript embeddings does not necessarily improve performance, especially on non visual videos.

However, a deeper analysis of specific videos within this dataset reveals a different pattern for slide-based videos. In the Yale Dataset, certain lectures feature slides as the primary visual

support, with slides visible for over 50% of the lecture duration. We present the improvements for these slide-centric lectures in Table 9.

Lecture Name	Number of videos	Percentage of slides	Improvement on BS
Environmental Politics and Law with John Wargo	22	67%	13.45%
Evolution, Ecology and Behavior with Stephen C, Stearns	31	83%	4.09%
Foundations of Modern Social Theory with Iván Szelényi	20	82%	23.34%
Freshman Organic Chemistry II with Michael McBride	22	86%	14.14%
Freshman Organic Chemistry I with Michael McBride	32	87%	30.12%
Frontiers/Controversies in Astrophysics with Charles Bailyn	19	67%	−7.18%
Philosophy and the Science of Human Nature w/ Tamar Gendler	21	53%	5.11%
Roman Architecture with Diana E, E, Kleiner	23	96%	4.20%

Table 4.6 – Lecture video segmentation performance on videos from Yale Dataset showing slides most of time (>50% of time). Improvement on boundary score can be seen over using just original text embeddings.

In these cases, using aligned multimodal embeddings demonstrate clear advantages, showing that visual information and especially OCR, enhance segmentation performance when slides are present, informative and readable.

However, the performance on the lecture "Frontiers/Controversies in Astrophysics with Charles Bailyn" is noticeably lower. This is because our slide detection model misclassified handwritten text as slide content instead of blackboard material. Since the content is actually blackboard-based, the OCR struggles due to poorer image quality and the difficulty of recognizing handwriting. This, in turn, impacts the performance of our segmentation algorithm, as it relies heavily on accurate OCR outputs. These challenges are illustrated in Figure 46.

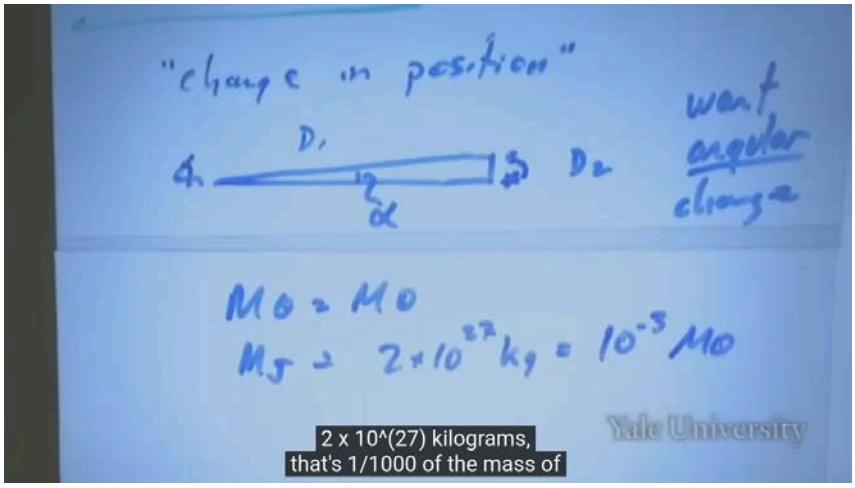


Figure 4.12 – Sample frame from the lecture "Frontiers/Controversies in Astrophysics with Charles Bailyn"

This finding aligns with our observations that lectures with a greater proportion of slides facilitate the effective use of multimodal features, especially OCR and other visual information. It suggests that multimodal embeddings are increasingly beneficial in lectures where slides serve as the primary medium, providing valuable cues for segmentation.

To optimize clustering performance across diverse lecture videos, it is essential to differentiate video types and tailor approaches according to the prevalence of slide-based content.

4.5.4 Baselines

Our main comparison is against AVLectures (Gupta et al. 2023), but we included the results of several baselines used in (Gupta et al. 2023) for completeness. These other baselines are:

- Naïve: The video is simply split in equal length based on a number of segments.
- Content-Aware Detector: Scene detection algorithm (Castellano 2018) is used to detect shot transition. It finds cuts by computing the difference in HSV values of consecutive frames. The resulting scenes are the segments.
- Latent Dirichlet Allocation (LDA) (Blei et al. 2003): a mixture of topics is extracted from the transcript of each individual video clip and are then hierarchically clustered using TW-FINCH (Sarfraz et al. 2021b).
- Continuous Temporal Embedding (CTE) (Kukleva et al. 2019): frame embeddings are first clustered using K-means and then grouped to form temporally consistent segments using Viterbi algorithm.
- Vanilla TW-FINCH: Features extracted from a video, without any consideration to produce a temporally coherent segmentation, are clustered with the clustering algorithm from (Sarfraz et al. 2021b).

- Large Language Model Zero Shot: We utilize the Qwen-2.5-32B (Team 2024) model to segment a text transcript into chapters. Due to hardware limitations preventing the entire transcript from fitting within the model's context window, we adopt a two-step process. First, we prompt the model to generate titles for 5-minute segments of the transcript, along with their corresponding timestamps. For each 5-minute segment, the model is provided with the relevant transcript section and the most contextually relevant titles.

> **Prompt**
>
> Here is the transcript of a video, with timestamps: {transcript}
>
> Analyze the following transcript chunk and identify any new topic changes, each element of the list should represent a section of the video. Give timestamps for each title. Here's the last titles found to have the context: {last_titles}. Here's an example of the expected JSON output: ["[hh:mm:dd] First section", "[hh:mm:dd] Second section"]. Only output the JSON.

Next, we use these generated titles to prompt the model to create the final chapters.

> **Prompt**
>
> Here is the titles of a video with timestamps: {titles}
>
> Group the titles into {number_chapters} chapters with timestamps. Here's an example of the expected JSON output: ["[hh:mm:dd] First chapter", "[hh:mm:dd] Second chapter"]. Only output the JSON.

The reported results on these baselines from previous work is summarized in Table 4.7.

4.5.5 Results on AVLectures Dataset

The results in Table 4.7 demonstrate that our approach outperforms all other methods, including the best-performing model from (Gupta et al. 2023). Our method achieves superior clustering quality and especially boundary detection, as evidenced by higher NMI and boundary scores at 30-second intervals. Additionally, our segmentation metrics surpass those of (Gupta et al. 2023).

Specifically, we achieve notable gains in NMI, boundary score, MoF, IoU, and F1-score, with improvements of +1.4%, +5.6%, +1.4%, +2.5%, and +1.9%, respectively. While both approache, with and without text projection, improve performance, the boundary score at 30 seconds is marginally higher without text projection. This indicates that preserving the original text embeddings supports more accurate boundary detection, likely because projection introduces distortions to the semantic space of the base model.

Method	Feature modality			Metrics (%)				
	visual	textual	learned	NMI↑	BS@30↑	MOF↑	IOU↑	F1↑
Naïve (Equal Splits)	-	-	-	71.8	32.5	75.5	62.7	74.0
Content-Aware Detector (Castellano 2018)	✓	-	-	72.9	57.0	73.3	59.4	65.9
LDA (Blei et al. 2003)	-	✓	-	70.0	38.8	72.4	57.6	68.2
CTE (Kukleva et al. 2019)	-	-	✓	67.2	41.5	67.3	48.1	57.3
LLM (Team 2024)	-	✓	-	73.0	47.6	73.5	59.5	68.7
	✓	-	-	71.6	46.9	71.3	56.5	66.4
Vanilla TW-FINCH (Sarfraz et al. 2021b)	-	✓	-	74.6	48.9	75.4	62.0	71.2
	✓	✓	-	74.9	52.1	75.1	61.7	70.9
SELF-SUPERVISED TRAINING								
(Gupta et al. 2023) (only pretrained)	-	-	✓	77.7	57.1	78.0	66.0	74.2
(Gupta et al. 2023) (only finetuned)	-	-	✓	78.5	57.2	79.0	67.2	75.3
(Gupta et al. 2023) (pretrained and finetuned)	-	-	✓	79.8	58.7	80.3	69.2	76.9
Ours (w/ text projection)	-	-	✓	80.5	63.2	**81.9**	71.3	**78.8**
Ours (w/o text projection)	-	-	✓	**81.2**	**64.3**	81.7	**71.7**	78.7

Table 4.7 – Lecture video segmentation performance on AVLectures evaluation dataset (350 videos). Our approach performs as well or better than previous state-of-the art

Table 4.8 presents the detailed performance of our approach and Table 4.9 presents the detailed performance of (Gupta et al. 2023) approach on the four different types of lecture videos from the dataset.

We obtain better results on each type and particularly good results on slide-based video (mixed, computer screen and slide types) mostly due to the use of transition information described in Chapter 4.4.1 and Chapter 4.4.2.

Type	Number of videos	NMI	BS@30	MOF	F1	IOU
Blackboard	157	**77.8**	**56.5**	**78.1**	**74.6**	**66.7**
Mixed	88	**82.0**	**63.0**	**82.7**	**80.4**	**73.4**
Computer screen	12	**85.3**	**83.2**	**90.3**	**88.9**	**84.0**
Slide	93	**85.7**	**76.3**	**85.6**	**82.8**	**76.8**

Table 4.8 – Results of our approach on different types of video.

Type	Number of videos	NMI	BS@30	MOF	F1	IOU
Blackboard	157	77.2	50.9	77.9	74.1	65.8
Mixed	88	80.4	59.0	80.9	78.1	70.4
Computer screen	12	83.4	75.6	86.0	83.2	77.6
Slide	93	83.3	69.4	83.0	79.6	72.7

Table 4.9 – Results of AVLectures algorithm on different types of video.

4.5.6 Results on VideoLectures and Yale Dataset

We display the results of our approach against AVLectures algorithm on the VideoLectures and the Yale Datasets in Table 4.10. The results demonstrate that our approach consistently outperforms AVLectures across all evaluated metrics on the VideoLectures Dataset. Notably, we improve the boundary score at 30 seconds, highlighting our approach's ability to effectively leverage the transitions of slide-based videos. The overall performance on this dataset is lower than on the AVLectures dataset, likely due to the removal of the aforementioned bias.

Our approach achieves better overall performance on the Yale Dataset compared to the AVLectures algorithm, primarily due to the use of multimodal embeddings for videos with detected slides or blackboards, and text embeddings for conference-style videos (labeled as other). However, as the Yale Dataset contains a relatively small proportion of slide-based content, the performance improvement, while noticeable, is not as pronounced. The high performance in MOF, F1, and IoU for the Naive method, which splits videos into equal-length chapters, highlights that chapters have relatively the same length within a video. However, this method is less accurate in boundary detection, as suggested by its lower BS@30 score.

These two different datasets highlight the strength of our approach when it comes to slide-based video while giving satisfactory results on non-slide video.

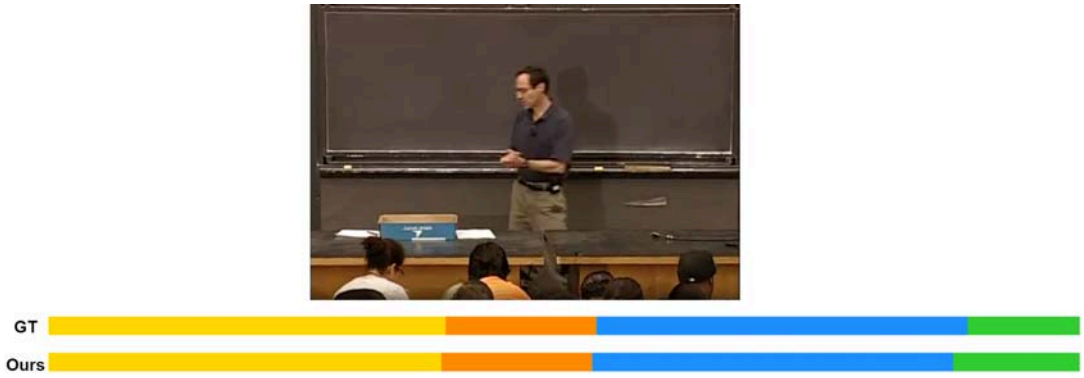
Dataset	Method	NMI	BS@30	MOF	F1	IOU
VideoLectures Slide Dataset	Naive	65.1	30.6	61.7	56.3	44.9
	AVLectures	70.0	43.2	70.0	57.5	48.6
	Ours	**74.6**	**56.2**	**73.5**	**64.3**	**56.0**
Yale Dataset	Naive	70.5	27.7	71.2	**69.1**	57.1
	AVLectures	71.3	39.1	70.6	65.1	54.9
	Ours	**73.2**	**41.7**	**72.8**	68.0	**58.0**

Table 4.10 – Lecture video segmentation performance on our new datasets. We improve performance by using transition information and separate video types.

4.5.7 Qualitative Results

We display a visualization of our predictions on three video lectures from AVLectures evaluation dataset in Figure 4.13. On the blackboard video (4.5), our system successfully identifies the segments but boundaries are imprecise. In contrast, predicted boundaries are very precise on the slide video (4.6) but the segments do not always perfectly match. Our algorithm also performs very well on the transparent screen video (4.7) but we argue that the boundaries on this one is too perfect sometimes. This highlights the main flaw of AVLectures dataset: the concatenation of short videos to make one long video with ground truth boundaries. Other results are displayed on Figure 4.14. We can see that our approach has some difficulties with videos containing a high number of segments.

We also visualize our approach's performance on another video outside of the dataset to demonstrate the use of hierarchical clustering as shown in Figure 4.15. This video from Youtube has manually annotated chapters but also contains a hierarchy of chapters. The use of hierarchical clustering means that we can group segments together by going higher in the hierarchy without any processing.



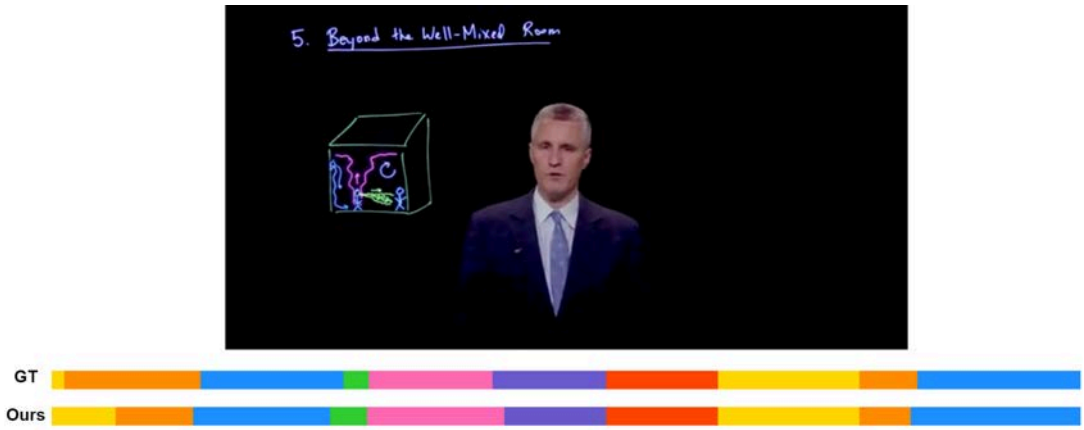
Single Variable Calculus

Max-min problems



Introduction to Probability

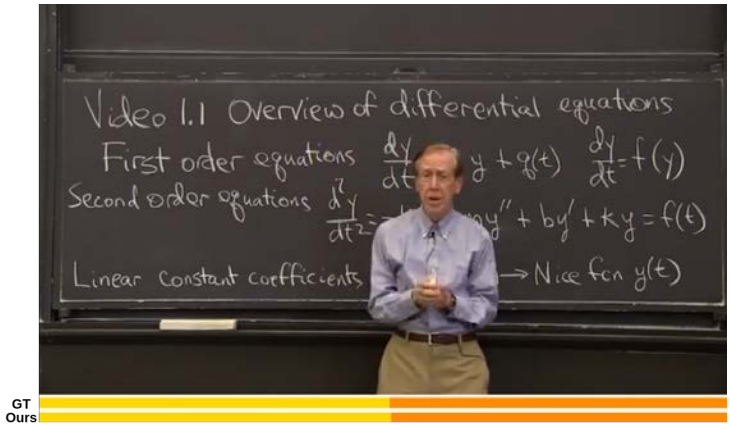
Independance



Physics of COVID-19 Transmission

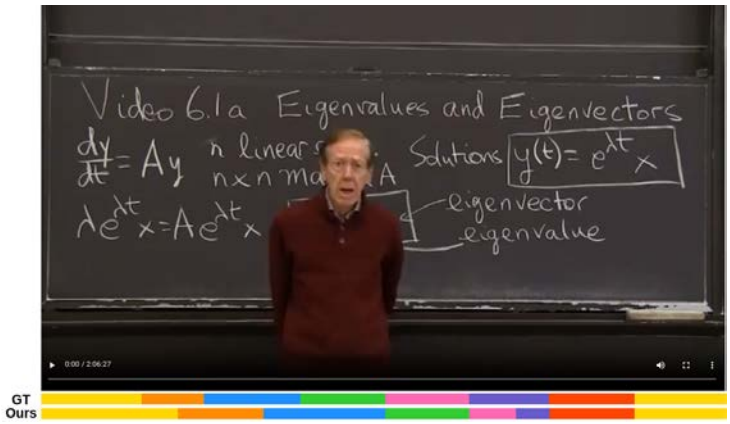
Beyond the well-mixed room

Figure 4.13 – Segmentation outputs of on three videos from AVLectures evaluation dataset. The result (a) is almost perfect compared to the ground truth. The result (b) fails on some segments but is still relatively close to ground truth. The result (c) is close to the ground truth and highlights the main flaw of AVLectures dataset: the bias of concatenating short videos to form a longer one with transitions, which is captured by our approach.



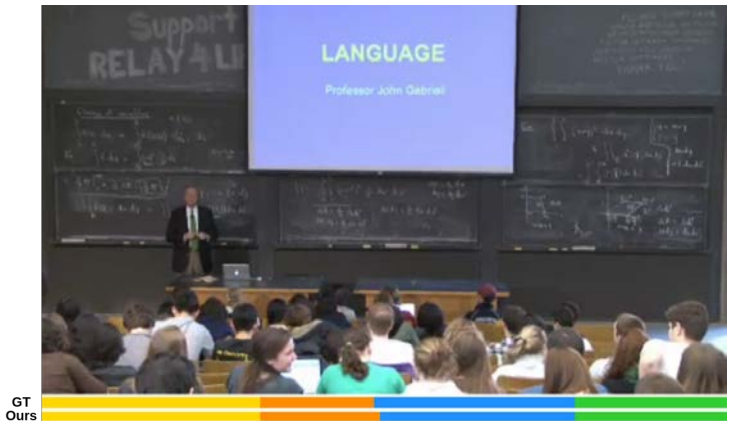
Learning differential equations

Overview of differential equations



Learning differential equations

Eigen values and Eigen vectors



Introduction to Psychology

Language

Figure 4.14 – Segmentation outputs on three other videos from AVLectures evaluation dataset. The result (a) and (b) are almost perfect compared to the ground truth. The result (c) fails on some segments.

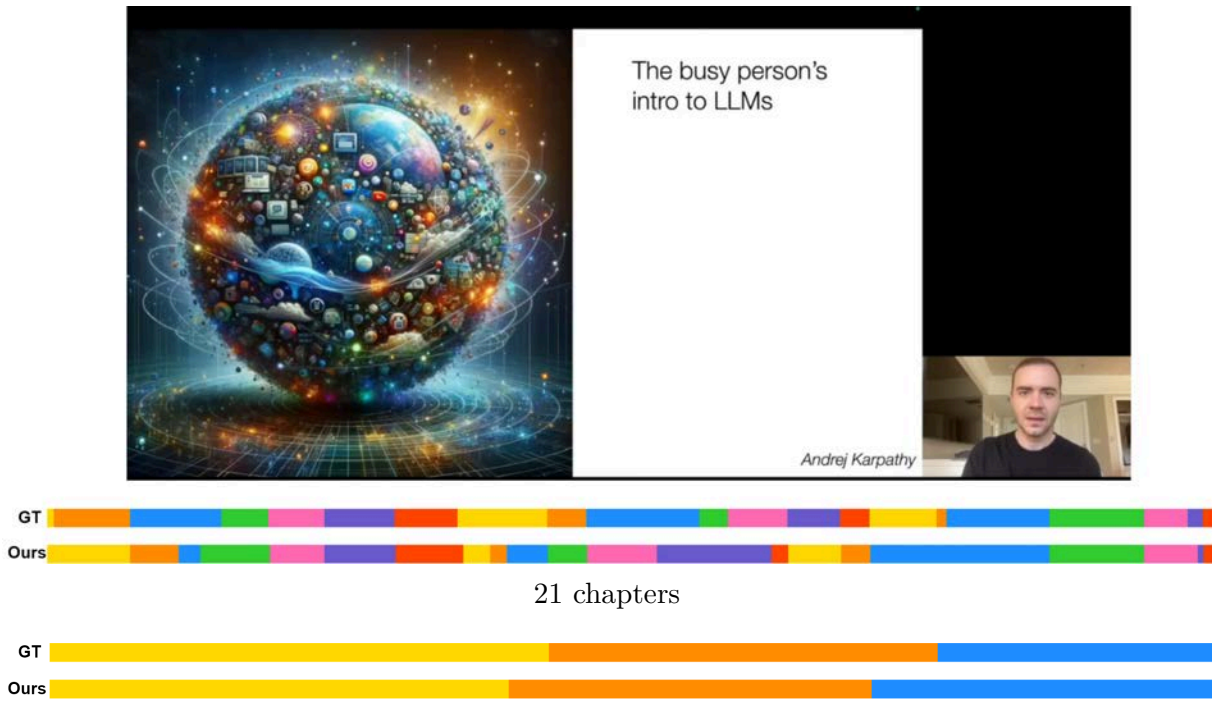


Figure 4.15 – Segmentation output of on a YouTube video outside AVLectures dataset. Ground truth is taken from the description of the YouTube video. The author of the video provides chapters but also group chapters in part. We demonstrate the use of hierarchical clustering to first identify the 21 chapters and then, by moving higher in the hierarchy, segments the content into the 3 broader parts.

4.6 Ablation Study

We conducted experiments to evaluate the impact of different parameters on our method. This section summarizes the key adjustments and their effects on segmentation performance.

4.6.1 Choice of contrastive loss

In this subsection, we evaluate the impact of different contrastive loss functions on the performance of our method. Specifically, we compare the original AVLectures max-margin loss with InfoNCE (Oord et al. 2019), a more commonly used contrastive loss. To ensure a fair comparison, we conduct 5 independent runs for each loss function and report the average performance along with the standard deviation.

Loss	NMI	BS@30	MoF	F1	IoU
InfoNCE	**80.95** ± 0.05	**63.25** ± 0.53	**81.61** ± 0.13	**78.52** ± 0.23	**71.23** ± 0.16
Max Margin	79.89 ± 0.36	61.35 ± 0.49	80.18 ± 0.31	76.84 ± 0.35	69.28 ± 0.40

Table 4.11 – Comparison of performance metrics between InfoNCE and Max Margin losses, showing averages and standard deviations over 5 runs.

Our results show that InfoNCE consistently outperforms max-margin loss, yielding better average performance and lower variability across runs. We attribute this to the design of InfoNCE, which normalizes over all negative samples and provides a smoother optimization landscape. In contrast, max-margin loss is less stable because it only considers a fixed margin threshold, making it more sensitive to outliers or variations in embeddings. This sensitivity likely introduces instability in the training process, leading to less consistent results.

Overall, these findings highlight the importance of the choice of contrastive loss, with InfoNCE proving to be a more robust option for our method.

4.6.2 Use of modalities in the clustering process

Visual features are used in our system to segment the video. In this experiment, we investigate the impact of each feature on performance.

OCR	Image	Video	NMI	BS@30	MOF	F1	IOU
✓	-	-	80.7	62.2	81.5	78.2	71.0
-	✓	-	79.4	59.9	79.8	76.5	68.6
-	-	✓	77.0	55.0	77.7	74.2	65.6
✓	-	✓	79.1	58.4	79.7	76.6	68.6
✓	✓	-	**81.2**	**64.3**	**81.7**	**78.7**	**71.7**
✓	✓	✓	81.1	63.6	81.7	78.6	71.2

Table 4.12 – Impact of the visual modalities during the clustering process on AVLectures dataset

Table 4.12 presents the results on visual features by using different combinations of OCR, image, and video embeddings during the clustering process. The results show that OCR embeddings consistently achieve the best individual performance, as they carry meaningful textual information directly extracted from the slides. Image embeddings, while slightly less effective than OCR, still produce strong results, demonstrating their value in capturing visual content. In contrast, video embeddings perform the worst, as they do not provide sufficient semantic information for clustering, likely due to the lack of direct relevance to topic boundaries in lecture videos. The best overall performance is achieved by combining OCR and image embeddings, highlighting the complementary nature of these modalities in capturing both textual and visual information effectively.

4.6.3 Use of transition information during clustering

> ***i*** **Reminder**
>
> $$\text{TotalDistance}(i,j) = \text{Distance}_{\text{embeddings}}(i,j) * \left(\text{Distance}_{\text{temporal}}(i,j) + \alpha * \text{Distance}_{\text{transitions}}(i,j)\right)$$

We use transition information during clustering only for slide-based videos, where transitions can indicate topic changes. By adjusting the parameter $\boldsymbol{\alpha}$, which weights the importance of transition information, we analyze its influence on the clustering process. The tables below compare results with different alpha values to demonstrate the impact of incorporating transition information on the different slide datasets.

$\boldsymbol{\alpha}$	NMI	BS@30	MOF	F1	IOU
0	83.0	71.6	83.0	79.9	73.0
1	84.6	74.1	84.7	81.6	75.2
2	**85.9**	76.3	85.6	82.6	76.6
3	85.7	76.3	85.6	82.8	76.8
4	85.4	**76.6**	**85.7**	**82.9**	**76.9**
5	85.2	76.1	85.4	82.4	76.3

Table 4.13 – Impact of transition information on clustering performance for slide-based videos with varying alpha values on **AVLectures Dataset**

The results demonstrate that incorporating transition information improves clustering performance across all metrics. As α increases from 0 to 2, there is a consistent improvement in metrics like NMI, BS@30, MOF, F1, and IOU, with the best results observed at $\alpha = 2$. Further increasing α to 4 shows marginal gains or stabilization, suggesting diminishing returns beyond this point.

$\boldsymbol{\alpha}$	NMI	BS@30	MOF	F1	IOU
0	70.6	46.6	68.9	60.5	51.2
1	70.8	48.5	70.3	61.0	52.0
2	72.5	51.1	70.4	60.9	52.2
3	74.0	54.9	72.4	62.8	54.5
4	74.6	56.2	73.5	**64.3**	**56.0**
5	**74.8**	**56.5**	**73.6**	64.0	**56.0**

Table 4.14 – Impact of transition information on clustering performance for slide-based videos with varying alpha values on **VideoLectures Slide Dataset**

The results on the VideoLecture Slide dataset show a more substantial improvement in clustering performance compared to the AVLectures slide dataset. This was expected, as the VideoLecture Slide dataset exclusively features videos with only slides (no talking heads or other distractions), making transitions more reliable indicators of topic changes. In such videos, a slide transition often directly corresponds to a shift in topic or concept, enhancing the utility of transition information for segmentation.

The improvement across metrics—particularly at $\alpha = 4$-highlights the effectiveness of leveraging transition cues in this context. The clear structure of slide-based videos in this dataset

allows the model to better align transitions with topic boundaries, resulting in more accurate clustering and segmentation.

These results highlight the effectiveness of using transition information, particularly in datasets with cleaner and more structured transitions, where it provides a significant boost to segmentation performance.

4.7 Discussion

The experimental results demonstrate that our approach achieves comparable or superior performance to state-of-the art lecture video segmentation by optimizing the use of multimodal information and using a more stable loss. It also outperforms other unsupervised approaches.

The effectiveness of our approach can be attributed to the analysis and use of the inherent structure of lecture videos, especially on slide-based video for which slide changes are good indicators of the lecture structure. This helps to significantly improve boundary detection.

However, on non-slide based video a clear structure can be hard to identify. There often is no obvious visual cues (like slide changes in slide videos) that indicate a segment change.

We also observed that our method performs better on blackboard when OCR embeddings are used, as seen in the AVLectures dataset. However, the effectiveness of OCR can be limited by the challenges of recognizing handwritten text and the low quality of some datasets (like Yale blackboard videos), which hinder the extraction of high-quality embeddings.

Another key point is that the ground truth annotations of the segmentation don't hold the absolute truth, especially regarding the number of segments. In educational videos, different annotators might have varying perceptions about the appropriate granularity for dividing a lecture into segments. This variability poses a challenge: the subjective nature of segmentation means that there is not always a 'correct' number of segments into which a video should be divided. For instance, one expert might consider a long explanation of a concept as a single segment, whereas another might see it as several smaller segments, each focusing on different facets of the concept. In response to this challenge, we show that hierarchical clustering in Figure 4.15 allows to dynamically adjust the granularity of the segmentation.

4.8 Conclusion and Future Work

Recent advances in foundational language and visual models have made it easier to extract meaningful feature embeddings from multimedia content. However, achieving strong performance still depends on using these features wisely. In this paper, we revisit lecture video segmentation and demonstrate that clustering lightly aligned multimodal embeddings, when tailored to the type of lecture, can exceed the performance of state-of-the-art methods that require extensive training.

Our findings show the value of adapting features to different lecture types. For example, slide-based lectures benefit from slide transition features while a conference without support does not need visual features. Additionally, our method naturally supports hierarchical segmentation, enabling the clustering of topics at different granularities.

There are several possible extensions to our approach. As we perform hierarchical clustering, it is feasible to perform hierarchical segmentation to cluster the segments by different granularity of topics. The main avenues for improvement is to specifically target blackboard videos. Detecting handwriting text and tracking it over time could help identify sequences when the lecturer erases the board, or changes board, which would be good candidate transitions between segments. Better automated handwritten recognition models would also help, as the extracted text could be added to the transcript in a similar way as we do for slide segments. We also plan to investigate text structure extraction techniques in order to guide segment generation with the transcript alone.

Summary

In this chapter, we propose a method for temporally segmenting lecture videos without relying on supervised training or large datasets of pre-segmented videos. Our approach combines text and image embeddings extracted from lecture content using foundational visual and language models with features like slide transitions to improve segmentation. By adapting the use of these embeddings and features to the type of lecture, we achieve better performance with a simple alignment process than previous state-of-the-art methods. Our approach works effectively across various lecture formats, including in-classroom recordings with blackboards or slides and screen-recorded presentations, as demonstrated through evaluation on new datasets.

Chapter 5

Lecture Video Analysis Toolkit

This chapter presents a proof-of-concept implementation developed to explore the practical integration of the methods and ideas discussed throughout this thesis. While the application demonstrates a functioning pipeline and user interface, it is not intended as a production-ready tool, but rather as a research prototype aimed at validating the feasibility and potential of multimodal lecture video analysis. Apart from early exploratory work focused on accessibility —particularly with neurodivergent learners—no formal user studies or large-scale evaluations have yet been conducted. As such, the focus of this chapter is not on comprehensive assessment, but on showcasing how previously developed components can be combined into an interactive system that supports multimodal navigation, content linking, and accessibility-aware design of slide-based lecture videos..

A demonstration of the application is available at https://travis-seng.fr/lecturetoolkit.

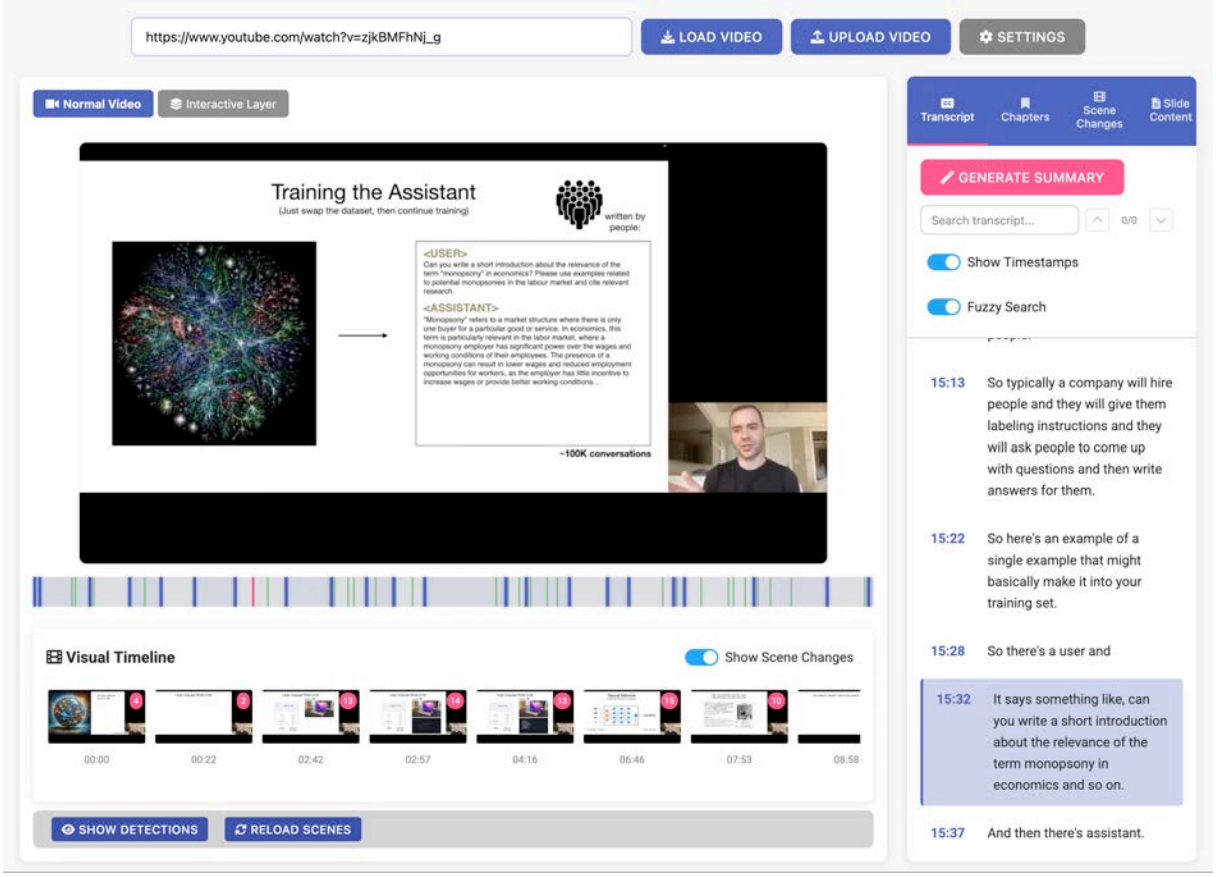


Figure 5.1 – Interface of our application

5.1 Introduction

As reviewed in Chapter 2, multiple systems have been proposed to enhance the accessibility and navigability of lecture videos. Some focus on indexation (Adcock et al. 2010a, Yoo et al. 2021) or summarization (Biswas et al. 2015); others provide navigation aids or aim to improve accessibility through new interfaces (Zhao et al. 2019a, Sechayk et al. 2024), retargeting (Kim et

al. 2022a, Weng et al. 2022). While these systems offer valuable contributions, most are closed-source and difficult to evaluate, reproduce, or adapt in educational and research contexts.

To address this gap, we present a proof-of-concept lecture video analysis toolkit that integrates key capabilities from prior work—such as scene detection, transcript generation, OCR, and cross-modal content linking—into a unified and openly accessible application for slide-based lecture videos. Unlike many existing systems, this toolkit is designed to be extensible, reproducible, and adaptable to diverse use cases. It builds directly on the components and findings developed in Chapter 3 and Chapter 4.

What distinguishes this application is not only the combination of these multimodal processing components into a coherent workflow, but also the emphasis on interactivity and accessibility. Drawing from early feedback from neurodivergent learners, we incorporated features that support customizable interfaces, predictable interaction patterns, and alternative ways of engaging with dense video content. Users expressed the need for clearer structure, the ability to simplify or reorganize visual layouts, and more control over how information is presented. These insights have directly informed our interface design and interaction model.

The remainder of this chapter first outlines the key design considerations that guided the development of the application, followed by a detailed presentation of the system architecture, the core processing modules, and the interactive interface. The chapter concludes with a discussion of potential use cases, with particular emphasis on inclusive education and accessibility-driven scenarios.

5.2 Design considerations

The design of the application was shaped by a combination of technical priorities, a synthesis of existing work, and early-stage feedback from neurodivergent users. Our goal was to create a modular and open application that not only reimplements successful ideas from prior work, but also serves as a flexible basis for exploring new accessibility- and user-centered directions.

From summarization and indexing tools such as TalkMiner (Adcock et al. 2010a), MMToC (Biswas et al. 2015), and VideoKen (Mahapatra et al. 2018), we adopted the principle of extracting slide content and organizing it for structured navigation. Rather than relying on external metadata or platform-provided cues like SlideGestalt (Peng et al. 2023), we focus on direct scene detection and OCR applied to the video stream itself, enabling independence from underlying presentation tools. Inspired by (Kim et al. 2022a) and (Sechayk et al. 2024), which attempt to bridge visual and spoken modalities, we implement semantic alignment between transcript and visual text using embeddings. This allows the system to infer connections that are not based solely on string similarity or timing overlap, but instead on broader contextual meaning.

In terms of user experience, systems like the work from (Zhao et al. 2019a), HiVideo (Weng et al. 2022), FitVid (Kim et al. 2022a) have demonstrated the benefits of multi-layered interaction and adaptive layouts. Our application emphasizes simplicity and modularity by offering three main interaction paths: through the spoken transcript, the extracted slide sequence (as scene thumbnails), and the OCR-derived text zones. These modalities are cross-linked so that actions in one—such as clicking a line of transcript—highlight and navigate to corresponding content in the others.

Accessibility played a central role in our design. Prior works such as Slidecho (Peng et al. 2021d) and Say It All (Peng et al. 2021b) have shown how enriching video content with additional layers—like slide descriptions or spoken-word alignment—can significantly improve usability for visually impaired or neurodivergent learners. While our application does not yet implement audio feedback or assistive narration, it was developed with the intention of making as much structured information available as possible. By extracting transcripts, OCR text, scene transitions, and visual regions, we aim to provide as much information that could support future accessibility features—such as simplified views, personalized layouts, or non-visual navigation modes.

A first version of the application was showcased to neurodivergent users by psychologists for the CERPPS (Centre d'études et de recherches en psychopathologie et santé), in a joined collaboration with Atypie-Friendly. Their feedback further shaped several key interface principles. The ability to search both spoken and visual content was considered especially helpful, as was the slide overview, which provides a predictable overview of the presentation. Users highlighted that connecting visual and verbal elements through simple, visible links could improve comprehension. Some also suggested that the ability to modify or annotate content—such as correcting visual typos, hiding distracting elements, or transforming the layout—could help reduce visual clutter and cognitive load. Maintaining a consistent visual template across the interface was also cited as an important factor in user comfort and orientation. Hiding the presenter's face, for example, was seen as a desirable personalization option for certain users.

These insights guided the decision to design an interface that is predictable, content-rich, and adaptable. Rather than replicating any single prior system, our application is intended as an exploratory tool—one that integrates well-established techniques while creating space for further experimentation with personalization and accessibility.

5.3 System Architecture

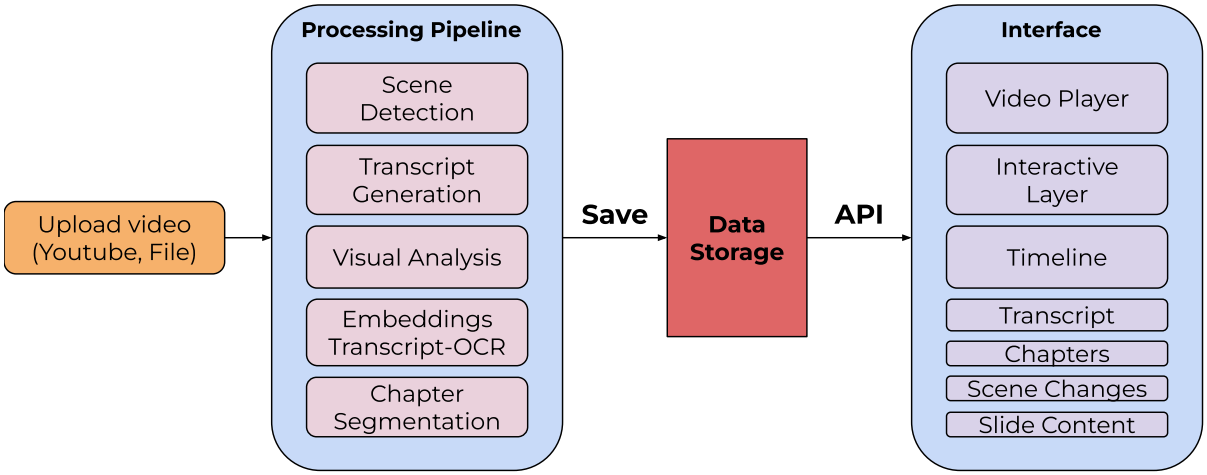


Figure 5.2 – System architecture of our application designed around three components. The processing pipeline extract multimodal information from the video then stores it in a file-based storage. A web interface queries data from the storage to display interactive information.

The application is structured around three main components: a backend responsible for video processing, a frontend interface for user interaction, and a file-based storage system for managing outputs. The architecture is designed to support modular, fast, and reproducible analysis of lecture videos by combining multiple processing steps into a coherent pipeline. A diagram of the architecture is shown in Figure 5.2.

High-Level Overview

At a conceptual level, the system takes a video input—either uploaded by the user or downloaded from YouTube—and passes it through a sequence of processing modules. These include scene detection, transcript generation, visual content extraction, OCR, and semantic linking between spoken and visual elements. The backend coordinates this workflow and exposes API endpoints to trigger processing tasks, retrieve results, and provide progress updates. The frontend allows users to monitor processing in real time and explore the output through an interactive interface that includes a transcript viewer, scene browser, visual timeline and visual overlays as seen in Figure 5.1. Intermediate and final outputs are stored locally using a structured directory layout based on a unique identifier for each video.

Backend API

The backend is built using FastAPI, which provides a lightweight and efficient framework for serving asynchronous operations. Upon receiving a new video, the backend computes a content hash to uniquely identify the file. This identifier is used to manage storage and avoid redundant processing. The backend then schedules the various processing steps, including downloading

or receiving the video, detecting scene transitions, generating transcripts, detecting visual elements, running OCR, and generating semantic embeddings. Status updates are streamed to the frontend using Server-Sent Events (SSE), allowing users to track progress in real time and begin interacting with results as they become available.

Performance Considerations

Although the focus of the application is not real-time processing, its architecture allows for fast execution of large-scale analysis. By combining asynchronous task management, background workers, and concurrency mechanisms, the system is able to process a one-hour video in approximately one minute on a laptop with a RTX 4070.

Data Storage

Processed data is stored using a simple file-based structure. Each video is assigned a hash that is used to organize related files—including scene metadata, image thumbnails, transcripts, OCR results, and embeddings—into named subdirectories. This approach avoids external dependencies and makes intermediate results easy to inspect, share, or reuse.

5.4 Core Modules

The application's functionality is structured around a different modules that each operate on a specific modality of the input lecture video. These modules were developed using components presented in earlier chapters, and they are orchestrated to function together in a streamlined and automated pipeline. The following section describes the main stages of processing: from video ingestion and segmentation, to speech transcription, visual content analysis, optical character recognition, and cross-modal linking using semantic embeddings.

Video Ingestion and Preparation

The system accepts input in two forms: direct video uploads from users, or lecture videos hosted on YouTube. In the case of YouTube content, the application uses the yt-dlp utility to download the video in a predefined format, optimizing for resolution and processing speed. For uploaded videos, the system computes a content-based hash of the file, which is used to uniquely identify and organize all processing outputs. This approach ensures consistency across modules and enables deduplication, avoiding unnecessary reprocessing of the same content.

Once downloaded or uploaded, the video is registered within the processing pipeline. The raw video file is stored locally.

Scene Detection

Scene detection is used to segment the lecture video into visually distinct intervals, typically aligned with slide transitions or changes in presentation structure. We reuse the same system as in Chapter 4.

For each detected scene, the system extracts two types of visual representations. A thumbnail image is generated for use in the frontend interface, allowing users to browse the lecture by slide. A full-resolution frame is also extracted and saved for downstream tasks such as object detection and OCR. These images serve as the visual backbone of the application and provide the primary units of analysis for slide content.

Transcript Generation

Speech transcripts are generated using two possible sources: the YouTube transcript API, or OpenAI's Whisper model (Radford et al. 2023b). When automatic captions are available on YouTube, the application fetches them directly, offering a fast and relatively accurate option. In cases where transcripts are missing or inadequate, the system falls back on Whisper to produce a high-quality, segment-level transcript of the spoken content. The resulting transcript is saved in a structured format and used both for direct display to the user and for linking with visual content via embeddings.

Visual Content Analysis

After scenes have been identified and high-resolution images extracted, each image is analyzed to detect regions of interest using the trained object detection model from Chapter 3. This model is capable of identifying slide elements such as titles, paragraph text, captions, diagrams, images and more.

For each scene image, the system passes it through the detection model and records bounding boxes, class labels, and confidence scores. These detections serve two purposes: they provide a visual map of the slide layout and define regions detected by text. Detected regions flagged as containing text are forwarded to the OCR module.

Optical Character Recognition (OCR)

Text regions detected in the previous step are passed through one of two OCR engines, depending on system configuration and user preferences. The default engine is Tesseract (Kay 2007), which is applied to each cropped text region individually. The application can alternatively use Surya (Paruchuri 2025), an open-source deep learning-based OCR system that processes entire images and offers improved performance on complex or degraded visual inputs, but at lower processing speed.

OCR results are matched back to the original scene image and aligned with the bounding boxes generated by the detection module. This step enriches the visual layer of the video with actual textual content, enabling search and cross-modal linking.

Multimodal Embedding and Linking

Once both the transcript and the OCR outputs are available, the application computes semantic relationships between these two modalities. This is done using sentence-level embeddings computed with **all-mpnet-base-v2** (Song et al. 2020), which are compared using cosine similarity to match spoken and visual elements that refer to the same concepts or topics, within the same slide segment. A similarity score of 0.4 or higher is used as the threshold to confirm a match. This value was selected after experimenting with different thresholds and observing that it provided a good balance: it successfully identified relevant connections while minimizing false positives and avoiding matches between unrelated content.

The embedding module generates bidirectional links: spoken sentences that relate to specific slide text, and slide text that can be contextualized by corresponding segments of speech. These links are used to create interactive connections in the frontend and enable semantic navigation across the video. Those links can be seen in Chapter 5.5.

Chapter Summarization

The application includes a chapter summarization feature designed to support users in understanding and navigating the structure of lecture videos. This functionality builds on the segmentation algorithm introduced in Chapter 4, which groups transcript segments into coherent chapters using visual and linguistic cues. To generate a concise title or summary for each chapter, we use a large language model—specifically Gemini Flash 2.0 (DeepMind et al. 2024)—which processes the grouped transcript segments and outputs a description of the content or a title.

Beyond this integration, we are also experimenting with prompt-based interactions using Gemini to explore alternative segmentation strategies. When prompted appropriately, the model can perform its own content-based segmentation of the transcript, identifying meaningful boundaries and generating summaries without relying on the initial algorithm. Here is the prompt that we used:

Prompt

Based on the following video transcript with timestamps, create chapters that outline the main topics or sections. For each chapter, provide:

1. The timestamp where the chapter starts (in MM:SS format)
2. A title

Format each chapter exactly like this example:

[{{“timestamp”: “00:00”, “description”: “Introduction to Topic”}},

{{“timestamp”: “02:30”, “description”: “Key Concept Explained”}},

{{“timestamp”: “05:45”, “description”: “Practical Examples”}}]

Transcript: {full_text}

This dual approach allows us to compare rule-based and generative methods for structuring long-form video content.

This summarization feature is especially motivated by early feedback from neurodivergent users, who emphasized the value of having clearer overviews and simplified entry points into complex material.

5.5 User Interface

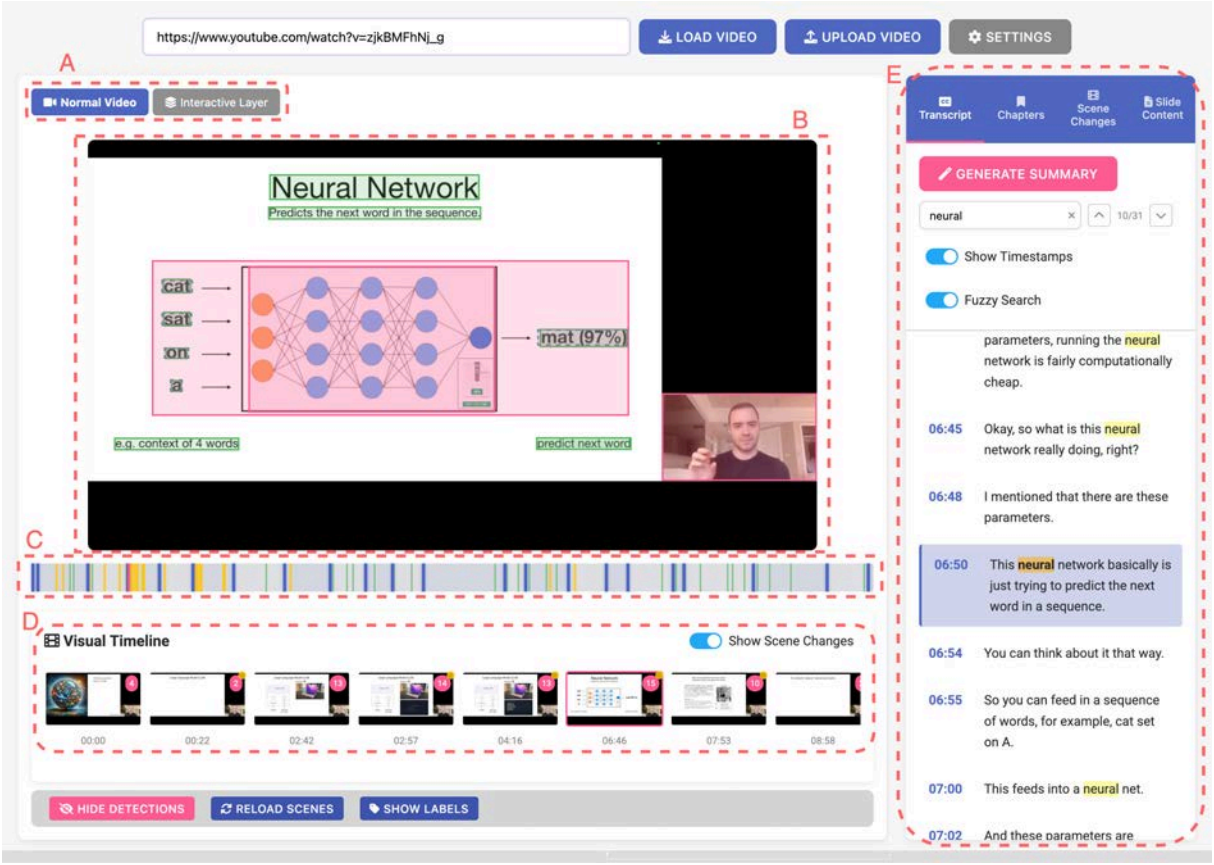


Figure 5.3 – Interface of our application. (A) allows to user to turn on the interactive layer to move and resize elements in the video player (B). (C) represents the events timeline (slide transitions are marked in green, chapters are marked in blue) and search results in yellow. (D) represents a visual timeline with thumbnails for quick overview. (E) shows the transcript, chapters, scene changes and slide OCR content.

The frontend interface of the application displayed in Figure 5.3 is designed to present video analysis results in a clear, structured, and interactive format. It allows users to explore lecture content from different perspectives, including speech transcripts, slide images, chapter segmentation, and visual detections.

At the top of the interface, users can either upload a video file or paste a YouTube link. Once submitted, the system begins processing the video, and users can monitor progress in real time. Status updates for each stage—such as downloading, scene detection, transcription, and OCR —are streamed to the frontend.

Once processing is complete, the main interface is divided into several coordinated panels. A central video player displays the lecture video with optional overlays, such as slide regions or OCR-detected text. Directly below the player, an event timeline summarizes the structure

of the lecture: green marks slide transition, blue marks chapter transitions derived from our chapter summarization and yellow marks search results.

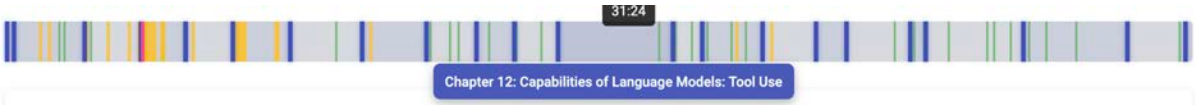


Figure 5.4 – The event timeline contains chapter transitions in blue and slide transitions in green. If there is a search query, the timeline displays the occurences in yellow.

Below the event timeline is a visual timeline with scene thumbnails displayed in Figure 5.5. The thumbnails provide a linear preview of the detected slide transitions. Each thumbnail corresponds to a detected scene and can be clicked to jump to that point in the video. These frames are extracted using the scene detection module described earlier and serve as visual anchors throughout the interface.

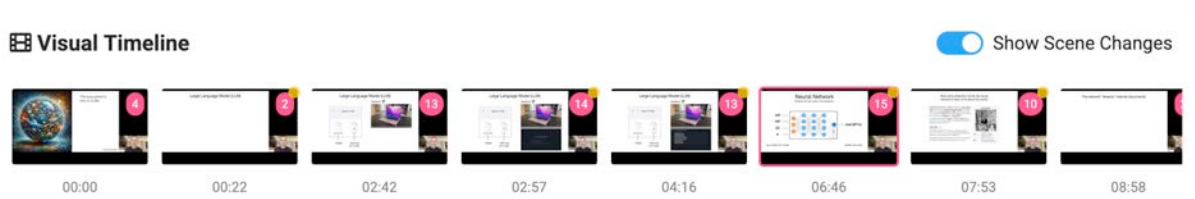


Figure 5.5 – The visual timeline displays extracted slides as thumbnails with timestamps for a quick overview.

On the right-hand side of the interface, a vertical tab panel provides access to multiple synchronized content views. The Chapters tab presents a chronologically ordered list of chapter titles and timestamps, generated through the segmentation and summarization modules. Clicking on a chapter title immediately seeks the video to the corresponding section, offering a structured overview of the lecture. The Transcript tab displays the full speech transcript, segmented by time, with each line linked to its exact position in the video. The Scene Changes tab provides a timestamped list of detected slide transitions, while the Slide Content tab shows text extracted from each slide via OCR. Interacting with any element—whether a transcript sentence or a visual text segment—automatically seeks the video to the relevant moment. To further support efficient exploration, the interface includes a search function that allows users to locate keywords across both the transcript and the OCR-extracted slide text. This functionality provides a fast way to jump to specific content, particularly useful in longer videos. An overview of this content navigation panel is shown in Figure 5.6.

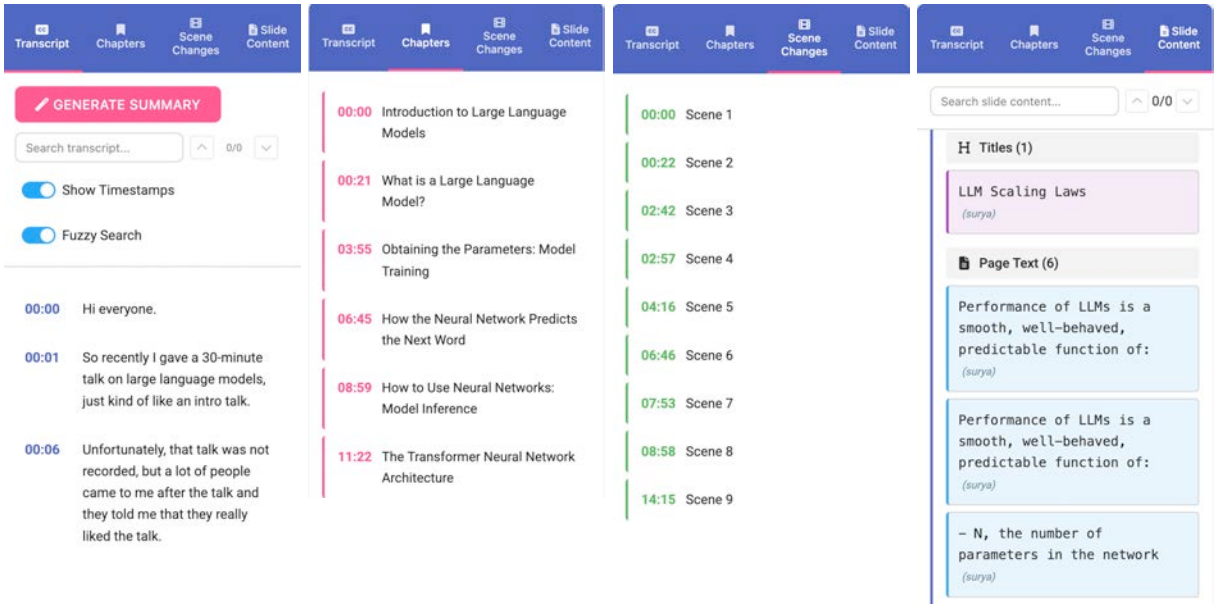


Figure 5.6 – The interface contains a menu to display information. From left to right: the transcript menu, chapters menu, scene changes menu and slide content menu.

A "Show Detections" button allows the user to display object detection results. The result can be seen in Figure 5.7.



Figure 5.7 – The interface can display object detection results directly on the video player.

Interactive Layer

In addition to the standard playback view, the application offers an Interactive Layer mode that enables direct manipulation of visual elements on top of the video. When this mode is

activated, the user can move, resize, or reorganize any individual components—such as text regions detected on slides or images—allowing for a more customizable viewing experience. The elements stay animated such as the talking head, animations or videos. This flexibility supports various user needs, from focusing attention on specific content to adapting the layout for different screen sizes or cognitive preferences. The user can choose to remove elements or the presenter and zoom on any element. An example is shown in Figure 5.8.

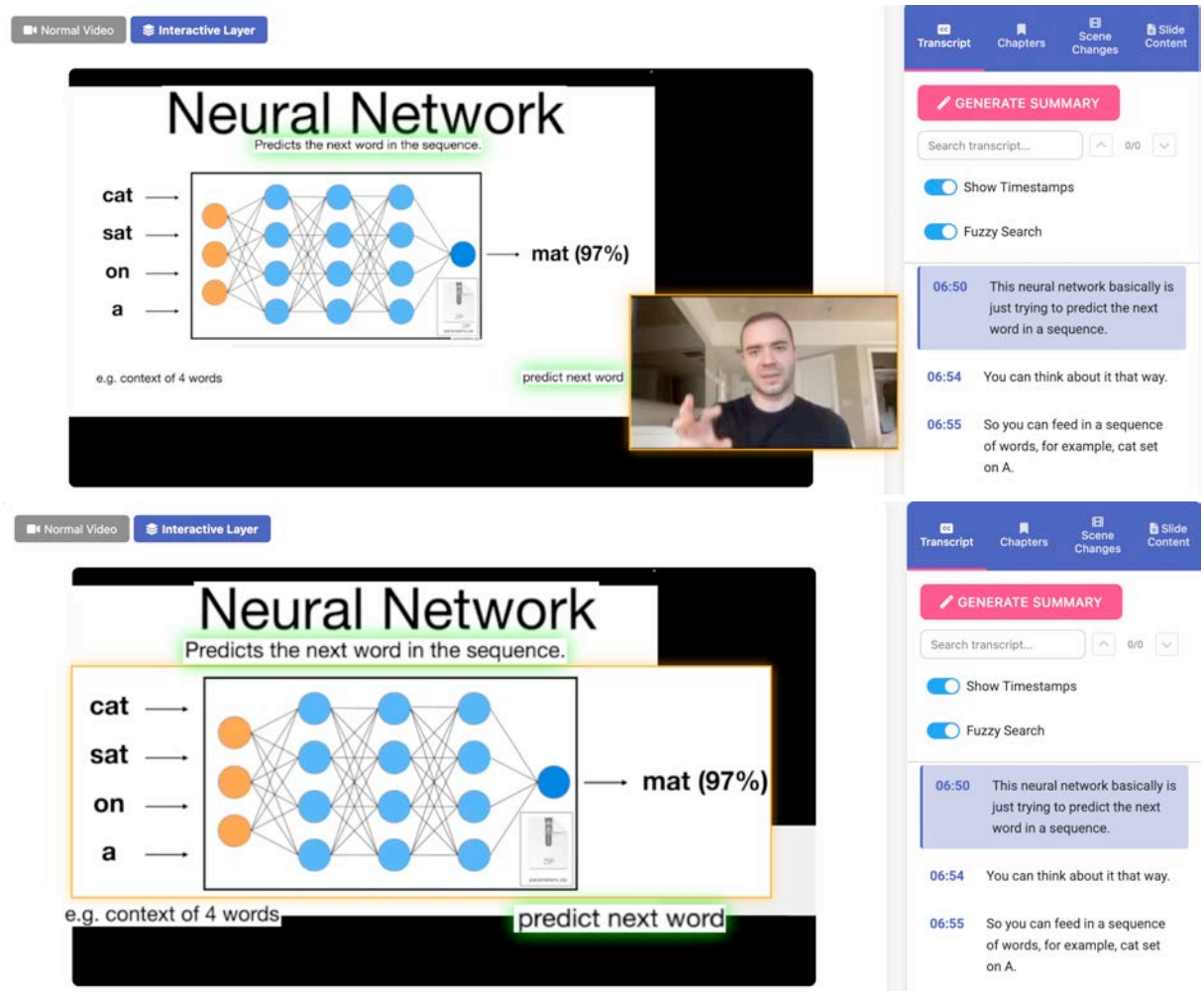


Figure 5.8 – Two versions of a video at the same timestamp. The interactive layer allows user to customize elements size and position. They can also remove elements such as the presenter.

As shown in Figure 5.9, the interactive layer also visualizes the semantic links established between spoken and visual content thanks to the multimodal embeddings linking detailled in Chapter 5.4. When a transcript segment is played or selected, related text regions on the slide are highlighted with a green glow, helping users quickly identify connections between what is being said and what is being shown. The same applies when the user clicks on a slide text element. The relevant transcript will be shown with a green glow as displayed in Figure 5.10.

This bidirectional interaction reinforces comprehension by making abstract or implied relationships explicit through visual feedback.

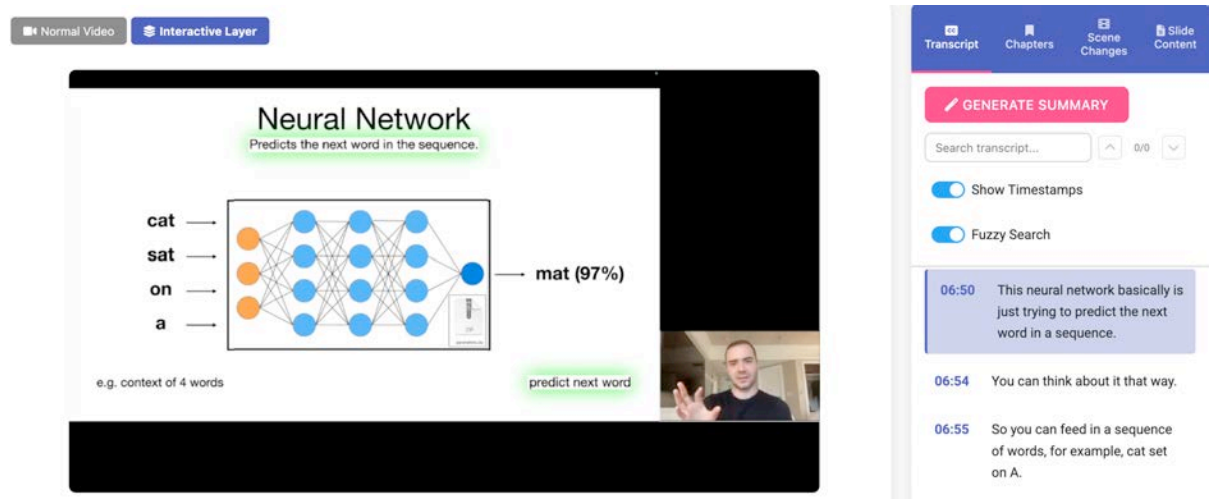


Figure 5.9 – When a transcript segment is played, related text regions on the slide are highlighted. Here, we use a green glow.

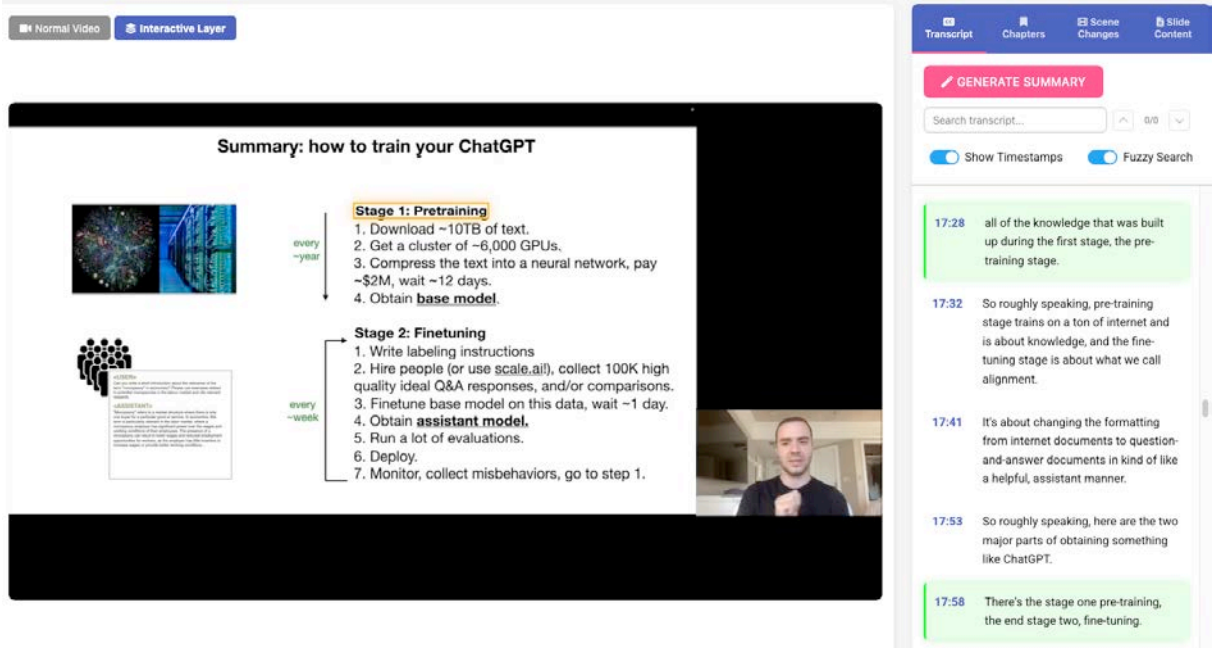


Figure 5.10 – When the user clicks on an OCR text on the slide, the related transcript segments are highlighted in green.

The interactive layer also enables the direct manipulation of the text thanks to OCR and allows automatic translation as highlighted in Figure 5.11.

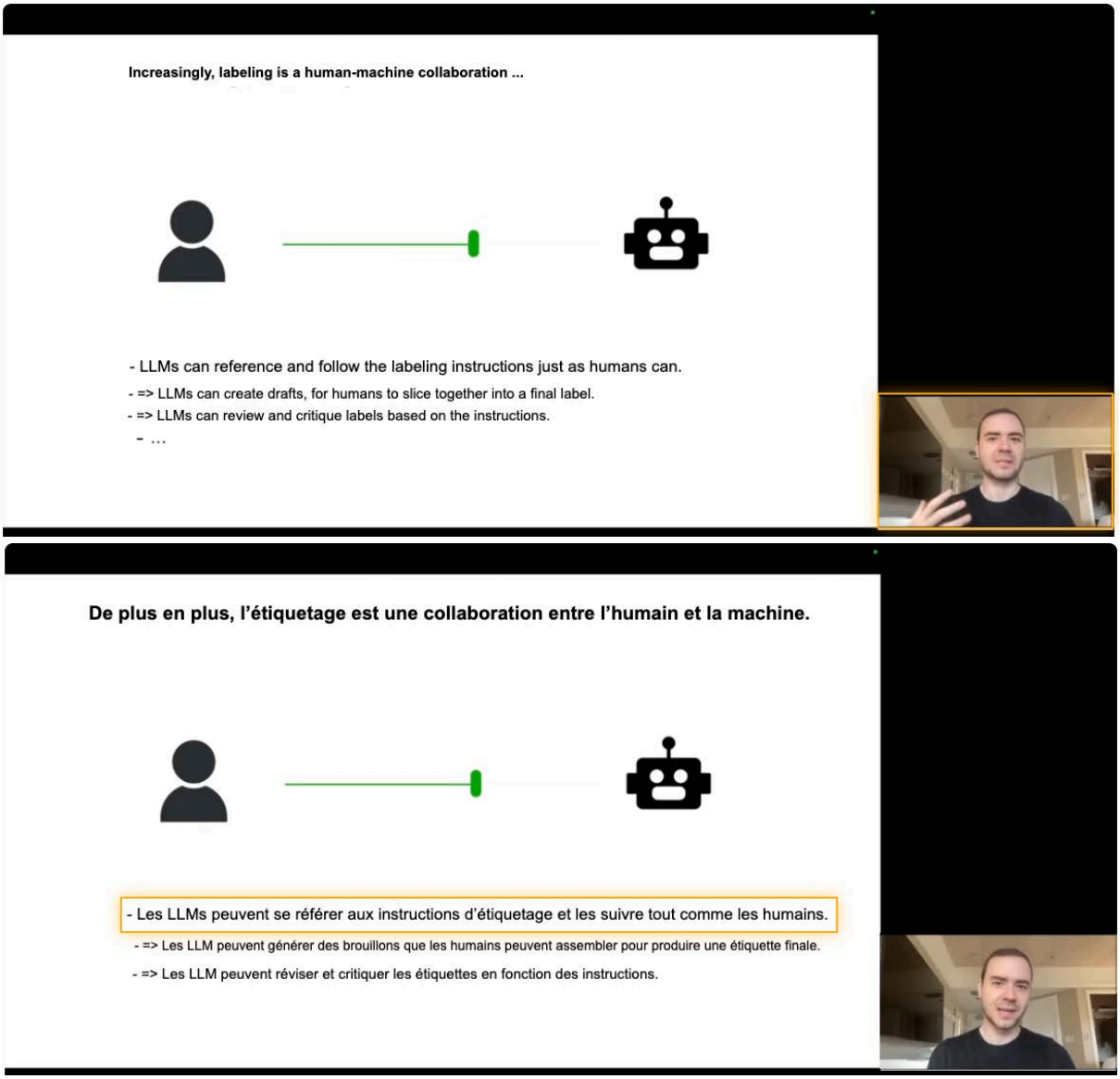


Figure 5.11 – Two versions of a video with different translations. Top: English, Bottom: French.

5.6 Conclusion and Future Work

This chapter presented a proof-of-concept lecture video analysis toolkit that integrates scene detection, transcript generation, OCR, and semantic linking into a unified and interactive system. Built upon the components developed in earlier chapters, the application demonstrates how multimodal processing techniques can be combined to enhance the way users engage with lecture video content. By offering synchronized access to spoken, visual, and textual elements, the toolkit provides flexible and meaningful entry points for navigation, review, and exploration.

A key contribution of this work lies in its open-source nature. While many existing systems in this domain remain closed or inaccessible, this application is openly available and designed to be extended, evaluated, and adapted by researchers, educators, and developers.

Beyond its technical scope, the application also reflects a broader shift toward user-centered and accessibility-aware design. Early discussions with neurodivergent users played a central role in shaping the interface and interaction model. Features such as the interactive slide layer, semantic linking between transcript and visual content, and customizable navigation views were introduced in response to user feedback highlighting the need for predictable layouts, customizable content, and multimodal engagement. While these elements are still evolving, they point toward promising directions for improving cognitive accessibility in educational media systems.

Several limitations remain. The accuracy of transcript generation and OCR depends on input quality, and scene detection may miss subtle slide transitions. Linking across modalities using embeddings, while effective in many cases, may require further tuning for specific domains. Most importantly, the system has not yet undergone formal user testing or large-scale evaluation, particularly in terms of accessibility impact or learning outcomes.

Future work will focus on extending the personalization and accessibility dimensions of the application. This includes adding support for audio feedback, enabling user-driven content retargeting, and experimenting with alternative layouts tailored to different learner profiles. Co-design sessions with neurodivergent learners and educators are planned to refine the interface and better understand how users with diverse cognitive needs interact with structured video content. A study to validate the final interface should take place in early 2026.

Another important direction is the extension of the system's event detection capabilities. In addition to slide transitions, the application could be augmented to identify events such as transitions to the presenter's camera, embedded video segments, annotations, presenter gestures, or pointer and laser movements. Capturing these rich interaction signals would offer a more complete understanding of the lecture.

Finally, while the current system is designed for slide-based lecture videos, future work should also explore its adaptation to non-slide formats, particularly blackboard- or whiteboard-based lectures. These formats pose unique challenges for segmentation, text extraction, and multimodal alignment, and extending the toolkit to handle them would significantly broaden its applicability across educational contexts.

Summary

This chapter presents a proof-of-concept application for lecture video analysis that integrates multiple modalities—speech, visual content, and semantic structure—into a unified and interactive system. Building on the methods and components developed in earlier chapters, the application performs automatic scene detection, transcript generation, OCR-based slide content extraction, and semantic alignment between spoken and visual elements. Designed with an emphasis on openness, reproducibility, and accessibility, the system is fully open source and intended as a research platform for further development. The interface enables non-linear navigation through lecture content via synchronized views of the transcript, scene thumbnails, and extracted text, while also supporting interactive exploration through features such as semantic linking and a customizable visual layer. Early feedback from neurodivergent users informed key design choices aimed at improving cognitive accessibility and layout consistency. The application not only demonstrates the feasibility of combining state-of-the-art techniques into an extensible pipeline, but also highlights opportunities for future work in personalization, adaptive learning, and inclusive educational technology. A demonstration of the application is available at https://travis-seng.fr/svla.

Chapter 6

Conclusion

6.1 Contributions

In this thesis, we tackled three key challenges in the multimodal analysis of educational videos: data scarcity, user accessibility, and content understanding. To address these issues, we introduced three main contributions.

First, to mitigate the problem of data scarcity and enhance visual analysis, we developed a tool capable of generating realistic synthetic slides with annotations. This tool improves model training by producing various types of annotations, including bounding boxes, semantic segmentation masks, and image-to-text labels, tailored to specific tasks. Our analysis revealed that, while numerous document datasets exist, few focus specifically on slides, and those that do often suffer from inconsistent labeling and quality issues. We demonstrated that training models with our synthetic slides improves object detection performance while significantly reducing the need for manual annotation. This work was published at the ICDAR conference in 2024 (Seng et al. 2024).

Building on this, our second contribution extends beyond visual analysis to a broader multimodal and temporal perspective, aiming to improve downstream tasks such as summarization and indexing. We proposed a method for segmenting long educational videos into meaningful sections without relying on large language models, which, while powerful, are computationally expensive. Instead, we leveraged lightweight pretrained models to extract multimodal features and aligned them in an unsupervised manner. Additionally, we classified videos into categories such as slides, blackboards, and others, allowing for more tailored processing during segmentation. Our method achieved state-of-the-art results among unsupervised approaches across multiple datasets. As part of this work, we also introduced two new datasets: the Yale Dataset and the VideoLecture Dataset. The classification of lecture videos was a work published at the Doctoral Symposium of ACM Multimedia 2022 (Seng 2022), while the remaining part is submitted at WACV 2026.

Finally, to apply the insights from our research, we designed a Lecture Video Toolkit, an open-source platform for lecture video analysis and applications. The toolkit includes a set of tools that make lecture videos more interactive, structured, and informative. By releasing it as open-source, we aim to make these capabilities widely accessible and support further development and use across the community. This work is accepted at the open-source track of ACM Multimedia 2025.

Together, these contributions lay a baseline for future advancements in educational video analysis and its applications. Ultimately, this work seeks to create a more engaging and effective learning environment for students, instructors, and researchers alike.

6.2 Future work

While this thesis has addressed key challenges in the multimodal analysis of educational videos, several promising directions remain for future exploration.

One important avenue is improving our methods beyond slide-based lectures to other types of instructional content, such as blackboard-based lectures, conference presentations, and meetings to enable better applications for these types. Each of these formats presents unique challenges. Blackboard lectures involve continuous handwriting and evolving visual content, requiring specialized recognition techniques to track and interpret information over time. Conference presentations often feature dynamic speaker movements, varied visual materials, and audience interactions that complicate segmentation and analysis. Meetings, on the other hand, are highly conversational and require advanced speaker diarization, dialogue structuring, and topic modeling. Adapting our techniques to these diverse settings would further enhance the applicability of lecture video analysis.

Another promising direction is the integration of multimodal large language models (MLLMs) to improve content understanding. While our approach deliberately avoids reliance on MLLMs due to their high computational costs, these models hold significant potential for tasks such as automatic lecture summarization, key concept extraction, and topic segmentation. Future work could explore hybrid approaches that combine the efficiency of lightweight multimodal methods with the semantic understanding provided by MLLMs, selectively applying them where their benefits outweigh the costs.

Beyond segmentation and content structuring, future research could also focus on expanding the range of applications for lecture video analysis. Potential developments include adaptive learning systems that personalize content based on user engagement, automated assessment tools that generate quizzes from lecture content, and interactive search systems that allow users to retrieve specific concepts or explanations within a video. Additionally, integrating lecture analysis with augmented and virtual reality technologies could open new possibilities for immersive and interactive learning experiences.

An important ongoing effort focuses on enhancing accessibility, particularly for neurodivergent users who may face challenges with traditional lecture video formats. In collaboration with psychologists from the CERPPS (Centre d'études et de recherches en psychopathologie et santé) and Atypie-Friendly, we aim to improve comprehension and reduce cognitive load through inclusive design. One current development involves an animated avatar that acts as a visual guide to help maintain attention and break down complex content, especially for neurodivergent students.

By exploring these directions, future research can further improve the accessibility, usability, and overall impact of educational video analysis, ultimately enhancing learning experiences for a wide range of users.

Bibliography

Adcock, John, Matthew Cooper, Laurent Denoue, Hamed Pirsiavash, and Lawrence A Rowe. 2010a. "Talkminer: A Lecture Webcast Search Engine". In *Proceedings of the 18th ACM International Conference on Multimedia*, 241–50.

Adcock, John, Matthew Cooper, Laurent Denoue, Hamed Pirsiavash, and Lawrence A. Rowe. 2010b. "TalkMiner: A Search Engine for Online Lecture Video". In *Proceedings of the International Conference on Multimedia - MM '10*, 1507. Firenze, Italy: ACM Press. https://doi.org/10.1145/1873951.1874263.

Alliance for Open Media. 2023. "AV1 Bitstream & Decoding Process Specification."

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. "VQA: Visual Question Answering". In , 2425–33. https://openaccess.thecvf.com/content_iccv_2015/html/Antol_VQA_Visual_Question_ICCV_2015_paper.html.

Araujo, A., J. Chaves, H. Lakshman, R. Angst, and B. Girod. 2015. "Large-Scale Query-by-Image Video Retrieval Using Bloom Filters". *Arxiv*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate". In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Yoshua Bengio and Yann LeCun. http://arxiv.org/abs/1409.0473.

Biswas, Arijit, Ankit Gandhi, and Om Deshmukh. 2015. "MMToC: A Multimodal Method for Table of Content Creation in Educational Videos". In *Proceedings of the 23rd ACM International Conference on Multimedia*, 621–30. Brisbane Australia: ACM. https://doi.org/10.1145/2733373.2806253.

Blanc-Beyne, Thibault, Axel Carlier, Sandrine Mouysset, and Vincent Charvillat. 2021. "Unsupervised Human Pose Estimation on Depth Images". In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*, 358–73.

Blei, David M, Andrew Y Ng, and Michael I Jordan. 2003. "Latent Dirichlet Allocation". *Journal of Machine Learning Research* 3 (Jan): 993–1022.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. "Enriching Word Vectors with Subword Information". Edited by Lillian Lee, Mark Johnson, and Kristina Toutanova. *Transactions of the Association for Computational Linguistics* 5:135–46. https://doi.org/10.1162/tacl_a_00051.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. "On the Opportunities and Risks of Foundation Models". arXiv. July 2022. https://doi.org/10.48550/arXiv.2108.07258.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020a. "Language Models Are Few-Shot Learners". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, 33:1877–1901. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020b. "Language Models Are Few-Shot Learners". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, 33:1877–1901. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Capobianco, Samuele, and Simone Marinai. 2017. "DocEmul: A Toolkit to Generate Structured Historical Documents". In , 1186–91. https://doi.org/10.1109/ICDAR.2017.196.

Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. "Emerging Properties in Self-Supervised Vision Transformers". In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Castellano, Brandon. 2018. "Pyscenedetect: Intelligent Scene Cut Detection and Video Splitting Tool."

Chan, William, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition". In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 0:4960–64. https://doi.org/10.1109/ICASSP.2016.7472621.

Chand, Dipesh, and Hasan Oğul. 2021. "A Framework for Lecture Video Segmentation from Extracted Speech Content". In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 299–304.

Che, Xiaoyin, Haojin Yang, and Christoph Meinel. 2013. "Lecture Video Segmentation by Automatically Analyzing the Synchronized Slides". In *Proceedings of the 21st ACM International Conference on Multimedia*, 345–48.

Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, et al. 2021a. "Evaluating Large Language Models Trained on Code". arXiv. July 2021. https://doi.org/10.48550/arXiv.2107.03374.

Chen, Min-Hung, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. 2020. "Action Segmentation with Joint Self-Supervised Temporal Domain Adaptation". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9454–63.

Chen, Shixing, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. 2021b. "Shot Contrastive Self-Supervised Learning for Scene Boundary Detection". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9796–9805.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation". In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1724–34. Doha, Qatar: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179.

Choi, Freddy YY. 2000. "Advances in Domain Independent Linear Text Segmentation". In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Chung, Junyoung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Cisco. 2022. "Cisco Visual Networking Index: Forecast and Trends, 2017–2022". https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2022_Forecast_Highlights.pdf.

Costley, Jamie, Mik Fanguy, Chris Lange, and Matthew Baldwin. 2021. "The Effects of Video Lecture Viewing Strategies on Cognitive Load". *Journal of Computing in Higher Education* 33 (1): 19–38. https://doi.org/10.1007/s12528-020-09254-y.

Cotsaces, Costas, Nikos Nikolaidis, and Ioannis Pitas. 2006. "Video Shot Detection and Condensed Representation. A Review". *IEEE Signal Processing Magazine* 23 (2): 28–37.

Das, Ananda, and Partha Pratim Das. 2019. "Automatic Semantic Segmentation and Annotation of MOOC Lecture Videos". In *International Conference on Asian Digital Libraries*, 181–88.

Das, Ananda, and Partha Pratim Das. 2020. "Incorporating Domain Knowledge to Improve Topic Segmentation of Long MOOC Lecture Videos". *Arxiv Preprint Arxiv:2012.07589*.

Davila, Kenny, Fei Xu, Srirangaraj Setlur, and Venu Govindaraju. 2021. "FCN-LectureNet: Extractive Summarization of Whiteboard and Chalkboard Lecture Videos". *IEEE Access* 9:104469–84. https://doi.org/10.1109/ACCESS.2021.3099427.

DeepMind, Gemini, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, et al. 2024. "Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context". arXiv. December 2024. https://doi.org/10.48550/arXiv.2403.05530.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. "Imagenet: A Large-Scale Hierarchical Image Database". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 0:248–55. https://doi.org/10.1109/CVPR.2009.5206848.

Deng, Yuntian, Anssi Kanervisto, Jeffrey Ling, and Alexander M. Rush. 2017. "Image-to-Markup Generation with Coarse-to-Fine Attention."

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, edited by Jill Burstein, Christy Doran, and Thamar Solorio, 4171–86. Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423.

Dimitsas, Markos, and Jochen L Leidner. 2023. "Topic Segmentation of Educational Video Lectures Using Audio and Text". In *European Conference on Artificial Intelligence*, 447–58.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. 2021. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy.

Ferreira, Anselmo, Ehsan Nowroozi, and Mauro Barni. 2021. "VIPPrint: Validating Synthetic Image Detection and Source Linking Methods on a Large Scale Dataset of Printed Documents". *Journal of Imaging* 7 (3): 50. https://doi.org/10.3390/jimaging7030050.

Galanopoulos, Damianos, and Vasileios Mezaris. 2019. "Temporal Lecture Video Fragmentation Using Word Embeddings". In *Multimedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part II 25*, 254–65.

Galley, Michel, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. "Discourse Segmentation of Multi-Party Conversation". In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 562–69. Sapporo, Japan: Association for Computational Linguistics. https://doi.org/10.3115/1075096.1075167.

Gandhi, Ankit, Arijit Biswas, and Om Deshmukh. 2015. "Topic Transition in Educational Videos Using Visually Salient Words.". *International Educational Data Mining Society*.

Gao, Shang-Hua, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. 2021. "Global2local: Efficient Structure Search for Video Action Segmentation". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16805–14.

Geri, Nitza, Amir Winer, and Beni Zaks. 2017. "Challenging the Six-Minute Myth of Online Video Lectures: Can Interactivity Expand the Attention Span of Learners?". *Online Journal of Applied Knowledge Management ISSN 2325-4688* 5:101–11. https://doi.org/10.36965/OJAKM.2017.5(1)101-111.

Giannakos, Michail N., Konstantinos Chorianopoulos, and Nikos Chrisochoides. 2015. "Making Sense of Video Analytics: Lessons Learned from Clickstream Interactions, Attitudes, And Learning Outcome in a Video-Assisted Course". *International Review of Research in Open and Distributed Learning* 16 (1): 260–83. https://doi.org/10.19173/irrodl.v16i1.1976.

Glava s, Goran, and Swapna Somasundaran. 2020. "Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation". *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (5): 7797–7804. https://doi.org/10.1609/aaai.v34i05.6284.

Glavaš, Goran, Federico Nanni, and Simone Paolo Ponzetto. 2016. "Unsupervised Text Segmentation Using Semantic Relatedness Graphs". In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, edited by Claire Gardent, Raffaella Bernardi, and Ivan Titov, 125–30. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/S16-2016.

Guo, Philip J., Juho Kim, and Rob Rubin. 2014. "How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos". In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, 41–50. Atlanta Georgia USA: ACM. https://doi.org/10.1145/2556325.2566239.

Gupta, Anchit, CV Jawahar, Makarand Tapaswi, and others. 2023. "Unsupervised Audio-Visual Lecture Segmentation". In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5232–41.

Haurilet, Monica, Ziad Al-Halah, and Rainer Stiefelhagen. 2019a. "SPaSe - Multi-Label Page Segmentation for Presentation Slides". In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 726–34. Waikoloa Village, HI, USA: IEEE. https://doi.org/10.1109/WACV.2019.00082.

Haurilet, Monica, Alina Roitberg, Manuel Martinez, and Rainer Stiefelhagen. 2019b. “WiSe — Slide Segmentation in the Wild”. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 343–48. Sydney, Australia: IEEE. https://doi.org/10.1109/ICDAR.2019.00062.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. “Mask R-Cnn”. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 0:2980–88. https://doi.org/10.1109/ICCV.2017.322.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition”. arXiv. December 2015. https://doi.org/10.48550/arXiv.1512.03385.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition”. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 770–78. CVPR '16. Las Vegas, NV, USA: IEEE. https://doi.org/10.1109/CVPR.2016.90.

Hearst, Marti A. 1997. “Texttiling: Segmenting Text into Multi-Paragraph Subtopic Passages”. *Comput. Linguist.* 23 (1): 33–64.

Hershey, Shawn, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, et al. 2017. “CNN Architectures for Large-Scale Audio Classification”. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 0:131–35. https://doi.org/10.1109/ICASSP.2017.7952132.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory”. *Neural Computation* 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Huang, Jie, and Kevin Chen-Chuan Chang. 2023. “Towards Reasoning in Large Language Models: A Survey”. In *Findings of the Association for Computational Linguistics: ACL 2023*, edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 1049–65. Toronto, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.67.

ITU-T Video Coding Experts Group (VCEG). 2013. “High Efficiency Video Coding (HEVC)”. ITU-T.

Jeong, Hyun Ji, Tak-Eun Kim, and Myoung Ho Kim. 2012. “An Accurate Lecture Video Segmentation Method by Using Sift and Adaptive Threshold”. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, 285–88.

Ji, Lei, Chenfei Wu, Daisy Zhou, Kun Yan, Edward Cui, Xilin Chen, and Nan Duan. 2022. “Learning Temporal Video Procedure Segmentation from an Automatically Collected Large Dataset”. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1506–15.

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. "Survey of Hallucination in Natural Language Generation". *ACM Comput. Surv.* 55 (12): 1–38. https://doi.org/10.1145/3571730.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, et al. 2023. "Mistral 7b."

Jocher, Glenn, Ayush Chaurasia, and Jing Qiu. 2023. "Ultralytics YOLO". January 2023. https://github.com/ultralytics/ultralytics.

Kay, Anthony. 2007. "Tesseract: An Open-Source Optical Character Recognition Engine". *Linux Journal* 2007 (159): 2.

Kembhavi, Aniruddha, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. "A Diagram Is Worth a Dozen Images". Edited by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling 9908:235–51. https://doi.org/10.1007/978-3-319-46493-0_15.

Kim, Jeongyeon, Yubin Choi, Minsuk Kahng, and Juho Kim. 2022a. "Fitvid: Responsive and Flexible Video Content Adaptation". In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–16.

Kim, Jeongyeon, Yubin Choi, Minsuk Kahng, and Juho Kim. 2022b. "FitVid: Responsive and Flexible Video Content Adaptation". In *CHI Conference on Human Factors in Computing Systems*, 1–16. New Orleans LA USA: ACM. https://doi.org/10.1145/3491102.3501948.

Kim, Juho, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, and Robert C. Miller. 2014. "Data-Driven Interaction Techniques for Improving Navigation of Educational Videos". In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, 563–72. UIST '14. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2642918.2647389.

Koka, Raga Shalini, Farah Naz Chowdhury, Mohammad Rajiur Rahman, Thamar Solorio, and Jaspal Subhlok. 2020. "Automatic Identification of Keywords in Lecture Video Segments". In *2020 IEEE International Symposium on Multimedia (ISM)*, 162–65. https://doi.org/10.1109/ISM.2020.00035.

Koprinska, Irena, and Sergio Carrato. 2001. "Temporal Video Segmentation: A Survey". *Signal Processing: Image Communication* 16 (5): 477–500.

Koshorek, Omri, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. "Text Segmentation as a Supervised Learning Task". In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, edited by Marilyn Walker, Heng Ji, and Amanda Stent, 469–73. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-2075.

Kota, Bhargava Urala, Alexander Stone, Kenny Davila, Srirangaraj Setlur, and Venu Govindaraju. 2021. "Automated Whiteboard Lecture Video Summarization by Content Region Detection and Representation". In *2020 25th International Conference on Pattern Recognition (ICPR)*, 10704–11. Milan, Italy: IEEE. https://doi.org/10.1109/ICPR48806.2021.9412386.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks". In *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kukleva, Anna, Hilde Kuehne, Fadime Sener, and Jurgen Gall. 2019. "Unsupervised Learning of Action Classes with Continuous Temporal Embedding". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12066–74.

Lange, Christopher, and Jamie Costley. 2020. "Improving Online Video Lectures: Learning Challenges Created by Media". *International Journal of Educational Technology in Higher Education* 17 (1): 16. https://doi.org/10.1186/s41239-020-00190-6.

LeCun, Yann, Leon Bottou, Yoshua Bengio, and Patrick Ha. 1998. "Gradient-Based Learning Applied to Document Recognition."

Lee, Kenton, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. "Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding". arXiv. June 2023. https://doi.org/10.48550/arXiv.2210.03347.

Li, Junnan, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. "BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation". In *Proceedings of the 39th International Conference on Machine Learning*, edited by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, 162:12888–900. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v162/li22n.html.

Li, Minghao, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2020b. "TableBank: Table Benchmark for Image-Based Table Detection and Recognition". In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, et al., 1918–25. Marseille, France: European Language Resources Association. https://aclanthology.org/2020.lrec-1.236.

Li, Minghao, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020a. "DocBank: A Benchmark Dataset for Document Layout Analysis". In *Proceedings*

of the 28th International Conference on Computational Linguistics, edited by Donia Scott, Nuria Bel, and Chengqing Zong, 949–60. Barcelona, Spain (Online): International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.82.

Li, Yinghao Aaron, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2023. "StyleTTS 2: Towards Human-Level Text-to-Speech Through Style Diffusion and Adversarial Training with Large Speech Language Models". In *Advances in Neural Information Processing Systems*, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, 36:19594–621. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2023/file/3eaad2a0b62b5ed7a2e66c2188bb1449-Paper-Conference.pdf.

Lin, Tsung-Yi, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. "Feature Pyramid Networks for Object Detection". In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–44. Honolulu, HI: IEEE. https://doi.org/10.1109/CVPR.2017.106.

Liu, Ye, Lingfeng Qiao, Di Yin, Zhuoxuan Jiang, Xinghua Jiang, Deqiang Jiang, and Bo Ren. 2022b. "Os-Msl: One Stage Multimodal Sequential Link Framework for Scene Segmentation and Classification". In *Proceedings of the 30th ACM International Conference on Multimedia*, 6269–77.

Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022a. "A ConvNet for the 2020s."

Liwicki, Marcus, and Horst Bunke. 2005. "IAM-Ondb-an on-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard". In *Eighth International Conference on Document Analysis and Recognition (Icdar'05)*, 956–61.

Mahapatra, Debabrata, Ragunathan Mariappan, Vaibhav Rajan, Kuldeep Yadav, Seby A., and Sudeshna Roy. 2018. "VideoKen: Automatic Video Summarization and Course Curation to Support Learning". In *Companion Proceedings of the The Web Conference 2018*, 239–42. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3184558.3186988.

Masry, Ahmed, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. "ChartQA: A Benchmark for Question Answering About Charts with Visual and Logical Reasoning". In *Findings of the Association for Computational Linguistics: ACL 2022*, edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 2263–79. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.177.

Methani, Nitesh, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. "PlotQA: Reasoning over Scientific Plots". arXiv. February 2020. http://arxiv.org/abs/1909.00997.

Mihalcea, Rada, and Paul Tarau. 2004. "TextRank: Bringing Order into Text". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, edited

by Dekang Lin and Dekai Wu, 404–11. Barcelona, Spain: Association for Computational Linguistics. https://aclanthology.org/W04-3252.

Mikolov, Tomás, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space". In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, edited by Yoshua Bengio and Yann LeCun. http://arxiv.org/abs/1301.3781.

Mukhopadhyay, Sugata, and Brian Smith. 1999. "Passive Capture and Structuring of Lectures". In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1) - MULTIMEDIA '99*, 477–87. Orlando, Florida, United States: ACM Press. https://doi.org/10.1145/319463.319690.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. 2019. "Representation Learning with Contrastive Predictive Coding". arXiv. January 2019. https://doi.org/10.48550/arXiv.1807.03748.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2024. "GPT-4 Technical Report". arXiv. March 2024. https://doi.org/10.48550/arXiv.2303.08774.

Oquab, Maxime, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, et al. n.d. "DINOv2: Learning Robust Visual Features Without Supervision."

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. "Training Language Models to Follow Instructions with Human Feedback."

Pappano, Laura. 2012. "The Year of the Mooc". *The New York Times* 2 (12): 2012.

Paruchuri, Vik. 2025. "VikParuchuri/surya". May 2025. https://github.com/VikParuchuri/surya.

Peng, Yi-Hao, Jeffrey P Bigham, and Amy Pavel. 2021a. "Slidecho: Flexible Non-Visual Exploration of Presentation Videos". In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, 1–12.

Peng, Yi-Hao, Jeffrey P Bigham, and Amy Pavel. 2021d. "Slidecho: Flexible Non-Visual Exploration of Presentation Videos". In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, 1–12. Virtual Event USA: ACM. https://doi.org/10.1145/3441852.3471234.

Peng, Yi-Hao, Peggy Chi, Anjuli Kannan, Meredith Ringel Morris, and Irfan Essa. 2023. "Slide Gestalt: Automatic Structure Extraction in Slide Decks for Non-Visual Access". In

Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 1–14. Hamburg Germany: ACM. https://doi.org/10.1145/3544548.3580921.

Peng, Yi-Hao, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. 2021b. "Say It All: Feedback for Improving Non-Visual Presentation Accessibility". In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–12. Yokohama Japan: ACM. https://doi.org/10.1145/3411764.3445572.

Peng, Zhiliang, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. 2021c. "Conformer: Local Features Coupling Global Representations for Visual Recognition". In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 367–76.

Pfitzmann, Birgit, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. "DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation". In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3743–51. Washington DC USA: ACM. https://doi.org/10.1145/3534678.3539043.

Piczak, Karol J. 2015. "Environmental Sound Classification with Convolutional Neural Networks". In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. https://doi.org/10.1109/MLSP.2015.7324337.

Pisaneschi, Lorenzo, Andrea Gemelli, and Simone Marinai. 2023. "Automatic Generation of Scientific Papers for Data Augmentation in Document Layout Analysis". *Pattern Recognition Letters* 167 (March):38–44. https://doi.org/10.1016/j.patrec.2023.01.018.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, et al. 2021. "Learning Transferable Visual Models from Natural Language Supervision". In *International Conference on Machine Learning*, 8748–63.

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023a. "Robust Speech Recognition via Large-Scale Weak Supervision". In *International Conference on Machine Learning*, 28492–518.

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023b. "Robust Speech Recognition via Large-Scale Weak Supervision". In *Proceedings of the 40th International Conference on Machine Learning*, edited by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, 202:28492–518. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v202/radford23a.html.

Rahman, Mohammad Rajiur, Shishir Shah, and Jaspal Subhlok. 2020. "Visual Summarization of Lecture Video Segments for Enhanced Navigation". In *2020 IEEE International Symposium on Multimedia (ISM)*, 154–57. Naples, Italy: IEEE. https://doi.org/10.1109/ISM.2020.00033.

Rao, Anyi, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. "A Local-to-Global Approach to Multi-Modal Movie Scene Segmentation". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10146–55.

Redmon, Joseph, and Ali Farhadi. 2018. "Yolov3: An Incremental Improvement". *Arxiv*.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–88.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In *Advances in Neural Information Processing Systems*, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

Richter, Stephan R, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. "Playing for Data: Ground Truth from Computer Games". In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 102–18.

Robertson, Barbara, and Mark J. Flowers. 2020. "Determining the Impact of Lecture Videos on Student Outcomes". *Learning and Teaching: The International Journal of Higher Education in the Social Sciences* 13 (2): 25–40. https://doi.org/10.3167/latiss.2020.130203.

Sarfraz, M. Saquib, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. 2021a. "Temporally-Weighted Hierarchical Clustering for Unsupervised Action Segmentation". In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11220–29. Nashville, TN, USA: IEEE. https://doi.org/10.1109/CVPR46437.2021.01107.

Sarfraz, Saquib, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. 2021b. "Temporally-Weighted Hierarchical Clustering for Unsupervised Action Segmentation". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11225–34.

Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, et al. 2024. "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models". In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 25278–94. NIPS '22. Red Hook, NY, USA: Curran Associates Inc.

Sechayk, Yotam, Ariel Shamir, and Takeo Igarashi. 2024. "SmartLearn: Visual-Temporal Accessibility for Slide-Based E-Learning Videos". In *Extended Abstracts of the CHI Conference*

on Human Factors in Computing Systems, 1–11. Honolulu HI USA: ACM. https://doi.org/10.1145/3613905.3650883.

Seng, Travis. 2022. "Enriching Existing Educational Video Datasets to Improve Slide Classification and Analysis". In *Proceedings of the 30th ACM International Conference on Multimedia*, 6930–34. MM '22. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3503161.3548758.

Seng, Travis, Axel Carlier, Thomas Forgione, Vincent Charvillat, and Wei Tsang Ooi. 2024. "SlideCraft: Synthetic Slides Generation for Robust Slide Analysis". In *Document Analysis and Recognition - ICDAR 2024*, edited by Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng, 79–96. Cham: Springer Nature Switzerland.

Shah, Rajiv Ratn, Yi Yu, Anwar Dilawar Shaikh, and Roger Zimmermann. 2015. "TRACE: Linguistic-Based Approach for Automatic Lecture Video Segmentation Leveraging Wikipedia Texts". In *2015 IEEE International Symposium on Multimedia (ISM)*, 217–20.

Shah, Rajiv Ratn, Yi Yu, Anwar Dilawar Shaikh, Suhua Tang, and Roger Zimmermann. 2014. "ATLAS: Automatic Temporal Segmentation and Annotation of Lecture Videos Based on Modelling Transition Time". In *Proceedings of the 22nd ACM International Conference on Multimedia*, 209–12.

Simonyan, Karen, and Andrew Zisserman. 2015. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Yoshua Bengio and Yann LeCun. http://arxiv.org/abs/1409.1556.

Soares, Eduardo R, and Eduardo Barrére. 2018. "Automatic Topic Segmentation for Video Lectures Using Low and High-Level Audio Features". In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, 189–96.

Soares, Eduardo R, and Eduardo Barrére. 2019. "An Optimization Model for Temporal Video Lecture Segmentation Using Word2vec and Acoustic Features". In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, 513–20.

Solbiati, Alessandro, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. "Unsupervised Topic Segmentation of Meetings with BERT Embeddings". 2021. https://arxiv.org/abs/2106.12978.

Song, Kaitao, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. "MPNet: Masked and Permuted Pre-Training for Language Understanding". arXiv. November 2020. https://doi.org/10.48550/arXiv.2004.09297.

Souček, Tomáš, and Jakub Lokoč. 2020. "Transnet V2: An Effective Deep Network Architecture for Fast Shot Transition Detection". *Arxiv Preprint Arxiv:2008.04838*.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. “Sequence to Sequence Learning with Neural Networks”. In *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc.. https://proceedings.neurips.cc/paper_files/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Tan, Mingxing, and Quoc Le. 2019. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In *Proceedings of the 36th International Conference on Machine Learning*, edited by Kamalika Chaudhuri and Ruslan Salakhutdinov, 97:6105–14. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v97/tan19a.html.

Tanaka, Ryota, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. “SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images”. arXiv. January 2023. https://doi.org/10.48550/arXiv.2301.04883.

Team, Qwen. 2024. “Qwen2.5: A Party of Foundation Models”. September 2024. https://qwenlm.github.io/blog/qwen2.5/.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. 2023. “LLaMA: Open and Efficient Foundation Language Models”. arXiv. February 2023. https://doi.org/10.48550/arXiv.2302.13971.

Tuna, Tayfun, Mahima Joshi, Varun Varghese, Rucha Deshpande, Jaspal Subhlok, and Rakesh Verma. 2015. “Topic Based Segmentation of Classroom Videos”. In *2015 IEEE Frontiers in Education Conference (FIE)*, 1–9.

Turnbull, Darren, Ritesh Chugh, and Jo Luck. 2021. “Transitioning to E-Learning during the COVID-19 Pandemic: How Have Higher Education Institutions Responded to the Challenge?”. *Education and Information Technologies* 26 (5): 6401–19. https://doi.org/10.1007/s10639-021-10633-w.

Vaswani, A. 2017. “Attention Is All You Need”. *Advances in Neural Information Processing Systems*.

Walk, Ian, Arnold Yim, Ed Novak, Charles Reiss, and Daniel Graham. 2020. “Redesigning the Online Video Lecture Player to Promote Active Learning”. In *2020 IEEE Frontiers in Education Conference (FIE)*, 1–5. Uppsala, Sweden: IEEE. https://doi.org/10.1109/FIE44824.2020.9273898.

Wang, Jing, Weiqing Min, Sujuan Hou, Shengnan Ma, Yuanjie Zheng, and Shuqiang Jiang. 2022. “Logodet-3k: A Large-Scale Image Dataset for Logo Detection”. *ACM Transactions on Multimedia Computing, Communications, And Applications (TOMM)* 18 (1): 1–19.

Weng, Jiahao, Chao Zhang, Xi Yang, and Haoran Xie. 2022. “HiVideo: Hierarchical Browsing Interface for Educational Videos”. In *ACM SIGGRAPH 2022 Posters*, 1–2. Vancouver BC Canada: ACM. https://doi.org/10.1145/3532719.3543226.

Xu, Chengpei, Ruomei Wang, Shujin Lin, Xiaonan Luo, Baoquan Zhao, Lijie Shao, and Mengqiu Hu. 2019. "Lecture2Note: Automatic Generation of Lecture Notes from Slide-Based Educational Videos". In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 898–903. https://doi.org/10.1109/ICME.2019.00159.

Yoo, Taewon, Hyewon Jeong, Donghwan Lee, and Hyunggu Jung. 2021. "LectYS: A System for Summarizing Lecture Videos on YouTube". In *26th International Conference on Intelligent User Interfaces*, 90–92. College Station TX USA: ACM. https://doi.org/10.1145/3397482.3450722.

Zhao, Baoquan, Shujin Lin, Xin Qi, Ruomei Wang, and Xiaonan Luo. 2018. "A Novel Approach to Automatic Detection of Presentation Slides in Educational Videos". *Neural Computing and Applications* 29 (5): 1369–82. https://doi.org/10.1007/s00521-017-3276-1.

Zhao, Baoquan, Songhua Xu, Shujin Lin, Ruomei Wang, and Xiaonan Luo. 2019a. "A New Visual Interface for Searching and Navigating Slide-Based Lecture Videos". In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 928–33. https://doi.org/10.1109/ICME.2019.00164.

Zhao, Baoquan, Songhua Xu, Shujin Lin, Ruomei Wang, and Xiaonan Luo. 2019b. "A New Visual Interface for Searching and Navigating Slide-Based Lecture Videos". In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 0:928–33. https://doi.org/10.1109/ICME.2019.00164.

Zhong, Ming, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, et al. 2021. "Qmsum: A New Benchmark for Query-Based Multi-Domain Meeting Summarization". *Arxiv Preprint Arxiv:2104.05938*.

Zhong, Xu, Jianbin Tang, and Antonio Jimeno-Yepes. 2019. "PubLayNet: Largest Dataset Ever for Document Layout Analysis". In . https://doi.org/10.1109/ICDAR.2019.00166.

"Marp-Team/marp." 2024. Marp. February 2024. https://github.com/marp-team/marp.

Titre : Analyse des données multimodales enregistrées lors de visioconférences

Mots clés : Traitement de contenus multimédia, Intelligence artificielle, Vidéo numérique

Résumé : La vidéo est devenue un média central dans l'enseignement d'aujourd'hui, et les vidéos de cours jouent un rôle croissant dans la diffusion des savoirs. Leur évolution, depuis les enregistrements bruts de cours en salle jusqu'aux contenus structurés et professionnels des MOOCs, témoigne de transformations profondes dans les pratiques pédagogiques et les infrastructures numériques. Malgré leur importance croissante, les vidéos de cours posent encore des défis en termes d'accessibilité, de navigation et d'engagement, en particulier en raison de leur nature multimodale, combinant informations visuelles, sonores et textuelles.
Cette thèse explore comment l'analyse multimodale peut améliorer l'utilisabilité et la valeur pédagogique des vidéos de cours. Elle s'appuie sur les avancées récentes de l'apprentissage profond, notamment les modèles de type transformer, pour développer de nouvelles méthodes d'analyse et de segmentation du contenu vidéo éducatif. Deux contributions techniques principales sont proposées.
La première est un outil de génération de jeux de données synthétiques de diapositives avec des annotations détaillées et cohérentes. Les bases de données existantes présentent de nombreuses limites : annotations incomplètes, classes ambiguës, et manque d'homogénéité. L'outil permet de créer des jeux de données diversifiés, équilibrés et proches des présentations réelles, tout en réduisant le besoin d'annotation manuelle. L'outil offre également la possibilité de personnaliser les classes et les styles, facilitant la constitution de corpus standardisés adaptés à des besoins spécifiques et améliorant ainsi la fiabilité de l'apprentissage automatique pour l'analyse de diapositives.
La seconde contribution concerne la segmentation temporelle des vidéos de cours basé sur des représentations multimodales. Cette méthode aligne les caractéristiques visuelles extraites des images vidéo, le texte issu de l'OCR sur les diapositives, et les transcriptions audio, à l'aide de modèles visuo-textuels pré-entraînés. En intégrant des informations de transition et en s'adaptant à différents types de vidéos (diapositives projetées, tableau noir, captures d'écran), le système permet une segmentation plus précise, plus rapide et plus robuste que les approches existantes, sans nécessiter de grands ensembles de données annotées.
Enfin, une application de démonstration illustre le potentiel pratique de ces contributions : elle améliore la recherche, l'interactivité et l'accessibilité des vidéos de cours. En structurant mieux le contenu et en facilitant la navigation, ce travail ouvre la voie à des expériences d'apprentissage plus efficaces. En somme, cette thèse propose des outils et des méthodes pour une analyse multimodale plus fiable des vidéos éducatives, en vue de les rendre plus accessibles, engageantes et pédagogiquement pertinentes.

Title: Analysis of multimodal data recorded during videoconferences

Key words: Multimedia content processing, Artificial intelligence, Digital video

Abstract: Video has become a central medium in education, with lecture videos playing an increasingly important role in knowledge dissemination. Lecture videos have evolved from simple classroom recordings to professionally produced MOOCs, reflecting wider shifts in both teaching methods and digital infrastructure. Despite their growing presence, lecture videos still pose challenges for learners and educators, particularly in terms of accessibility, navigability, and content engagement. These challenges are heightened by the multimodal nature of lecture videos, which combine visual, auditory, and textual elements.
This thesis investigates how multimodal analysis can improve the usability and educational value of lecture videos. It builds on recent progress in deep learning, especially transformer-based models, to develop new techniques for analyzing and segmenting educational video content. The work makes two main technical contributions.
First, it introduces a tool for generating synthetic slide datasets with detailed and consistent annotations. Existing slide datasets suffer from inconsistencies, incomplete labeling, and ambiguous class definitions—issues that compromise the training of layout and object detection models. The tool addresses this by producing diverse, high-quality synthetic slides that closely imitate real-world presentations. This synthetic data complements existing resources, reduces the need for manual annotation, and allows for tailored dataset creation depending on the research task. The tool supports the creation of standardized and customizable slide corpora, enabling more reliable and focused model training.
Second, the thesis proposes a novel framework for temporal segmentation of lecture videos using multimodal embeddings. The method combines visual features from video frames, OCR-extracted slide content, and text embeddings from transcripts, aligning them through pre-trained vision-language models. By integrating transition cues and adapting to different lecture formats—such as screen recordings or blackboard sessions—the framework segments videos faster and more accurately than existing approaches. It achieves strong performance without relying on large-scale annotated training sets, making it both scalable and generalizable across diverse lecture types.
Finally, a proof-of-concept application demonstrates how these methods can enhance video search, interactivity, and accessibility. By improving structure and content alignment, the system supports more effective engagement with educational video material. Overall, this research contributes tools and methods that support more robust, multimodal processing of lecture videos, with the goal of making them more navigable, inclusive, and pedagogically effective.