





KAUST Academy
AI Summer School Program - Oxford

Project Final Report



V2N: Deep Learning-Based System for Transforming Educational Videos into Interactive Jupyter Notebooks

Team Members

Mohammed Alsaif
Qasim Mahfoodh
Sheikha Shaheen
Wajd Alhuzaym
Feras Alsadeq

Project Mentor
Dr. Salman Khan

2025

V2N: Deep Learning-Based System for Transforming Educational Videos into Interactive Jupyter Notebooks

Abstract

YouTube has emerged as a leading platform for self-directed learning, offering vast educational content across domains such as programming, science, and mathematics. However, its passive consumption model limits deep understanding and practical engagement. This project introduces an AI-powered pipeline that enhances educational YouTube videos by generating structured, interactive learning experiences. Using our web app, users can send a YouTube video URL to the system to automatically analyze the video and then generate an educational Jupyter notebook containing summaries, explanations, and interactive tasks to reinforce understanding. By combining Automatic Speech Recognition for transcripts with Vision-Language Models for visual analysis and Large-Language Models for language analysis and generation, the system bridges the gap between watching and active learning, turning YouTube into a dynamic, learner-centric environment.

Contents

1 Introduction **2**

2 Related Work **2**

3 Methodology and approach **3**

3.1 Transcript Extraction 3

3.2 Keyframe Detection 3

3.3 Transcript-Visual Alignment 4

3.4 Visual Understanding with VLM 4

3.5 Fusion and Cleaning 4

3.6 Notebook Generation 4

4 Evaluation **5**

4.1 Criteria 5

4.2 Dataset 5

4.3 Results 6

5 Challenges and Obstacles **7**

6 Final Product **7**

7 Conclusion and Future Work **7**

8 References **8**

1 Introduction

Educational videos, particularly on platforms like YouTube, have become a cornerstone of modern learning, offering accessible knowledge on a vast array of subjects. However, this medium often encourages passive consumption rather than active engagement. For technical fields such as programming, science, and engineering, this presents a significant challenge. True comprehension in these areas is built upon hands-on practice, experimentation, and direct interaction with the material—an experience that traditional video formats cannot provide on their own.

There is a critical gap between watching a tutorial and actually solving and engaging with the material. This project addresses this gap by creating a bridge from passive viewing to active learning. By automatically transforming a selected video by the user through our Web app into interactive and executable formats, we can dramatically enhance the educational value of online content. The real-world impact is a more effective, engaging, and efficient learning experience that empowers self-learners to master complex technical skills.

2 Related Work

While working on the project, we came across a related paper that inspired us to proceed with this architecture. According to the paper "Context Selection and Rewriting for Video-based Educational Question Generation"[1] Using all content of the video (e.g., lecture, course) as context will result in vague and general responses. However, if you select relevant contexts from both transcript and keyframes, then merge them and rewrite them into coherent points. This would yield far more relevant and specific responses. The work of this paper was a huge influence in our project.

3 Methodology and approach

Our project follows a multi-stage pipeline to transform educational videos into structured Jupyter notebooks. The methodology combines speech recognition, computer vision, and large language models (LLMs) to automatically extract, process, and organize both audio and visual information into an executable notebook format. The overall workflow can be summarized in the following steps: Transcript Extraction, Keyframe Detection, Transcript-Visual Alignment, Visual Understanding with VLM, Fusion and Cleaning And finally Notebook Generation.

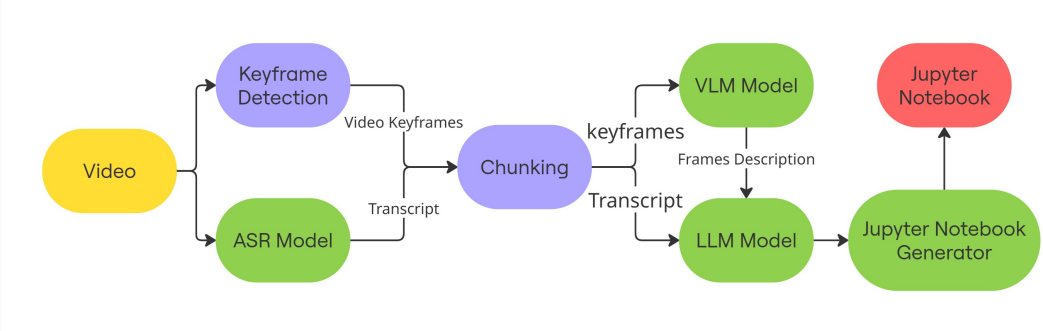


Figure 1: An overview of the system pipeline.

3.1 Transcript Extraction

We begin by extracting the textual content from the video using Whisper-base "Automatic Speech Recognition (ASR)" model. This step converts the spoken narration into a raw transcript, forming the backbone of the notebook content.

3.2 Keyframe Detection

To complement the transcript with visual context, we identify keyframes in the video using a Scenedetect "Python-based video processing library". These frames are selected based on scene changes or significant visual transitions, ensuring that only informative frames are retained, rather than every video frame.

3.3 Transcript-Visual Alignment

The transcript and detected keyframes are temporally aligned and grouped. Each segment of the transcript is paired with the corresponding keyframe occurring in the same timeframe. This alignment ensures that both modalities (audio and visual) are captured together.

3.4 Visual Understanding with VLM

Each keyframe is then passed to SmolVLM-256M "a Vision-Language Model (VLM)", which generates natural language descriptions of the visual content. These descriptions capture diagrams, code snippets, slides, or visual demonstrations present in the video.

3.5 Fusion and Cleaning

The transcript and the visual descriptions are then merged and refined using a Qwen2.5-14B "a Large-Language Model (LLM)". At this stage: redundancies are removed, grammar and clarity are improved and the transcript is segmented into coherent sections aligned with the video's instructional flow.

3.6 Notebook Generation

Finally, the cleaned and structured text is passed to the LLM with instructions to transform the material into a Jupyter notebook format. This includes producing a clear introduction to the subject, generating explanatory text cells and creating executable code cells.

4 Evaluation

The system is able to produce logical, beneficial and runnable Jupyter notebooks. This satisfies our objective. However, since our project is novel, it was challenging to come up with formal benchmarks. And for that reason, we had to be creative and rely on special metrics.

4.1 Criteria

As previously mentioned, the novelty of this project idea came with an evaluation challenge. And for this reason, we had to be creative and use a special metric. This metric is **Judges Scores** inspired by "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena"[2] in which we use different LLMs to evaluate our model output based on the rubric we provide as the criteria we try to measure. The rubric consists of the following:

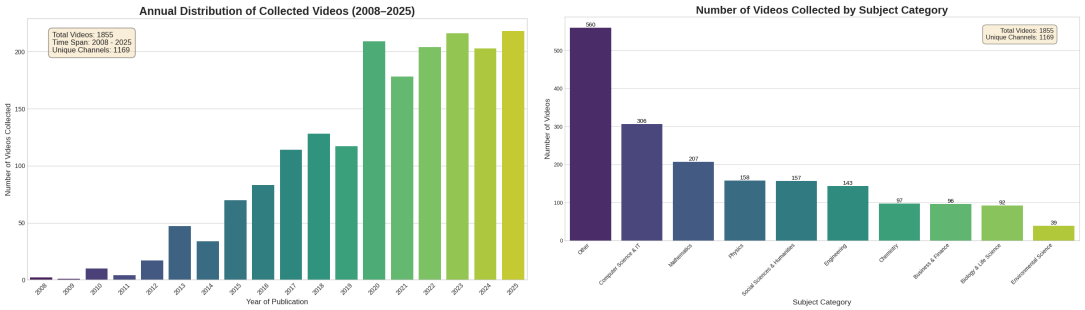
Instructional Clarity: How clear, well-explained and easy to understand the markdown cells are and if complex topics are broken down effectively.

Code Quality: if the code performs the task described in the context. And if the code is well-commented and clean.

Relevance and Completeness: if the generated Jupyter Notebook accurately reflect the content and concepts presented in the video context. And if any key concepts or code snippets are missing.

4.2 Dataset

We needed a dataset to serve as a benchmark for evaluating the performance of the pipeline. For this reason, we gathered a dataset consisting of 1,855 YouTube videos, each represented with rich metadata. For every video, the dataset includes its title, description, and URL, as well as duration (both in minutes and categorized form), publication date, and engagement metrics such as view count and like count. It also records the channel title and the subject of the video. This dataset could be beneficial to many other projects. However, what we are really interested in is the videos.



(a) Dataset Annual Distribution (b) Dataset Subject Distribution

Figure 2: Dataset distributions

4.3 Results

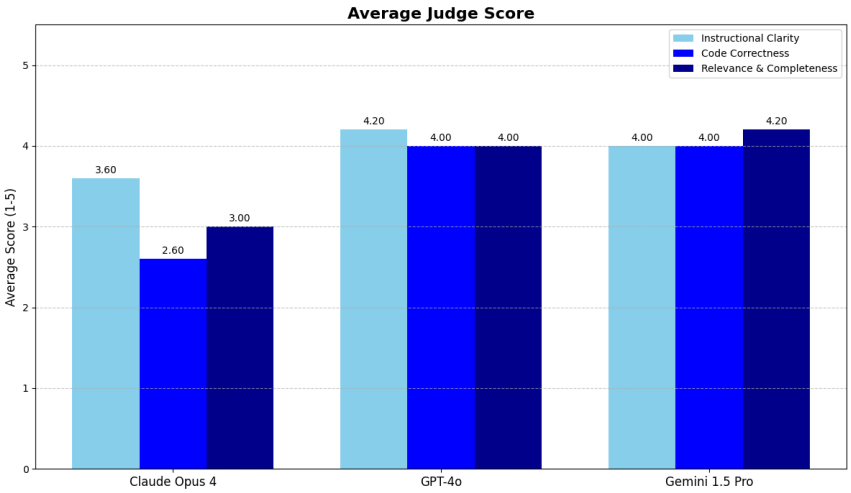


Figure 3: Average of Judges Scores

This shows that the pipeline is producing quality Jupyter Notebooks and satisfying the criteria we have previously defind to a good degree. These results were obtained by sampling from our dataset as input for the pipeline to generate Jupyter notebooks. After that, we evaluate each notebook using the judges (Claude Opus 4, GPT-4o and Gemini 1.5 pro), and then we average the scores for each judge.

5 Challenges and Obstacles

Many challenges were encountered during this project. The most critical ones were the lack of formal benchmarks, which meant that assessing model performance was challenging and required qualitative outcomes or creative metrics. Another challenge was the lack of a reliable dataset. Both of these challenges arise from the novelty of the project idea. Also, the lack of proper hardware limited our options of models to rely on, decreasing the quality of the output. Another challenge was coordination and project management given the limited time and busy schedule.

6 Final Product

The final result of our project is a fully functional web application. In this app, users can input the URL of any YouTube video they wish to transform. The web app sends the URL to a backend server that hosts the pipeline to generate the Jupyter notebook. Once the notebook is generated, the server sends it back to the web app, which automatically triggers a download to the user's local disk.

To optimize efficiency, the system also caches generated notebooks in a MySQL database. This means that if the same video is requested again, the notebook can be retrieved directly from the cache without regenerating it, significantly reducing computation time and resource usage.

7 Conclusion and Future Work

This work demonstrates the feasibility of transforming videos into structured educational Jupyter notebooks, providing an initial proof of concept for automated educational content generation. While promising, the system leaves room for improvement and expansion. For example, enhancing individual pipeline components—such as adapting the ASR model to better capture technical jargon and improving notebook generation with richer formatting and interactivity—will increase usability. Further, exploring alternative architectures that better integrate multimodal information and optimizing the system to reduce inference time are important directions. Ultimately, the development of formal benchmarks will be essential for systematic evaluation and for driving progress in this emerging area.

8 References

[1] Yu, Mengxia, Bang Nguyen, Olivia Zino, Meng Jiang. "Context Selection and Rewriting for Video-based Educational Question Generation." Last modified 2025. arXiv:2504.19406. https://arxiv.org/html/2504.19406v2

[2] Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, Ion Stoica. "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena." arXiv:2306.05685, 2023. https://arxiv.org/abs/2306.05685