

LearnLens - Video Summarization System

Divya Muluk¹, Supriya Gaikwad², Prathamesh Wadaskar³, Ashutosh Yadav⁴, Prof. Jayashri Mankar⁵

^{1,2,3,4}Student, Department Of Computer Engineering, Alard College Of Engineering And Management, Pune

⁵Professor, Department Of Computer Engineering, Alard College Of Engineering And Management, Pune

Abstract - The exponential growth of video content on the internet presents a significant challenge for users seeking to extract meaningful information efficiently. This paper presents LearnLens, a full-stack intelligent web application built on the FastAPI framework [11] that automates video transcription, abstractive summarization, sentiment analysis, concept mind map generation, prerequisite detection, structured Markdown note generation, and interactive transcript-grounded question answering. LearnLens integrates the faster-whisper implementation of OpenAI Whisper for automatic speech recognition, Groq-hosted large language models (LLaMA 3.3 70B Versatile) [13] via the LiteLLM unified multi-provider interface for natural language generation, and Sentence Transformers for semantic accuracy evaluation. The system accepts both YouTube URLs and MP4 video uploads, processes them through a multi-stage AI pipeline, and presents results through an HTML interface featuring an interactive Canvas-based concept mind map, structured Markdown study notes, prerequisite analysis tables, sentiment dashboards, and a real-time streaming Q&A chatbot. Evaluation across a diverse corpus of educational and informational videos demonstrates accuracy scores consistently above 65% using cosine semantic similarity, with end-to-end processing of a ten-minute video achievable in under three minutes on CPU-only hardware. LearnLens represents a deployable and original contribution to automated video comprehension and knowledge extraction.

Key Words: Video summarization, automatic speech recognition, large language models, FastAPI, Whisper, LiteLLM, Groq, concept mind map, sentiment analysis, prerequisite detection, streaming question answering, natural language processing.

1. INTRODUCTION

The digital age has witnessed an unprecedented surge in video content consumption. Platforms such as YouTube host over 800 million videos, with approximately 500 hours of new content uploaded every minute [1]. For students, researchers, and professionals, manually watching and comprehending lengthy educational video content is both time-consuming and cognitively demanding. The need for automated systems capable of distilling essential information from video has therefore become increasingly urgent.

Traditional approaches to video summarization have relied on extractive methods that select representative frames or transcript segments. With the advent of powerful large language models (LLMs), it has become possible to perform abstractive summarization that generates fluent, coherent summaries capturing the semantic essence of content rather than merely reproducing fragments. Simultaneously, automatic speech recognition (ASR) technology has matured to the point where high-accuracy transcription is achievable on commodity CPU-only hardware without GPU acceleration.

This paper presents LearnLens, an end-to-end intelligent video analysis system combining these advances into a unified, deployable web application. LearnLens accepts video content from two sources—YouTube URLs and direct MP4 uploads—and processes them through a multi-stage pipeline producing: (1) a high-quality abstractive summary in up to seven languages, (2) sentiment and topic analysis, (3) an interactive concept mind map, (4) a prerequisite knowledge assessment, (5) structured Markdown study notes downloadable as .md files, and (6) a real-time streaming transcript-grounded Q&A chatbot. The system runs efficiently on CPU-only hardware, making it accessible without expensive cloud GPU infrastructure.

The primary contributions of this work are: (a) a novel multi-stage AI pipeline integrating ASR, LLM-based analysis, and interactive Canvas visualization in a single cohesive web application; (b) a prerequisite detection module that automatically identifies prior knowledge requirements, learning outcomes, and estimated study time; (c) an interactive Canvas-based concept mind map renderer with zoom, pan, and click-to-explore functionality implemented without external charting libraries; (d) a unified LLM interface via LiteLLM supporting automatic fallback across Groq, Gemini, and Ollama providers; (e) a real-time streaming Q&A chatbot using FastAPI StreamingResponse; and (f) rigorous evaluation using cosine semantic similarity via Sentence Transformers.

2. RELATED WORK

A. Automatic Speech Recognition

Automatic Speech Recognition has undergone a transformative shift with end-to-end neural architectures. The Whisper model introduced by Radford et al. [2] represents a significant milestone: trained on 680,000 hours of multilingual audio data, it achieves near-human transcription accuracy across diverse

languages and acoustic conditions. LearnLens uses the faster-whisper reimplementation, which exploits CTranslate2 [3] to achieve approximately four times the throughput of the original Whisper on CPU, enabling practical deployment without GPU hardware.

B. Text Summarization

Text summarization approaches are broadly categorized as extractive or abstractive. Extractive methods such as TextRank and LexRank score and select existing sentences from the source document. Abstractive summarization, pioneered by BART [4] and T5 [5], generates novel text. The emergence of instruction-tuned LLMs such as GPT-4 [6], LLaMA, and Mistral has further raised the quality ceiling. LearnLens leverages LLaMA 3.3 70B Versatile [15] via the Groq inference API [13], achieving abstractive summarization with latency typically between two and four seconds per request due to Groq's Language Processing Unit hardware.

C. Video Understanding Systems

Prior systems for video content analysis have taken various approaches. VideoXum [7] addresses joint video and text summarization through cross-modal representations. More recent works such as VideoChat and Video-LLaMA extend multimodal language models to process video frames directly. LearnLens takes a deliberately practical approach: by extracting the audio channel and transcribing it, the system operates in the text domain, bypassing substantial computational cost while remaining effective for educational content where spoken language carries the primary information load.

D. Mind Map Generation and Question Answering

Automated construction of knowledge graphs from unstructured text has been extensively studied, with systems such as REBEL and OpenIE extracting subject-relation-object triples. Mind map generation from text remains underexplored. LearnLens's concept mind map module prompts an LLM to produce structured JSON describing a topic hierarchy, then renders this as an interactive Canvas visualization with Bezier-curved edges implemented natively in the browser. For question answering, Retrieval-Augmented Generation (RAG) [8] has emerged as the dominant paradigm; LearnLens implements a simplified context-window approach, passing the full transcript with each question and streaming answers token-by-token via FastAPI StreamingResponse [11].

3. METHODOLOGY

A. System Architecture

LearnLens is architected as a multi-layer web application following a service-oriented design pattern, divided into four primary layers: the Presentation Layer (Jinja2 HTML/JavaScript templates), the Application Layer (FastAPI

routers [11]), the Service Layer (AI pipeline components), and the Data Layer (SQLAlchemy with SQLite [14]). Four FastAPI routers handle distinct concerns: VideoRouter for video processing, FeedbackRouter for user feedback submission, ChatRouter for streaming Q&A requests, and NotesRouter for Markdown notes download. Database tables for VideoUrl, VideoMp4, and FeedBack records are created automatically on application startup via SQLAlchemy's create_all() mechanism.

B. Input Processing Pipeline

LearnLens accepts two input types: YouTube video URLs and local MP4 file uploads. For YouTube URLs, yt-dlp [12] downloads the audio stream in WAV format. For MP4 uploads, moviepy extracts the audio channel. Both pipelines converge at the transcription stage, where the faster-whisper model (base configuration, int8 quantization) is loaded as a singleton to avoid repeated initialization overhead across requests. Transcription is performed on CPU using CTranslate2's optimized inference runtime, yielding text segments concatenated into a full transcript string.

C. LiteLLM Multi-Provider Interface

A significant architectural contribution of LearnLens is the adoption of LiteLLM [9] as a unified LLM interface layer. Previous iterations of the system were tightly coupled to the Groq API client [13], meaning any provider outage or rate-limit event caused complete pipeline failure. LiteLLM resolves this by providing a single OpenAI-compatible interface routing requests to over 100 LLM providers including Groq, Google Gemini, Anthropic Claude, OpenAI, and locally-hosted Ollama models. The fallback chain is configured as: primary (Groq LLaMA 3.3 70B Versatile) then Gemini 2.0 Flash then local Ollama model, with three automatic retries before each fallback. LiteLLM also exposes response cost tracking, enabling monitoring of API expenditure per request.

D. AI Analysis Pipeline Modules

Once transcription is complete, the transcript is passed through five parallel LLM-based analysis modules, all powered via LiteLLM. The Summarization Module prompts the LLM to produce a 3–4 sentence abstractive summary in the user-selected output language, supporting English, Hindi, Spanish, French, German, Japanese, and Arabic via a two-step summarize-then-translate approach. The Sentiment and Topic Analysis Module returns a structured JSON object containing overall sentiment classification, a sentiment intensity score from 0 to 100, an emotional tone descriptor, five key topic tags, and three key insights. The Concept Mind Map Extraction Module produces a hierarchical JSON with a central topic, up to five main branches, and up to three subtopics per branch, rendered by a custom Canvas-based renderer featuring Bezier curve connections, color-coded nodes, and interactive zoom (0.3x to 4x), pan, and click-to-explore behavior. The Prerequisite Detection Module identifies prerequisite knowledge, learning outcomes, suggested follow-up topics,

difficulty level, and estimated study time. The Streaming Q&A Module accepts user questions via asynchronous fetch requests, streaming answers token-by-token using FastAPI StreamingResponse and LiteLLM’s stream=True interface, consumed by the JavaScript frontend via ReadableStream.

E. User Interface and Output Visualization

The LearnLens interface provides users with multiple forms of knowledge extraction from video content. After transcription and summarization, the system generates prerequisite information, sentiment analysis, concept mind maps, downloadable notes, and transcript-grounded question answering. Figures 1–3 illustrate the major user interface components of the system.



Figure 1: LearnLens Dashboard Showing Summary, Sentiment Analysis and Prerequisite Detection



Figure 2: Interactive Concept Mind Map Generated by LearnLens



Figure 3: Transcript-Grounded Question Answering and Markdown Notes Generation

F. Accuracy Evaluation and Technology Stack

Summary quality is quantified using semantic cosine similarity computed by the paraphrase-MiniLM-L6-v2 Sentence Transformer model [10]. This model encodes both the original transcript and the generated summary into dense 384-dimensional vector representations; cosine similarity between them is reported as an accuracy percentage. This metric captures semantic faithfulness more accurately than lexical overlap metrics such as ROUGE, which penalize legitimate paraphrasing. All LLM modules implement try/except wrappers with graceful default fallbacks to ensure the application never crashes due to an unexpected model output format.

Table I summarizes the complete technology stack used in LearnLens.

Component	Technology	Role
Web Framework	FastAPI 0.115 [11]	Async HTTP routing
Template Engine	Jinja2	HTML rendering & UI templates
ASR Engine	faster-whisper	CPU-optimized ASR
LLM Interface	LiteLLM [9]	Multi-provider fallback
Primary LLM	LLaMA 3.3 70B / Groq [13][15]	LPU inference
Semantic Similarity	Sentence Transformers [10]	Accuracy scoring
Database ORM	SQLAlchemy + SQLite [14]	Data persistence

Audio/Video	yt-dlp [12] + moviepy	Download and extraction
PDF Export	ReportLab 4.2	Summary PDF download

Table I: LearnLens Technology Stack

4. RESULTS AND DISCUSSION

A. Experimental Setup

LearnLens was evaluated on a corpus of 20 videos spanning five categories: educational lectures, programming tutorials, news reports, product reviews, and motivational talks. Video durations ranged from 2 to 15 minutes. All experiments were conducted on a machine with an Intel Core i5 processor, 8 GB RAM, and no GPU, running Ubuntu 24.04 Linux. The faster-whisper base model with int8 quantization was used for all transcriptions.

B. Transcription Quality

Transcription accuracy was assessed by computing Word Error Rate (WER) against manually prepared ground-truth transcripts for a subset of ten English videos. The mean WER was 8.3%, consistent with published results for Whisper base on English speech [2]. Transcription throughput scaled approximately linearly with audio duration at a mean ratio of 1:4—one second of audio transcribed per four seconds of wall-clock time.

C. Summarization Accuracy

Summary quality was evaluated using cosine semantic similarity. Table II presents results by video category. The overall mean accuracy of 70.1% demonstrates that generated summaries reliably capture the semantic essence of original transcripts. News reports achieved the highest accuracy (73.5%), likely because news content follows a structured, factual style that is easier for the model to condense faithfully. Product reviews showed the lowest accuracy (66.4%), potentially due to the conversational and opinion-laden nature of such content.

Category	Avg (%)	Min (%)	Max (%)
Educational Lectures	71.2	65.8	78.4
Programming Tutorials	68.9	62.1	74.7
News Reports	73.5	69.2	81.3
Product Reviews	66.4	58.9	72.1
Motivational Talks	69.8	64.0	75.6

Overall	70.1	58.9	81.3
---------	------	------	------

Table II: Summarization Accuracy By Video Category

D. End-to-End Processing Time

Table III shows mean processing time by pipeline stage. Whisper transcription dominates total processing time, accounting for 60 to 78 percent of end-to-end latency. LLM API calls via Groq collectively contribute under 15 seconds regardless of video length, owing to Groq's LPU-based inference acceleration. The complete pipeline processes a ten-minute video in approximately 174 seconds on CPU-only hardware, which is acceptable for the asynchronous nature of the task.

Pipeline Stage	2 min	5 min	10 min
Audio extraction	8 s	14 s	22 s
Whisper transcription	28 s	68 s	135 s
All LLM modules (total)	9 s	11 s	14 s
Accuracy scoring	1 s	2 s	3 s
Total	46 s	95 s	174 s

Table III: Processing Time By Pipeline Stage (Cpu-Only)

E. LiteLLM Fallback Resilience

The LiteLLM fallback architecture was validated by simulating Groq rate-limit scenarios. When the primary provider returned an error, LiteLLM automatically retried three times before switching to the Gemini 2.0 Flash fallback, adding an average overhead of 1.8 seconds. User-facing error rates decreased from approximately 12% (direct Groq client, no retry) to under 2% with the LiteLLM fallback chain enabled, demonstrating substantial resilience improvement suitable for production deployment.

F. Qualitative Assessment of Advanced Features

The prerequisite detection module was qualitatively assessed by three domain experts who verified generated prerequisite lists against their knowledge of the video content. In 16 of 20 cases (80%), the generated prerequisites were judged accurate and relevant. The concept mind map was assessed for structural coherence; in 18 of 20 cases (90%), the generated hierarchy was judged to reflect the actual conceptual structure of the video. The streaming Q&A module was assessed on 50 questions across the evaluation set; 82% of answers were judged accurate and grounded in the transcript, with the

remainder either partially correct or correctly declining to answer when information was absent from the video.

5. CONCLUSION

This paper has presented LearnLens, an intelligent video summarization and analysis system integrating automatic speech recognition, large language model inference via a unified multi-provider interface, and interactive visualization into a deployable web application. The system demonstrates that high-quality video comprehension tools can be built using freely available AI components, making them accessible to students and institutions without expensive cloud GPU infrastructure.

The key innovation lies in the integration of multiple AI capabilities into a coherent user experience: from submitting a YouTube URL to receiving an abstractive summary in up to seven languages, sentiment and topic analysis, an interactive Canvas concept mind map, a prerequisite assessment table, downloadable Markdown study notes, and a real-time streaming transcript-grounded chatbot—all within a single web session. The adoption of LiteLLM reduced user-facing error rates from approximately 12% to under 2% through automatic retry and fallback across Groq, Gemini, and Ollama providers.

Evaluation on a corpus of 20 videos demonstrates a mean summarization accuracy of 70.1% and end-to-end processing under three minutes for ten-minute videos on CPU-only hardware. Future enhancements will include multimodal frame analysis, an agentic AI architecture for multi-step learning path generation, vector-based RAG for long-form content, and quiz generation for interactive learning assessment.

REFERENCES

- [1] YouTube Press, "YouTube by the Numbers," Google LLC, 2024. Available: <https://www.youtube.com/about/press/>
- [2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv preprint arXiv:2212.04356, 2022.
- [3] G. Klein, F. Hernandez, T. Nguyen, and J. Senellart, "CTranslate2: Efficient Inference with Transformer Models," arXiv preprint arXiv:2003.02030, 2020.
- [4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proc. ACL, 2020.
- [5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.
- [6] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.
- [7] J. Lin, R. Ding, J. Liang, and X. Liu, "VideoXum: Cross-modal Visual and Textual Summarization of Videos," IEEE Transactions on Multimedia, 2023.
- [8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.
- [9] BerriAI, "LiteLLM: Call all LLM APIs using the OpenAI format," GitHub repository, 2024. Available: <https://github.com/BerriAI/litellm>
- [10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, 2019.
- [11] FastAPI, "FastAPI Documentation," 2024. Available: <https://fastapi.tiangolo.com/>
- [12] yt-dlp contributors, "yt-dlp: A youtube-dl fork with additional features and fixes," GitHub repository, 2024. Available: <https://github.com/yt-dlp/yt-dlp>
- [13] Groq Inc., "Groq Documentation and API Reference," 2024. Available: <https://console.groq.com/docs/>
- [14] SQLAlchemy, "SQLAlchemy Documentation," 2024. Available: <https://docs.sqlalchemy.org/>
- [15] AI@Meta, "The Llama 3 Herd of Models," arXiv preprint arXiv:2407.21783, 2024.