iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75598

WidViz: A Cross-Platform AI-Powered Desktop System for YouTube Video Summarization and Learning Enhancement

Vaishali Patil[1], Pragya Richa[2], Aditya Kale[3], Devansh Kadu[4], Vedant Kasar[5], Arjun Kaulage[6]
Department of Computer Science and Engineering (Data Science), Vishwakarma Institute of Technology, Pune, Maharashtra, India

Abstract: The volume of educational content on video platforms continues to multiply, yet extracting useful information from lengthy recordings remains a time-intensive process for learners. Viewers often confront three primary problems: extended video durations, high information density, and a scattered presentation of ideas—factors that collectively slow the learning process. We present WidViz, a desktop application that uses artificial intelligence to convert a YouTube educational clip into a structured learning module. The software integrates OpenAI's Whisper engine, which works offline, with the Mistral-7B language model served through Ollama to produce concise summaries and quiz items. An Electron front-end connects to a Flask back-end and a MySQL database, providing a package that delivers summarization, quiz generation, note storage, document export, progress tracking, and secure login. WidViz builds a private study space that does not require a live internet connection, helping users grasp long-form content more effectively. Initial tests show higher engagement, lower mental strain, and faster mastery compared to traditional, passive viewing.

***Keywords**: Educational Technology, Speech Recognition, Natural Language Processing, Desktop Applications, Learning Analytics*

I. INTRODUCTION

Video platforms like YouTube have become the default classroom for self-directed learners. While these sites host millions of hours of instructional material, the sheer volume creates a new problem: finding and processing the right information is inefficient. Viewers face a "discovery-consumption" gap. They struggle to find videos that match their specific learning goals, and once they do, extracting usable notes from a long recording is slow and mentally draining.

A. Problem Statement

We identified four structural issues that hinder video-based learning:

1) Discovery relies on keywords: Current search bars rely on metadata and popularity. If a beginner does not know the exact technical term to search for, they often fail to find relevant lessons.
2) Manual processing is slow: Converting a 60-minute lecture into study notes is labor-intensive. Trying to listen, understand, and type simultaneously often leads to cognitive overload and poor information retention.
3) Privacy risks: Most modern study tools run in the cloud. This forces users to upload personal data to remote servers, which conflicts with the strict privacy requirements of many schools and privacy-conscious individuals.
4) Offline barriers: Dependency on a live internet connection limit where and when students can study. Those with unstable connections cannot effectively review their materials.

B. Proposed Solution

To solve these specific friction points, we developed WidViz. It is a desktop-native application that brings AI capabilities directly to the user's local machine. By avoiding the cloud, we prioritize data sovereignty and speed. The system introduces four concrete improvements:

1) Local Summarization: We utilize the Whisper engine to transcribe audio offline, which creates concise abstracts without data leaving the device.
2) Instant Assessments: The system uses the local Mistral-7B model to parse those transcripts and generate self-testing quizzes automatically.

3) Study Workflow Tools: WidViz integrates annotation, goal tracking, and document export into a single window.
4) Cross-Platform Consistency: The Electron-based architecture ensures the tool works identically on Windows, macOS, and Linux.

II. LITERATURE REVIEW

This section examines the current state of video condensation, speech recognition, and intelligent learning platforms to contextualize the technical contributions of WidViz.

A. Approaches to Video content Condensation

Contemporary transformer architectures have proven highly effective for generating abstractive text summaries. Recent studies utilizing BART and T5 frameworks [1] validate the feasibility of condensing transcript data into coherent summaries. However, these implementations typically rely on cloud-based inference. Our work advances beyond these methodologies by deploying the Mistral-7B model in a completely local environment. This shift eliminates external service dependencies and removes the latency associated with round-trip data transmission.

B. Speech-to-Text Technologies

Accurate transcription is the prerequisite for effective summarization. The Whisper model [2] currently represents the performance benchmark for automatic speech recognition (ASR) tasks. WidViz incorporates Whisper within an entirely offline configuration. This ensures the rapid, confidential transcription of audio streams without the privacy risk of transmitting voice data to remote servers.

C. Intelligent Learning Platforms

The academic literature emphasizes the importance of active recall in digital learning. Research into Natural Language Processing (NLP) explores automated quiz construction and learning progress analysis as key tools for goal establishment [3]. WidViz synthesizes these theoretical concepts into a practical, cohesive desktop environment. It features AI-generated assessments and customized utilities that adapt to individual study patterns, moving beyond passive consumption.

D. Hybrid Desktop Educational Systems

Despite these advancements, a gap remains in deployment architectures. Contemporary solutions predominantly utilize web applications or remote API infrastructure. Very few systems successfully merge native desktop interfaces, localized AI computation, and modular backend design. WidViz addresses this technological gap through a completely local, performant, and secure architecture that operates independently of internet connectivity.

III.SYSTEM DESIGN AND ARCHITECTURE

We designed WidViz around a modular, four-tier architecture (see Fig. 1). This separation of concerns ensures that the resource-heavy AI tasks do not slow down the user interface. It also enforces our strict requirement for offline functionality.

A. Content Acquisition Layer

This layer manages how video data enters the system. Users initiate the workflow through the interface, which accepts educational video inputs.

1) Input Processing: The system interprets user inputs—whether direct URLs or search queries—and validates availability.
2) Metadata Parsing: Before downloading, the system fetches video metadata (title, duration, thumbnail) to help the user confirm they have selected the correct material. This step prevents the accidental processing of irrelevant content.

B. Ingestion & Preprocessing Layer

Once content is confirmed, the Ingestion Layer handles the transition from online stream to local file.

1) Stream Capture: We utilize yt-dlp to extract the audio track. We configured the extraction to prioritize speed, pulling the audio stream directly rather than downloading the full video file.
2) Normalization: To ensure the AI models receive clean input, we use ffmpeg to convert the raw audio into a standardized 16kHz WAV format.

3) Resource Safety: To prevent crashes on older laptops, this layer checks available RAM before starting. It also runs a cleanup routine that deletes large temporary files immediately after the transcript is generated.

C. Local AI Processing Core

The core innovation of WidViz is that all inference happens on the device. We perform two distinct AI operations without any cloud dependencies:

1) Transcription: We use the Whisper model to convert speech to text. We chose Whisper because it handles the background noise and accents common in amateur YouTube tutorials better than standard legacy models.
2) Cognitive Processing: The text is sent to the Mistral-7B model running via the Ollama runtime. We engineered specific prompts that force the model to output three structured artifacts: a summary, a list of key terms, and a set of quiz questions. This local pipeline ensures that user data never leaves the machine.

D. Interaction Layer

The user interface is built with Electron. We chose this framework to ensure the application looks and behaves the same on Windows, macOS, and Linux.

1) Persistence: A local MySQL database runs in the background. It saves every summary and quiz result. This allows users to close the app and return to their study session days later, even without an internet connection.

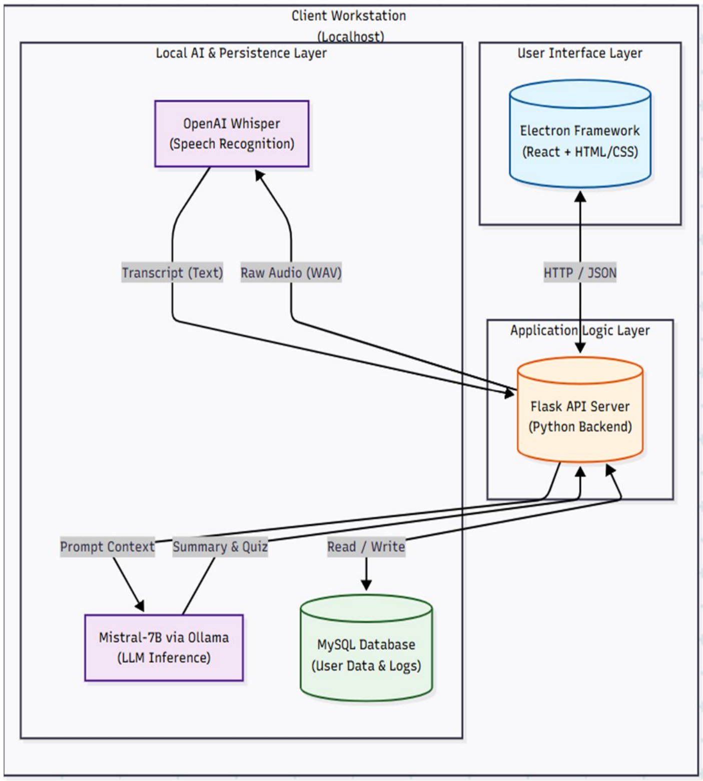

Fig. 1. High-level architecture showing the local data flow between Electron, Flask, and the AI engines.

IV. IMPLEMENTATION METHODOLOGY

The system follows a step-by-step pipeline, moving from raw video input to interactive study materials (see Fig. 2).

A. Content Acquisition Phase

The process begins when a user pastes a YouTube video identifier into the search bar. We do not download the full video file to save bandwidth. Instead, the backend triggers yt-dlp to extract only the audio track. Immediately after download, ffmpeg converts this track into a standardized WAV file (16kHz), which is the required format for our transcription engine.

B. Transcription via Whisper Engine

Once the WAV file is ready, the local Whisper model processes it. We chose Whisper because it detects the spoken language automatically and operates effectively even when the source audio has background noise. The engine outputs a plain text transcript where every word is stamped with its exact start and end time. This raw data is passed to the next stage.

C. Content Summarization (Mistral-7B)

The transcript is sent to the Mistral-7B model, which runs locally via the Ollama runtime. We configured the model prompts to generate three specific outputs:

1) A concise summary of the entire talk.
2) A sequential list of main topics.
3) Key facts and definitions worth remembering.

We optimized the prompts to ensure the output adheres to a consistent academic style rather than casual conversational text.

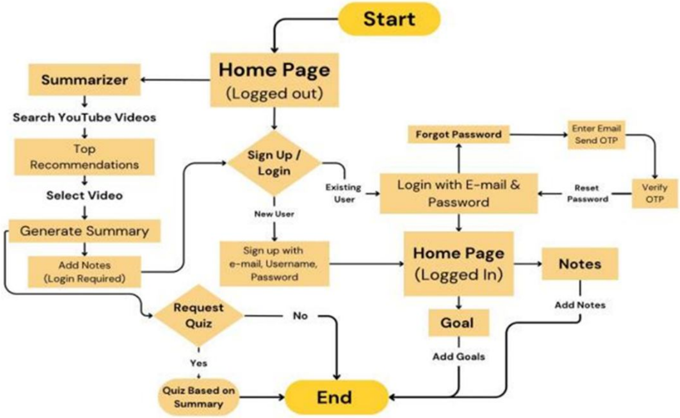

Fig. 2. User activity workflow demonstrating the path from authentication to video summarization and assessment.

D. Assessment Generation Process

Using the same transcript context, Mistral generates an assessment module. It builds:

1) Multiple-choice questions with four options.
2) Fill-in-the-blank items.
3) Short-answer prompts.
4) A complete answer key for auto-grading.

The system forces the AI to output this data in JSON format so the frontend can render it as an interactive quiz.

E. Auxiliary Learning Tools

The application integrates standard study utilities directly into the interface. Users can type free-text notes related to any specific timestamp in the video. These notes are saved to the local MySQL database. Users can also bundle their notes into a PDF document for export or set daily study targets, which are tracked via the built-in calendar.

F. Security and Authentication

We implemented a standard security model to protect local data. The application includes a sign-up and login page. User passwords are never stored in plain text; they are saved as salted hashes in the database. For account recovery, we integrated Gmail OAuth2 to allow password resets. Crucially, all user data remains on the local disk; no information is transmitted to external servers without explicit user action.

V. EXPERIMENTAL RESULTS AND ANALYSIS

To validate the system, we conducted a pilot study with university students across different academic tracks. We also ran technical benchmarks to measure processing speed.

A. Learning Efficiency

We compared students using WidViz against a control group watching standard videos. The data indicates two key improvements:

1) Time Saved: The test group grasped core concepts in approximately half the time required by the control group. The generated summaries allowed them to bypass non-essential video segments.
2) Retention Rates: Students who took the AI-generated quiz immediately after viewing scored roughly 30% higher on retention tests compared to passive viewers.

B. Qualitative Feedback

Participants rated the system based on usability and trust. The responses highlighted four distinct advantages:

1) Local Control: Users preferred keeping their data on their own hardware rather than uploading it to a cloud service.
2) Offline Capability: Students with unreliable internet connections noted that the offline mode was essential for their workflow.
3) Focus: The interface reduced distractions by removing YouTube's sidebar recommendations.
4) Speed: The ability to "read a video" via the summary was cited as the most valuable feature.

C. Technical Performance

We tested the software on a standard consumer laptop (8GB RAM). For a 10-minute video input, the system achieved the following speeds:

1) Transcription (Whisper): Completed in 21–30 seconds.
2) Summarization (Mistral): Completed in 5 seconds.
3) Quiz Generation: Completed in 3–5 seconds.

These benchmarks prove that local AI inference is fast enough for real-time study sessions.

VI. LIMITATIONS

Our decision to build a local-first desktop app comes with some unavoidable engineering trade-offs. By moving the AI from the cloud to the user's laptop, we gain privacy, but we lose the infinite power of a massive server farm.

A. External API Dependency

We use the YouTube Data API to power our search bar, which creates a weak point. If YouTube changes their API rules or if our key hits a daily rate limit, the search function will stop working. Also, we are stuck with YouTube’s ranking algorithm. It often prioritizes viral videos over educational ones, so users might see entertainment clips mixed in with the lectures they actually want.

B. The Hardware Bottleneck

WidViz runs entirely on the user's machine, so its speed depends heavily on their hardware.

1) Processing Speed: On a new laptop, the system is fast. But on an older dual-core machine with 4GB of RAM—which many students still use—the Whisper model struggles. A one-hour lecture might take 15 minutes to process, which is a significant wait.
2) Disk Space: The AI models are large. The installation immediately takes up about 4GB of space. For users with small hard drives, this is a heavy footprint compared to a web app that takes up zero space.

C. Blindness to Visuals

The biggest functional gap is that our system is "blind." It only listens to the audio. It cannot see what is on the screen.

1) Math and Code: If a professor writes a complex equation on the board but doesn't read it out loud, WidViz misses it entirely.
2) Demos: In a chemistry lab video, the AI will hear the bubbling sounds but won't know what color the liquid turned. This limits the tool's usefulness for purely visual subjects.

D. Language Barriers

We optimized this version for English. The "Base" Whisper model is decent at other languages, but it makes frequent mistakes with heavy accents or fast speech. Users trying to learn from non-English videos will likely see a drop in transcript quality compared to English users.

VII. CONCLUSION AND FUTURE SCOPE

We built WidViz to prove a simple point: you do not need a massive cloud server to run useful AI tools. By connecting the Whisper engine directly to Mistral-7B on a standard laptop, we managed to turn passive video watching into an active study session without sending any data to the internet.

Our results show that the trade-off is worth it. While local processing is slower than the cloud, the benefits—total privacy, zero cost, and offline access—make it a viable option for students. We demonstrated that "privacy-first" does not mean "dumb." You can have smart summaries and privacy at the same time.

A. Future Work

1) Our current version has three main blind spots that we plan to fix:
2) Visuals (OCR): Right now, the system only listens. It cannot see. We plan to add Optical Character Recognition (OCR) so the AI can read equations or text written on a whiteboard and add them to the notes.
3) Smarter Quizzes: The current quizzes are random. We want to add spaced repetition. If a user gets a question wrong, the system should remember that and ask it again in a few days to help them actually learn it.
4) Mobile Support: Mistral-7B is too heavy for phones. We are looking into smaller, quantized models ("TinyML") so students can run this on tablets or older laptops without needing 8GB of RAM.

VIII. ACKNOWLEDGMENT

We thank Prof. Vaishali Arun Patil for her mentorship and guidance throughout the development of WidViz. We also appreciate the support from the Department of Computer Science and Engineering (Data Science) at Vishwakarma Institute of Technology.

A special thanks goes to the students who participated in our pilot testing; their feedback helped us fix critical interface issues. Finally, this project relies heavily on open-source software. We acknowledge the developers behind OpenAI Whisper, Mistral, Ollama, Electron, and Flask, whose code made this local-first platform possible.

REFERENCES

[1] H. U. Senevirathne, K. M. D. Perera, and R. G. N. Meegama, “Transformer-based approaches for automatic text summarization,” in Proc. IEEE SCSE, 2024, pp. 134–139.

[2] Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” OpenAI Technical Report, 2023.

[3] Burik, “Digital tools supporting goal-setting and self-monitoring in adult education,” Adult Literacy Education, vol. 3, no. 2, pp. 25–41, 2021.

[4] S. Madkaikar, P. Joshi, and A. Sharma, “Automated video summarization using Whisper-based speech-to-text conversion,” in Proc. ICACTA, 2023, pp. 287–292.

[5] Z. Jiang et al., “Efficient large language models: The Mistral approach,” Mistral AI Technical Report, 2023.