International Journal of Information Technology & Computer Engineering

ISSN 2347-3657

Volume 13, Issue 3, 2025

Vocal Gist

[1]Ishrath Nousheen, [2]Yagandla Roshini, [3]Jukuri Srinija, [4]Polasani Sruthi

[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women

[2.3.4]B,tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women

psruthirao114@gmail.com

ABSTRACT

This project introduces "Vocal Gist," a web-based application designed to streamline the process of extracting and synthesizing information from YouTube videos. Addressing the challenge of time-consuming video consumption and the difficulty in quickly grasping key content, Vocal Gist provides an efficient solution for users seeking concise insights.

The application functions by first extracting the full transcript from a given YouTube video URL using the youtube_transcript_api. This raw text is then processed by a Google Generative AI model (gemini-2.0-flash), which intelligently summarizes the content into detailed, bullet-pointed notes, typically within 250 words. Furthermore, to enhance accessibility and usability, Vocal Gist offers the functionality to translate these generated summaries into various languages and allows users to download the notes as a plain text file for offline reference or further use.

Developed using Python, with Streamlit for the intuitive graphical user interface and google-generativeai for leveraging advanced AI capabilities, Vocal Gist significantly reduces the effort required to glean essential information from video content. It serves as a valuable resource for students, researchers, content creators, and anyone requiring quick, multilingual access to video summaries.

Keywords: YouTube Video Summarization, Transcript Extraction, youtube_transcript_api, Google Generative AI, Gemini 2.0 Flash, AI Text Summarization, Streamlit Web App, Multilingual Translation, Text File

1. INTRODUCTION

In today's digital age, video content has become an indispensable medium for learning, entertainment, and information dissemination. Platforms like YouTube host an unparalleled volume of videos, ranging from educational lectures and detailed tutorials to news analyses and conference talks. While this abundance of visual information is a tremendous resource, it also presents a significant challenge: the sheer volume and length of many videos make it time-consuming for users to efficiently extract key insights and actionable information. Manually sifting through hours of footage or meticulously taking notes during playback is often impractical and inefficient. This is precisely the challenge that **"Vocal Gist"** aims to solve.

Vocal Gist is a powerful yet intuitive web-based application designed to transform lengthy YouTube videos into concise, digestible, and multilingual textual summaries. Its core functionality involves automatically extracting video transcripts, intelligently summarizing these transcripts into bullet-pointed notes, and offering translation capabilities for global accessibility. Additionally, users can easily download these generated notes for offline access or further integration into their workflows. Built on a robust Python backend utilizing Google's Generative AI (gemini-2.0-flash) for sophisticated text processing and a user-friendly front-end powered by Streamlit, Vocal Gist provides a practical solution for students, researchers, content creators, and professionals who need to quickly grasp the essence of video content.

Existing System:

Before the advent of specialized tools like Vocal Gist, the traditional method for extracting information from online video content, particularly on platforms like YouTube, primarily involved manual engagement. Users were required to watch entire videos, often of significant length, and simultaneously take notes. This process is inherently inefficient and time-consuming, leading to challenges such as information overload, difficulty in pinpointing specific details quickly, and the practical impossibility of consuming large volumes of video content for research or learning purposes. Furthermore, language barriers posed a substantial obstacle, as users would struggle to comprehend content presented in languages they did not understand, lacking any integrated translation capabilities for summaries. This manual and disjointed approach often resulted in suboptimal information retention and hindered the rapid assimilation of knowledge from video resources.

Proposed System:

The proposed system, **"Vocal Gist,"** is a web-based application developed to efficiently extract and process information from YouTube videos, directly addressing the limitations of manual methods.[1] It operates by first retrieving the video's transcript via youtube_transcript_api. This transcript is then sent to a Google Generative AI model (gemini-2.0-flash) to generate concise, bullet-pointed summaries. Vocal Gist further enhances utility by providing integrated options for translating these summaries into multiple languages and downloading them as text files. Built with Python and Streamlit, this system offers a streamlined, efficient, and accessible solution for rapid video content analysis.

2.RELATED WORK

The increasing volume of online video content has spurred significant interest in tools that enhance information retrieval and consumption efficiency. Historically, extracting insights from videos primarily involved manual effort, where users would meticulously watch entire videos and take notes, a process both time-consuming and prone to human error. Early technological aids offered basic functionalities like keyword searching within automatically generated captions, which often suffered from accuracy issues. More advanced solutions then emerged focusing on video-to-text conversion, leveraging speech recognition technologies to transcribe audio into raw text. Libraries such as youtube_dl combined with various speech-to-text APIs (e.g., Google Cloud Speech-to-Text) facilitate this, but typically leave the user with the laborious task of sifting through extensive raw transcripts to find relevant information.Concurrently, the field of Natural Language Processing (NLP) has seen the development of numerous algorithms and models for automatic text summarization and translation. Text summarization techniques can be broadly categorized into extractive methods, which select significant sentences from the original text (e.g., using algorithms like TextRank), and abstractive methods, which generate entirely new sentences that capture the essence of the content, often utilizing advanced neural networks like transformer-based models (e.g., BERT, GPT). Libraries such as Sumy, NLTK, and spaCy provide foundational tools for these tasks. Similarly, sophisticated machine translation services (e.g., Google Translate, DeepL) are widely available. [2] **Vocal Gist** distinguishes itself by offering a unified and comprehensive solution that addresses these existing gaps. Unlike systems that merely provide raw transcripts or require users to employ separate tools for summarization and translation, Vocal Gist integrates youtube_transcript_api for reliable transcript retrieval with the advanced gemini-2.0-flash AI model.By consolidating these functionalities,Vocal Gist provides a superior, end-to-end experience that significantly streamlines the process of consuming and understanding video content, thereby outperforming fragmented traditional approaches and most standalone solutions.

3. REQUIREMENT ANALYSIS

Functional Requirements:

Functional requirements define the specific actions or services that the "Vocal Gist" system must perform to meet user needs and system objectives. These requirements detail what the application does at each stage of its operation, from user input to final output, ensuring that all core functionalities are clearly outlined for development and testing.

- YouTube Video URL Input
- Video ID Extraction
- Transcript Retrieval
- Error Handling for Transcript Availability
- Transcript Summarization
- Summary Translation
- Summary Download
- Thumbnail Display
- API Key Management
- User Feedback and Progress Indicators

Non-Functional Requirements:

Non-functional requirements specify quality attributes of the "Vocal Gist" system, defining how well the system performs its functions. These requirements are crucial for ensuring the overall user experience, system reliability, security, and maintainability.

Performance:

- Response Time**:** The system should aim to display summaries and translations within 5-10 seconds of clicking the respective buttons, depending on transcript length and API response times..
- Transcript Processing Speed: Transcript extraction should ideally complete within 3-5 seconds for most videos.

Scalability:

- The system should be capable of handling multiple concurrent users without significant degradation in performance. (Note: Streamlit's inherent single-threaded nature might limit this without deployment considerations, but it's a good NFR for a robust system).

Software Requirements:

Programming Language	:	Python 3.7
IDE	:	Visual Studio Code
Front end Technologies	:	Streamlit
Libraries	:	generativeai,pythondotenv,youtubetranscriptapi
Backend	:	Python, Google Gemini Ai

Hardware Requirements:

Processor	:	Intel core i5
Ram	:	8GB
Hard Disk	:	500GB

4.DESIGN

System Architecture:

The system architecture of **"Vocal Gist"** adheres to a robust client-server model, ensuring an efficient and seamless flow for processing user requests and delivering summarized, translated video notes. At its forefront, the **Presentation Layer** encompasses the user's web browser, which dynamically renders the interactive interface generated by Streamlit. This layer empowers users to input YouTube video URLs, initiate summarization or translation, select desired target languages, and view all displayed outputs, including video thumbnails, directly within their browser. All user-driven interactions are captured at this layer and securely transmitted as requests to the underlying Application Layer for processing.

The **Application Layer** serves as the central processing unit, implemented as a Python-based Streamlit Application running on a dedicated server. This core component is responsible for orchestrating the entire workflow: it receives and validates user requests, accurately extracts video IDs from the provided URLs, and then interacts with the **External Services Layer**. This interaction involves making calls to the youtube_transcript_api for retrieving raw video transcripts from YouTube. Subsequently, the extracted transcript is sent to Google's Generative AI (specifically the gemini-2.0-flash model via the google-generativeai library) for performing intelligent summarization and, upon request, multilingual translation. After processing, the Application Layer manages the internal state of the application and efficiently sends the processed data back to the Presentation Layer for display.

Finally, the **External Services Layer** is crucial, comprising specialized external resources that provide the necessary raw data and advanced AI capabilities. This includes the YouTube Transcript API, which supplies the textual content of videos, and Google Generative AI, which performs the complex summarization and translation operations. This integrated, multi-layered architecture ensures modularity, efficiency, and clear separation of concerns, providing a highly responsive and user-friendly experience by distributing tasks appropriately between the client, the application server, and external cloud services.

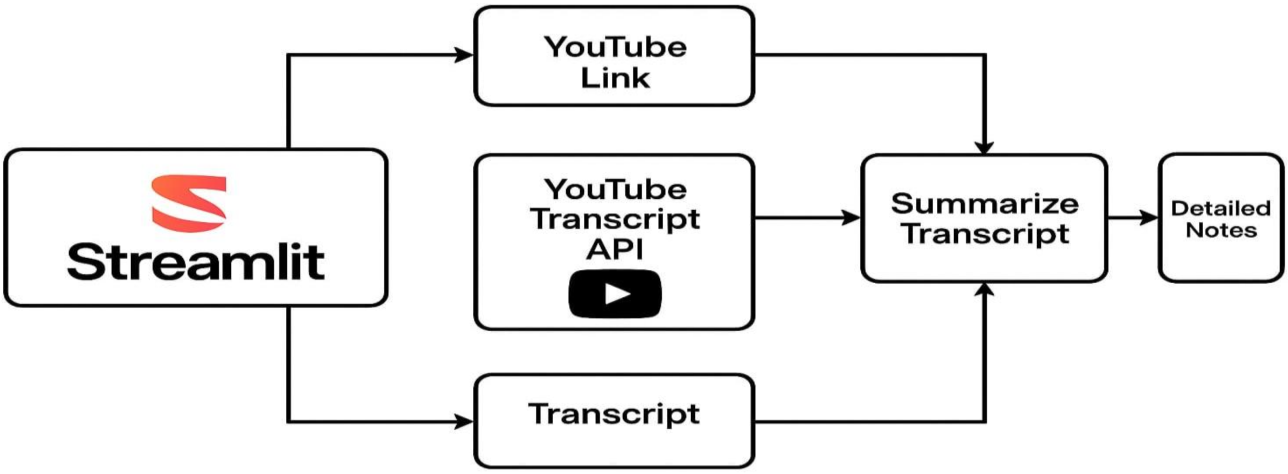


Fig. 4.1.1.1 System Architecture

Technical Architecture:

The technical architecture of **"Vocal Gist"** is fundamentally a Python-based web application leveraging the Streamlit framework, meticulously designed to provide a robust, scalable, and intuitive solution for comprehensive video content analysis. At the **Front-End Layer**, the user engages with a dynamically generated graphical interface, entirely constructed through Streamlit's Python components. This innovative approach means that despite the browser ultimately rendering standard web technologies like HTML, CSS, and JavaScript, developers are entirely abstracted from writing this low-level web code. Instead, all UI elements and their associated interaction logic, such as st.text_input for precise URL entry, st.button for triggering core actions like summarization and translation, st.selectbox for seamless language selection, and st.write/st.markdown for displaying the processed outputs and informative messages, are elegantly defined within the app.py Python script. This unique methodology significantly accelerates the development lifecycle, facilitating rapid prototyping and efficient deployment, as the entire user interface and its underlying interaction mechanisms are managed cohesively within a single, highly readable Python codebase, rendered dynamically and responsively in the client's web browser.

The **Application Logic Layer** forms the operational heart of Vocal Gist, residing on the server where the app.py Streamlit script is actively executed. This critical layer is meticulously engineered toorchestrate the entire workflow of the application, serving as the central hub for all data processing and control. It robustly processes incoming user requests originating from the web browser, accurately extracts crucial video identifiers from various YouTube URL formats utilizing Python's re module, and intelligently manages the application's state across diverse user interactions through st.session_state. This state management mechanism is absolutely vital for maintaining persistent context, such as the current summary or previously translated text, thereby preventing redundant data re-processing and optimizing performance.In such instances, it provides clear, actionable, and informative messages directly to the user through Streamlit's integrated alert mechanisms, ensuring a smooth and resilient user experience.

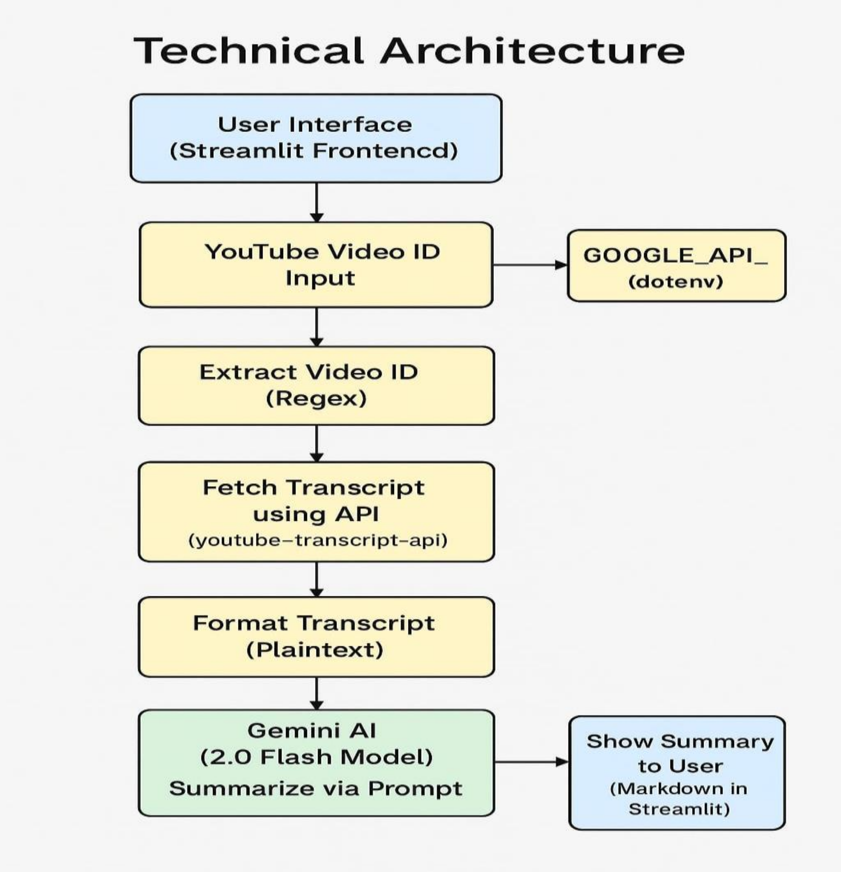

Technical Architecture
User Interface (Streamlit Frontencd)
YouTube Video ID Input
GOOGLE_API_ (dotenv)
Extract Video ID (Regex)
Fetch Transcript using API (youtube-transcript-api)
Format Transcript (Plaintext)
Gemini AI (2.0 Flash Model) Summarize via Prompt
Show Summary to User (Markdown in Streamlit)

Fig. 4.1.2.1 Technical Architectur

5.IMPLEMENTATION

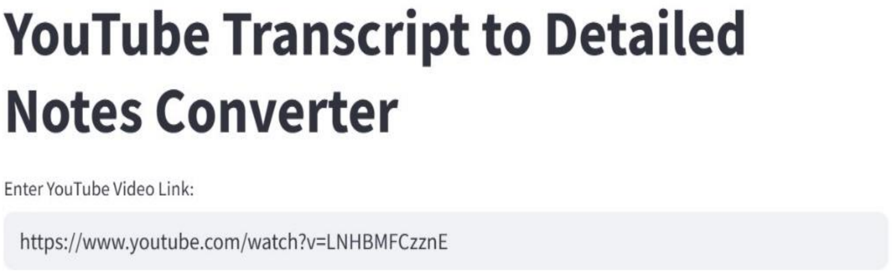
5.1Libraries

- **Streamlit:** This is the foundational library for the application's front-end and overall structure. Streamlit enables the rapid creation of interactive web applications entirely in Python, abstracting away the complexities of HTML, CSS, and JavaScript. It provides intuitive components like st.text_input for user input, st.button for triggering actions, st.selectbox for language selection, and st.write and st.image for displaying dynamic content. Streamlit also manages the session state (st.session_state), which is vital for maintaining data (like the generated summary or translated text) across user interactions without re-running entire scripts unnecessarily. This significantly simplifies development and allows for a pure Python development experience for web applications.
- **python-dotenv (via load_dotenv):** The dotenv library is essential for securely managing environment variables within the project. Specifically, load_dotenv() is called at the application's startup to load key-value pairs from a .env file into the operating system's environment variables. This is particularly critical for protecting sensitive information, such as the

GOOGLE_API_KEY, by keeping it out of the main codebase and public repositories. By using this library, the application adheres to best practices for credential management, making the system more secure and flexible for deployment in various environments.

- **google.generativeai (genai):** This is the core library for integrating "Vocal Gist" with Google's advanced Artificial Intelligence capabilities. It provides the necessary interface to interact with Google's Generative AI models, specifically the gemini-2.0-flash model. Through this library, the application sends raw video transcripts to the AI for abstractive summarization, crafting concise, bullet-pointed notes. Furthermore, it facilitates on-demand multilingual translation of these summaries, enabling the application to break down language barriers. The genai library handles the communication protocols, authentication, and data formatting required to leverage Google's powerful cloud-based AI services effectively.
- **youtube_transcript_api:** This specialized library is crucial for the initial data acquisition step, allowing "Vocal Gist" to retrieve the raw textual content from YouTube videos. It provides programmatic access to the timed transcripts (subtitles or closed captions) associated with a given YouTube video ID. The library handles the complexities of interacting with YouTube's data services, parsing the transcript data, and providing it in a usable format. It also includes robust error handling .
- **re (Regular Expressions):** The built-in Python re module is fundamental for precise pattern matching and string manipulation within "Vocal Gist." Its primary use case is the robust extraction of the unique YouTube video_id from various formats of YouTube URLs that users might input. Regular expressions allow the application to identify and isolate the 11-character video ID from long watch?v= links, shortened youtu.be/ links, or embed/ links. This ensures that regardless of the URL format provided by the user, the application can consistently obtain the correct identifier needed to access the video's transcript.
- **os (Operating System Interface):** The built-in Python os module provides a way of interacting with the operating system. In "Vocal Gist," its primary role is to access environment variables, most notably os.getenv("GOOGLE_API_KEY"). This function retrieves the Google API key that was loaded into the environment by python-dotenv, allowing the google.generativeai library to be securely configured for AI model access.
- **xml.etree.ElementTree (ET):** While not explicitly called directly for standard operations, this built-in Python module is implicitly referenced and crucial for robust error handling within the youtube_transcript_api. When the youtube_transcript_api attempts to parse XML-formatted transcript data received from YouTube, it can sometimes encounter malformed or empty XML. Catching ET.ParseError explicitly helps "Vocal Gist" to specifically identify and handle situations where transcript data is invalid or missing, providing more accurate error messages to the user and preventing the application from crashing unexpectedly.

6.SCREENSHOTS



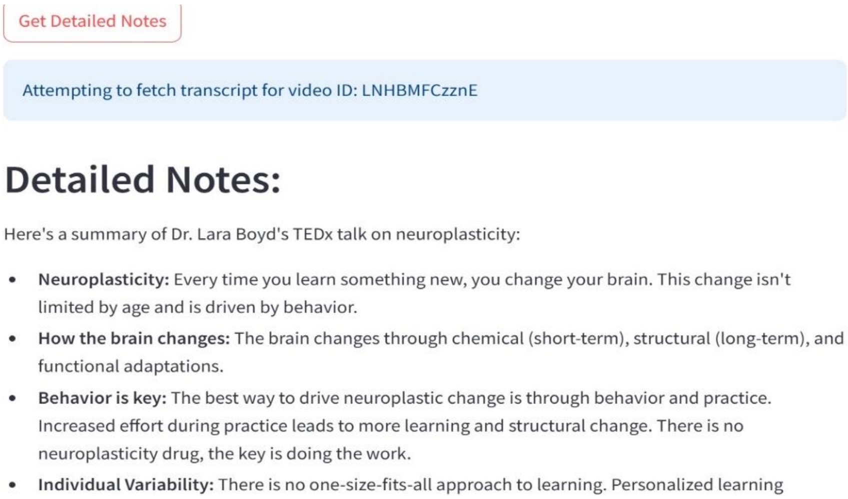
YouTube Transcript to Detailed Notes Converter

Enter YouTube Video Link:

https://www.youtube.com/watch?v=LNHBMFCzznE

Screenshot 6.1 Enter URL

Screenshot 6.2 youtube video thumbnail



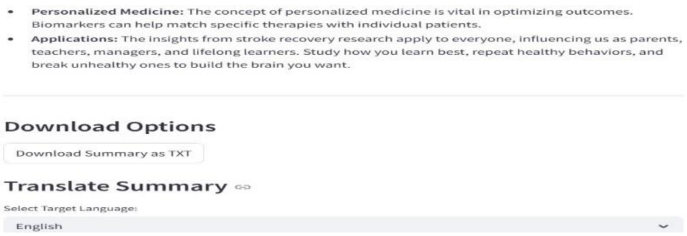
Attempting to fetch transcript for video ID: LNHBMFCzznE

Detailed Notes:

Here's a summary of Dr. Lara Boyd's TEDx talk on neuroplasticity:

- **Neuroplasticity:** Every time you learn something new, you change your brain. This change isn't limited by age and is driven by behavior.
- **How the brain changes:** The brain changes through chemical (short-term), structural (long-term), and functional adaptations.
- **Behavior is key:** The best way to drive neuroplastic change is through behavior and practice. Increased effort during practice leads to more learning and structural change. There is no neuroplasticity drug, the key is doing the work.
- **Individual Variability:** There is no one-size-fits-all approach to learning. Personalized learning

Screenshot 6.3 Generate summary



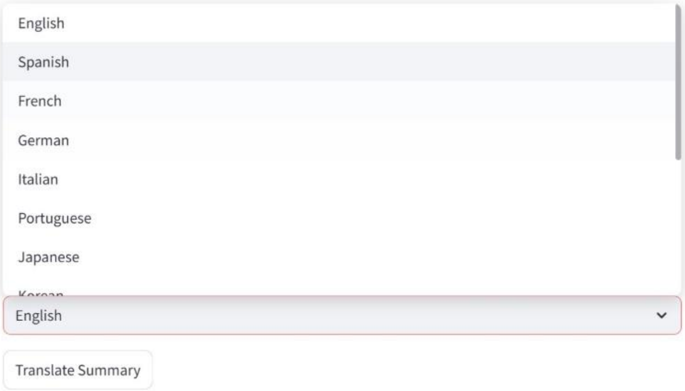
- **Personalized Medicine:** The concept of personalized medicine is vital in optimizing outcomes. Biomarkers can help match specific therapies with individual patients.
- **Applications:** The insights from stroke recovery research apply to everyone, influencing us as parents, teachers, managers, and lifelong learners. Study how you learn best, repeat healthy behaviors, and break unhealthy ones to build the brain you want.

Download Options

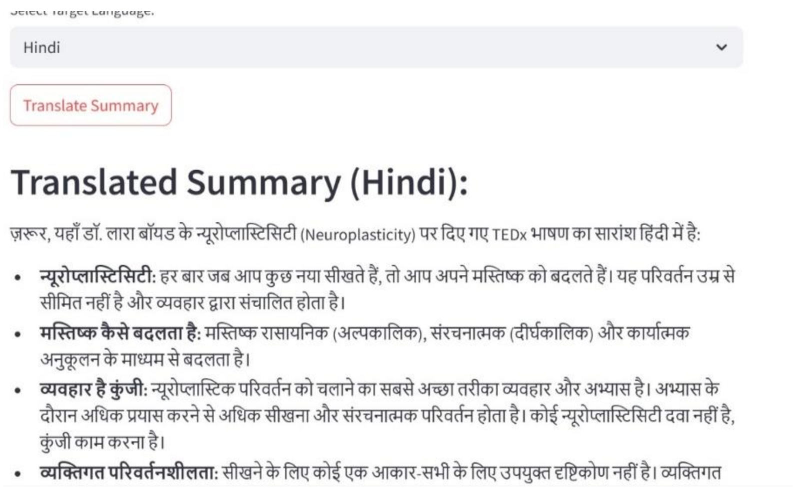
Translate Summary

Select Target Language: English

Screenshot 6.4 Download Summary



Screenshot 6.5 translation



Select Target Language:

Translated Summary (Hindi):

ज़रूर, यहाँ डॉ. लारा बॉयड के न्यूरोप्लास्टिसिटी (Neuroplasticity) पर दिए गए TEDx भाषण का सारांश हिंदी में है:

- **न्यूरोप्लास्टिसिटी:** हर बार जब आप कुछ नया सीखते हैं, तो आप अपने मस्तिष्क को बदलते हैं। यह परिवर्तन उम्र से सीमित नहीं है और व्यवहार द्वारा संचालित होता है।
- **मस्तिष्क कैसे बदलता है:** मस्तिष्क रासायनिक (अल्पकालिक), संरचनात्मक (दीर्घकालिक) और कार्यात्मक अनुकूलन के माध्यम से बदलता है।
- **व्यवहार है कुंजी:** न्यूरोप्लास्टिक परिवर्तन को चलाने का सबसे अच्छा तरीका व्यवहार और अभ्यास है। अभ्यास के दौरान अधिक प्रयास करने से अधिक सीखना और संरचनात्मक परिवर्तन होता है। कोई न्यूरोप्लास्टिसिटी दवा नहीं है, कुंजी काम करना है।
- **व्यक्तिगत परिवर्तनशीलता:** सीखने के लिए कोई एक आकार-सभी के लिए उपयुक्त दृष्टिकोण नहीं है। व्यक्तिगत

Screenshot 6.6 Translate Summary

7. CONCLUSION

The **"Vocal Gist"** project successfully delivers a robust and intuitive web-based solution that significantly enhances the efficiency of consuming information from YouTube videos. By effectively addressing the challenges of information overload and time-consuming manual data extraction, the application streamlines the process of transforming lengthy video content into concise, actionable, and multilingual summaries. Through its seamless integration of Streamlit for a user-friendly interface, youtube_transcript_api for reliable transcript retrieval, and Google's gemini-2.0-flash AI model for advanced summarization and translation, Vocal Gist stands as a practical tool for diverse users, from students and researchers to content creators, who require quick access to video insights. Its modular architecture and robust error handling further contribute to its reliability and maintainability.

In essence, Vocal Gist not only automates a traditionally manual and tedious process but also democratizes access to video knowledge by overcoming language barriers and providing easily downloadable notes. This empowers users to derive maximum value from video content with minimal effort, fostering greater productivity and more efficient learning. The project demonstrates the powerful synergy between modern web development frameworks and advanced artificial intelligence, laying a strong foundation for future enhancements and broader applications in digital content analysis.

REFERENCES

[1] IJCRT.ORG.“YOUTUBE TRANSCRIPT SUMMARIZER.” Ijcrt.org.

[2] Analytic Vidya. “Creating a Youtube Summariser – Mini NLP Project.” Analytics Vidhya.

[3] Rice, Damien, and Matt Galbraith. Video Transcript Summarizer, Atluri Naga Sai Sri Vybhavi.

[4] “YouTube Transcript Summarizer using Natural Language Processing.” International Journal of Advanced Research in Science, Communication and Technology.