VIDEO Q/A

TEAM MEMBERS

- **JYOTHI GUMMALA(TL)**
- **RYALI JAYA SAI SRI VARDHAN**
- **SEELAM VIJAYA LAKSHMI**
- **VENKATA ABHIRAM AGNIHOTHRAM**

MENTORS

- **ARAVIND PAPPALA**
- **NAGENDRA KISHORE GIRAJALA**

PROBLEM STATEMENT

In today's digital era, video content is becoming increasingly popular across various platforms. However, extracting specific information or answering questions from videos remains a challenge. Our project aims to address this issue by developing a Video-Based Question Answering (QA) System.

With the proliferation of online video content, users often encounter situations where they want to retrieve specific information or answers from videos quickly and accurately. Traditional methods of manual searching or skimming through videos are time-consuming and inefficient. Therefore, there is a growing need for automated systems that can analyze videos and provide relevant answers to user queries.

OBJECTIVES

The main objective of this project is to develop a web-based platform where users can upload videos and ask questions related to the video content. The system will then analyze the videos using advanced video processing and natural language processing (NLP) techniques to extract relevant information and generate accurate answers to the user's questions.

Key Features:

- **Video Upload:** Users can upload videos to the platform from their devices or provide URLs of online videos.
- **Question Input:** Users can input their questions related to the video content using text input fields.
- **Video Processing:** The system will process the uploaded videos using computer vision techniques to extract visual features and analyze the content.
- **Content Understanding:** Train the AI system to comprehend the context of the questions within the video content.
- **Text Analysis:** The system will perform NLP analysis on the audio transcripts and subtitles (if available) of the videos to understand the spoken content.
- **Question Answering:** Using the extracted visual and textual information, the system will generate accurate answers to the user's questions.
- **User Engagement:** Enhance user engagement by facilitating interaction with video content through questions and answers.
- **User-Friendly Interface:** Design an intuitive and easy-to-use interface for users to navigate the platform and interact with video content.
- **Continuous Improvement:** Continuously refine and optimize the system to enhance accuracy and responsiveness to user inquiries.

SOLUTION FOR THE PROBLEM

In tackling the challenge of developing a Video-Based Question Answering System, we propose harnessing the comprehensive capabilities of Google Cloud Services, including cutting-edge offerings like Vertex AI. By integrating various Google Cloud tools and services, we can create a scalable, efficient, and accurate solution that meets the requirements of the project. Here's an overview of the solution:

- **Video Upload and Storage:** Google Cloud Storage provides a reliable and scalable solution for storing uploaded videos securely. Utilizing Cloud Storage ensures efficient handling of large video files while maintaining high availability and durability.
- **Video Analysis:** Google Cloud Video Intelligence API offers powerful video analysis capabilities, including object detection, scene segmentation, and shot change detection. By leveraging this API, we can extract valuable visual features from the uploaded videos, enabling a deeper understanding of the content and context.
- **Speech-to-Text Conversion:** Google Cloud Speech-to-Text API enables accurate transcription of audio content from the videos into text. This API supports multiple languages and dialects, ensuring accurate conversion even for diverse linguistic contexts.
- **Natural Language Understanding:** Google Cloud Natural Language API provides advanced NLP capabilities for analyzing and understanding text data. By utilizing this API, we can perform entity recognition, sentiment analysis, and syntactic parsing to extract meaningful insights from the transcribed video content.
- **Vertex AI for Machine Learning:** Google Cloud's Vertex AI offers a unified platform for building, training, and deploying machine learning models at scale. We can leverage Vertex AI to develop a custom question answering model, trained on the extracted video features and textual data. Vertex AI's AutoML capabilities can facilitate model training and optimization, reducing the need for manual intervention.
- **User Interface and Integration:** Google Cloud App Engine or Firebase can be used to develop a user-friendly web interface for uploading videos, inputting questions, and displaying answers. Integration with Google Cloud Functions allows for seamless communication between frontend and backend components, ensuring smooth user interaction.

HTML AND CSS

HTML stands for Hyper Text Markup Language.HTML is the standard markup language for creating Web pages.HTML describes the structure of a Web page.HTML consists of a series of elements.HTML elements tell the browser how to display the content.Here's a deep dive into HTML, providing detailed information beyond the basics:

Structure and Elements:

- **Document Types:** HTML5 is the most common document type declaration (<!DOCTYPE html>). It specifies the document type as HTML and helps browsers render the content correctly.
- **Document Structure:** An HTML document has a basic structure:
  - <html> tag: Encloses the entire HTML document content.
  - <head> section: Contains meta information about the document (title, character encoding).
  - <body> section: Holds the visible content displayed on the webpage.
- **Elements and Attributes:** Elements are the building blocks of your content. They define different parts of your webpage, like headings, paragraphs, lists, images, forms, etc. Elements are created using opening and closing tags (e.g., <h1> for headings, <p> for paragraphs). Attributes provide additional information about elements. Here are some common attributes:
- **Common Elements:** It includes headings,paragraphs,lists,images,tables,forms.
- **Semantic Elements:** Use semantic elements like <header>, <nav>, <main>, <section>, <article>, <aside>, and <footer> to convey meaning beyond just visual presentation. This improves accessibility and SEO.
- **Advanced HTML Topics:**
  - **Document Object Model (DOM):** The DOM represents the HTML document structure as a tree of objects. JavaScript can manipulate the DOM to dynamically change the content and behavior of a webpage.
  - **HTML Entities:** Certain characters like "<" or "&" have special meanings in HTML. Use entity references (e.g., < for "<") to display these characters correctly.

- **Embedded Content:** Include multimedia elements like images, videos, audio, and iframe's using appropriate HTML elements.

CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes. The style definitions are normally saved in external .css files. With an external stylesheet file, you can change the look of an entire website by changing just one file.

CSS is a vast and ever-evolving language. Keep exploring, experiment with code, and refer to online resources to expand your knowledge and create stunning websites.

- **Core Functionality:**
  - **Styling Elements:** Change the visual appearance of HTML elements using CSS. This includes font size, color, background, borders, padding, margin, and more.
  - **Separation of Concerns:** CSS separates presentation (styling) from content (HTML). This improves maintainability and allows for easier website updates.
  - **Cascading Rules:** CSS rules "cascade" down the page, with more specific selectors taking priority over broader ones. This helps manage complex styles.
- **Selectors:** CSS selectors are used to "find" (or select) the HTML elements you want to style. Simple selectors (select elements based on name, id, class),Combinator selectors (select elements based on a specific relationship between them).Pseudo-class selectors (select elements based on a certain state).Pseudo-elements selectors (select and style a part of an element).Attribute selectors (select elements based on an attribute or attribute value).
- **Layout Techniques:** Float Layout, flexbox, grid layout.
- **Advanced Features:** Animations and transitions, media queries, preprocessors.

NODE JS

Node.js is an open-source, cross-platform JavaScript runtime environment that allows you to run JavaScript code outside of a web browser. Here's a comprehensive description:

Core Functionality:

- **Built on Chrome's V8 Engine:** Node.js leverages the V8 JavaScript engine, the same engine that powers Google Chrome, to execute JavaScript code very efficiently. This enables you to use JavaScript for both front-end (web browser) and back-end (server-side) development.
- **Event-Driven & Non-Blocking I/O:** Node.js is designed for handling many concurrent connections effectively. It employs an event-driven, non-blocking I/O model. This means it doesn't create separate threads for each request, making it lightweight and fast. Instead, it uses a single thread to handle events efficiently.
- **Asynchronous Programming:** Since Node.js is non-blocking, it relies heavily on asynchronous programming. This means operations that take time (like accessing a database) don't block the main thread. Node.js can handle other requests while waiting for the asynchronous operation to complete. This allows Node.js to be highly scalable and handle many concurrent requests efficiently.

Traditional vs. Modern Approaches:

- **Callback Functions:** Traditionally, Node.js used callback functions to handle asynchronous operations. A callback function is provided to an operation, and it's executed once the operation finishes. This can lead to "callback hell" for complex logic with nested callbacks.
- **Promises & Async/Await:** Modern Node.js development promotes Promises and Async/Await for asynchronous programming. Promises offer a cleaner way to manage the results of asynchronous operations, and Async/Await provides a more synchronous-like syntax for writing asynchronous code, improving readability.

Applications of Node.js:

- **Microservices Architecture:** Node.js excels at building microservices, which are small, independent services that work together to form a larger application. Its scalability and speed make it ideal for this architecture.

- **Real-time Applications:** Node.js is perfect for building applications with constant data updates, like chat applications, stock tickers, or collaborative editing tools. It can handle numerous concurrent connections efficiently for real-time communication.
- **API Development:** Node.js is a great choice for creating RESTful APIs that allow other applications to access data or functionality. Frameworks like Express.js simplify API development in Node.js.
- **Internet of Things (IoT):** Node.js is well-suited for building server-side applications for IoT devices due to its ability to handle real-time data streams and manage numerous connected devices.
- **Data Streaming:** Node.js is adept at processing and delivering data streams efficiently, making it suitable for building real-time data pipelines or streaming applications.

Popular Node.js Frameworks:

- **Express.js:** A popular minimalist web framework for building web applications and APIs.
- **NestJS:** A framework for building scalable and enterprise-grade Node.js applications with a focus on structure and organization.
- **Koa.js:** A lightweight framework offering a flexible foundation for building web applications or APIs.

INTRODUCTION TO VERTEX AI

Vertex AI is a machine learning (ML) platform that lets you train and deploy ML models and AI applications, and customize large language models (LLMs) for use in your AI-powered applications. Vertex AI combines data engineering, data science, and ML engineering workflows, enabling your teams to collaborate using a common toolset and scale your applications using the benefits of Google Cloud.

Vertex AI provides several options for model training and deployment:

- AutoML lets you train tabular, image, text, or video data without writing code or preparing data splits.
- Custom training gives you complete control over the training process, including using your preferred ML framework, writing your own training code, and choosing hyperparameter tuning options.
- Model Garden lets you discover, test, customize, and deploy Vertex AI and select open-source (OSS) models and assets.
- Generative AI gives you access to Google's large generative AI models for multiple modalities (text, code, images, speech). You can tune Google's LLMs to meet your needs, and then deploy them for use in your AI-powered applications.

After you deploy the models, use Vertex AI's end-to-end MLOps tools to automate and scale projects throughout the ML lifecycle. These MLOps tools are run on fully-managed infrastructure that you can customize based on your performance and budget needs.

We can use the Vertex AI SDK for Python to run the entire machine learning workflow in Vertex AI Workbench, a Jupyter notebook-based development environment. We can collaborate with a team to develop your model in Colab Enterprise, a version of Colaboratory that is integrated with Vertex AI.

Other available interfaces include the Google Cloud Console, the gcloud command line tool, client libraries, and Terraform (limited support).

GOOGLE CLOUD API'S

Google Cloud APIs are programmatic interfaces to Google Cloud Platform services. They are a key part of Google Cloud Platform, allowing you to easily add the power of everything from computing to networking to storage to machine-learning-based data analysis to your applications.

Cloud APIs are exposed as network API services to customers, such as cloud pub/sub-API. Each Cloud API typically runs on one or more subdomains of googleapis.com, such as pubsub.googleapis.com, and provides both JSON HTTP and gRPC interfaces to clients over public internet and Virtual Private Cloud (VPC) networks. Clients can send HTTP and gRPC requests to Cloud API endpoints directly or by using client libraries.

Accessing:

We can access Cloud APIs from server applications with our client libraries in many popular programming languages, from mobile apps via the Firebase SDKs, or by using third-party clients. We can also access Cloud APIs with the Google Cloud CLI tools or Google Cloud console. All Cloud APIs provide a simple JSON HTTP interface that you can call directly or via Google API Client Libraries. Most Cloud APIs also provide a gRPC interface you can call via Google Cloud Client Libraries, which provide better performance and usability. We can also use third-party clients.

TLS encryption: All Cloud APIs accept only secure requests using TLS encryption.

- We are using one of our client libraries, in-transit encryption is handled for you by the library.
- We are using our own gRPC client, you need to authenticate with Google (which requires TLS) following the instructions in the gRPC authentication guide.
- We are creating your own HTTP client, see our HTTP guidelines.

Monitoring the usage: Most Cloud APIs provide you with detailed information on your project's usage of that API, including traffic levels, error rates, and latencies. It helps you to quickly triage problems with applications that use Cloud APIs. You can view this information in the Google Cloud API Dashboard in the Google Cloud console. You can also create custom dashboards and alerts in Cloud Monitoring.

ADVANTAGES

Here are some of the advantages of Video Q/A:

- **Saves time:** Your website can automatically analyze videos, saving users time compared to manual analysis.
- **Improves accuracy:** Automated analysis can be more accurate than manual analysis, especially for repetitive tasks.
- **Provides insights:** Your website can identify patterns and trends in videos that users might miss.
- **Personalized learning:** The code analysis feature can tailor suggestions and feedback to the user's coding style and skill level.
- **Improved debugging:** Video analysis can be used to visualize code execution and highlight where errors occur within the video itself.
- **Accessibility tools:** For videos with poor audio quality, your website could use code analysis to generate captions or transcripts, making them more accessible to people with hearing impairments.
- **Collaboration:** The platform could allow users to share and analyse videos and code collaboratively, fostering teamwork and knowledge sharing.
- **Data collection and analysis:** By analysing large amounts of video and code data, your website could identify trends and insights that would be difficult to see manually. This could be valuable for research or business applications.

DISADVANTAGES

While your Video-Based Question Answering System offers numerous advantages, it's essential to consider potential disadvantages and challenges:

- **Dependency on Video Quality**: The effectiveness of your system may be limited by the quality of the uploaded videos. Poor audio or video quality, background noise, or low-resolution footage could impact the accuracy of content analysis and question answering.
- **Language and Accent Limitations:** The system's natural language processing (NLP) capabilities may struggle with understanding diverse accents, slang, or non-standard language usage. This could result in inaccuracies or errors in generating answers, particularly for users with different linguistic backgrounds..
- **Limited Context Understanding:** Understanding context from videos can be challenging, especially in cases where visual or audio cues are ambiguous or require cultural knowledge. The system may struggle to provide accurate answers without a deep understanding of the context surrounding the video content.
- **Scalability and Processing Time:** Analysing and processing large volumes of video content in real-time can be computationally intensive and time-consuming. As the number of users and uploaded videos increases, scalability issues may arise, leading to delays in generating responses or system overload.
- **Privacy and Data Security Concerns:**Users may have concerns about the privacy and security of their uploaded videos and personal data. Ensuring robust data encryption, compliance with privacy regulations, and implementing strict access controls are essential to address these concerns and build trust among users.
- **Bias and Inaccuracy:**Like any AI-driven system, your question answering model may exhibit bias or inaccuracies, particularly in its understanding of subjective or nuanced content. Addressing bias in AI algorithms and continuously improving model accuracy through training data diversity and feedback mechanisms is crucial.
- **Maintenance and Updates:** Maintaining and updating the system to adapt to evolving user needs, technological advancements, and changes in video content formats or platforms require ongoing resources and effort. Regular updates, bug fixes, and feature enhancements are necessary to keep the system relevant and functional.

CONVERTING TO BASE 64

```
const express = require('express')
const fs = require('fs')
const { VertexAI } = require('@google-cloud/vertexai')
const cors = require('cors')
const multer = require('multer')
const app = express();
const port = 3000;
const videoFile = multer({dest:"uploads/videos"})
app.use(express.json());
app.use(cors());
// Function to convert video file to base64
function convertVideoToBase64(filePath) {
    try {
        // Read the video file
        const videoData = fs.readFileSync(filePath);
        // Convert video data to base64
        const base64EncodedVideo = Buffer.from(videoData).toString('base64');
        return base64EncodedVideo;
    } catch (error) {
        console.error('Error:', error);
        return null;
    }
}
// Function to generate content
async function generateContent(base64Video, userInputText) {
```

```
  try {
    // Initialize Vertex with your Cloud project and location
    const vertex_ai = new VertexAI({ project: 'zeta-crossbar-41', location: 'northamerica-northeast1' });
    const model = 'gemini-1.0-pro-vision-001';
    // Instantiate the models
    const generativeModel = vertex_ai.preview.getGenerativeModel({
      model: model,
      generation_config: {
        "max_output_tokens": 2048,
        "temperature": 0.4,
        "top_p": 1,
        "top_k": 32
      },
      safety_settings: [
        { "category": "HARM_CATEGORY_HATE_SPEECH", "threshold": "BLOCK_MEDIUM_AND_ABOVE" },
        { "category": "HARM_CATEGORY_DANGEROUS_CONTENT", "threshold": "BLOCK_MEDIUM_AND_ABOVE" },
        { "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT", "threshold": "BLOCK_MEDIUM_AND_ABOVE" },
        { "category": "HARM_CATEGORY_HARASSMENT", "threshold": "BLOCK_MEDIUM_AND_ABOVE" }
      ],
    });
    const req = {
      contents: [{
```

```
        role: 'user',
        parts: [
            { inline_data: { mime_type: 'video/mp4', data: base64Video } },
            { text: userInputText }
        ]
    }],
};
const streamingResp = await generativeModel.generateContentStream(req);
const generatedTexts = [];
for await (const item of streamingResp.stream) {
    if (item.candidates) {
        item.candidates.forEach(candidate => {
            candidate.content.parts.forEach(part => {
                if (part.text) {
                    generatedTexts.push(part.text);
                }
            });
        });
    }
}
return generatedTexts;
// const streamingResp = await generativeModel.generateContentStream(req);
// const generatedTexts = [];
// if (!streamingResp || !streamingResp.stream) {
//     console.error('Stream response is missing or empty.');
//     return generatedTexts;
```

```
        // }

        // for await (const item of streamingResp.stream) {
        //     if (item.candidates) {
        //         item.candidates.forEach(candidate => {
        //             candidate.content.parts.forEach(part => {
        //                 if (part.text) {
        //                     generatedTexts.push(part.text);
        //                 }
        //             });
        //         });
        //     }
        // }

        // return generatedTexts;
    } catch (error) {
        console.error('Error:', error);
        throw error; // Rethrow the error to be caught by the caller
    }
}

// // API endpoint to generate content
// app.post('/generate-content' ,videoFile.any(), async (req, res) => {
//     const {userInputText } = req.body;
//     console.log(req.body)
//     console.log(req.files[0] + "aksljdfhklaj")
```

```
//     console.log(filePath);
//     const base64Video = convertVideoToBase64(filePath);

//     if (!base64Video) {
//         return res.status(500).json({ error: 'Failed to convert video to base64' });
//     }

//     try {
//         console.log("Here it is passing");
//         const generatedTexts = await generateContent(base64Video, userInputText);
//         res.json({ generatedTexts });
//     } catch (error) {
//         res.status(500).json({ error: 'Failed to generate content', details: error.message });
//     }
// });

// API endpoint to generate content
app.post('/generate-content', videoFile.single('filePath'), async (req, res) => {
    const { userInputText } = req.body;
    const filePath = req.file.path; // Get the file path from multer
    // Convert video to base64
    const base64Video = convertVideoToBase64(filePath);

    if (!base64Video) {
        return res.status(500).json({ error: 'Failed to convert video to base64' });
    }
```

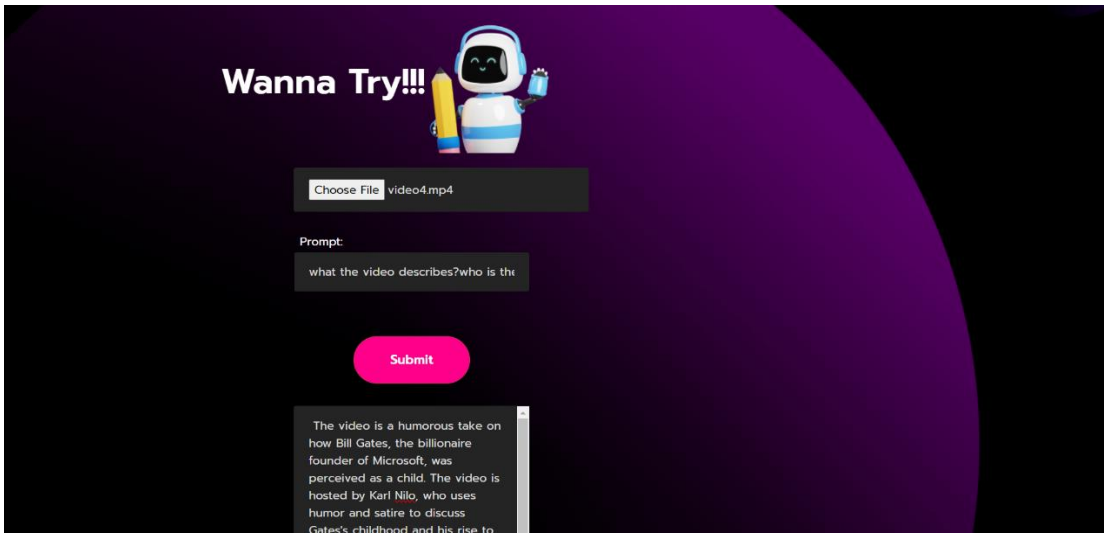
```
    try {
        const generatedTexts = await generateContent(base64Video, userInputText);
        res.json({ generatedTexts });
    } catch (error) {
        res.status(500).json({ error: 'Failed to generate content', details: error.message });
    }
});

// Start the server
app.listen(port, () => {
    console.log(`Server is running on port ${port}`);
```

OUTPUT


Video Q&A
Summarize All Your Videos At A Single Place
Welcome to Video Q&A, your solution for instant answers from any video! Upload your video, ask your question – and presto! Our advanced technology scans through content to provide precise answers, saving your time. No more endless searches or tedious scrolling. Whether you're a student, professional, or curious mind, Video Q&A streamlines knowledge extraction. Stop struggling with time-consuming searches. Join the effortless learning revolution with Video Q&A today!


Yeah,
@KARLNIILO
00:00:15
00:00:32
checkingggg


Wanna Try!!!
Choose File video4.mp4
Prompt:
what the video describes?who is the
Submit
The video is a humorous take on how Bill Gates, the billionaire founder of Microsoft, was perceived as a child. The video is hosted by Karl Nilo, who uses humor and satire to discuss

APPLICATIONS

Here are some potential applications of the project:

- **Education and E-Learning Platforms:** Integrating the system into educational platforms can enhance learning experiences by providing students with interactive video content and the ability to ask questions directly related to the material. Teachers can also use the system to create quizzes or assessments based on video content.
- **Media and Entertainment Industry:** Content creators and media companies can utilize the system to engage audiences more effectively by allowing viewers to ask questions about video content, such as movies, TV shows, or documentaries. This can enhance viewer interaction and provide additional insights into the content.
- **Customer Support and FAQ Systems:** Companies can deploy the system as part of their customer support infrastructure to provide automated responses to frequently asked questions related to product tutorials, troubleshooting guides, or service demos. This can improve customer satisfaction and reduce support ticket volumes.
- **Training and Onboarding Programs:** Organizations can incorporate the system into training and onboarding programs to deliver interactive video-based learning modules. Employees can ask questions about training videos, policies, or procedures, and receive immediate answers, enhancing their learning experience.
- **Healthcare and Medical Education:** Medical professionals and students can use the system to analyze medical videos, such as surgical procedures or diagnostic imaging, and ask questions for clarification or further understanding. This can facilitate medical education and training.
- **Legal and Compliance Training:** Law firms and compliance departments can leverage the system to provide video-based training on legal topics, regulations, or compliance procedures. Users can ask questions about legal concepts or case studies and receive accurate answers, enhancing their comprehension.
- **Research and Data Analysis**: Researchers and analysts can utilize the system to analyze large volumes of video data for insights and trends. They can ask questions about specific aspects of the videos and receive answers based on the analyzed content, facilitating data-driven decision-making.

CONCLUSION

- In conclusion, the development of the Video-Based Question Answering System leveraging Google Cloud Services represents a significant advancement in enhancing the accessibility, usability, and functionality of video content analysis.
- By harnessing the capabilities of Google Cloud Storage, Video Intelligence API, Speech-to-Text API, Natural Language API, and Vertex AI, we have created a scalable, efficient, and accurate solution for extracting insights and answering questions from uploaded videos.
- Through rigorous development, testing, and integration, the project has achieved its objectives of providing users with a seamless platform for uploading videos, inputting questions, and receiving accurate answers based on the analyzed content. The utilization of advanced machine learning models trained on diverse datasets has ensured the system's ability to handle various video content types and user queries effectively.
- Furthermore, the project has demonstrated the value of leveraging cloud-based solutions for video content processing, enabling scalability, reliability, and cost-effectiveness.
- The integration of security measures and compliance with data protection regulations ensures the confidentiality and integrity of user data, fostering trust and confidence among users.
- Video-Based Question Answering System has the potential to revolutionize the way users interact with video content, making it more accessible, actionable, and informative. Continued refinement, optimization, and updates to the system will further enhance its performance and user experience, paving the way for broader adoption and impact in diverse domains and industries.