Department of Computer Applications

National Institute of Technology Kurukshetra (136119)

Artificial Intelligence

(MCA-202), 2024-25

Semester Long Course Project

Submitted by:

Name : Rupali Priya

Roll no: 523110051

Group no: 5

Semester: 4

Submitted to:

Dr. Anshu Parashar

Agentic AI Video Summarizer

Abstract

In the digital era, where vast multimedia content is consumed daily, extracting meaningful insights from long-form videos is a challenge. This project presents an **Agentic AI Video Summarizer**, a Streamlit-based web application that allows users to upload videos, generate intelligent summaries using Google Gemini via the Phi Agent framework, and translate the content into multiple languages using Deep Translator. It provides a seamless, interactive, and accessible way to summarize and analyze videos, showcasing the power of Agentic AI in multimedia content analysis.

Keywords

Agentic AI, Video Summarization, Google Gemini, Streamlit, Deep Translator, Natural Language Processing.

I. Introduction

Multimedia content is growing at an exponential rate. Users often find it difficult to extract core ideas from lengthy videos, especially when the content is in a foreign language. Traditional summarization methods lack interactivity and real-time response. This project addresses the need for a smart, interactive AI tool that can summarize video content and provide answers to user-specific queries with translation support.

II. Problem Statement

Most users do not have the time or tools to process long videos for key takeaways. Additionally, language barriers make global video content less accessible. A practical solution is required to:

- Analyze and summarize long videos quickly.
- Respond to custom user prompts about the video content.
- Provide multilingual accessibility for the summarized content.

III. Objectives

- **Ob_1**: To develop a web application that can accept video uploads.
- **Ob_2**: To process and analyze video content using the Gemini AI model.
- **Ob_3**: To implement an agentic framework that allows intelligent interaction with video insights.
- **Ob_4**: To allow translation of generated summaries into multiple languages.
- **Ob_5**: To create a user-friendly UI using Streamlit.

IV. Technologies and Tools Used

- **Major Domain**: Artificial Intelligence, Natural Language Processing
- **Language**: Python
- **Frameworks**: Streamlit
- **Libraries/Tools**:
    - phi: for creating an agent and using Google Gemini models
    - google.generativeai: for uploading and fetching video files
    - deep_translator: for translation functionality
    - DuckDuckGo: for knowledge augmentation
    - .env, dotenv: for secure key handling
- **APIs**: Google Gemini API via AI Studio

V. Methodology

1. **Video Upload**: Users upload .mp4, .mov, or .avi files.
2. **Processing**: Files are uploaded to Google Gemini's platform.
3. **Querying**: User enters a custom question. The agent uses the video and external tools to answer.
4. **Summary Generation**: AI model returns a markdown summary.
5. **Translation**: Summary is translated into selected language using GoogleTranslator.
6. **UI/UX**: Implemented using Streamlit layout and CSS customization.

VI. Outcomes and Deliverables

- Source Code (Python files, Streamlit app)

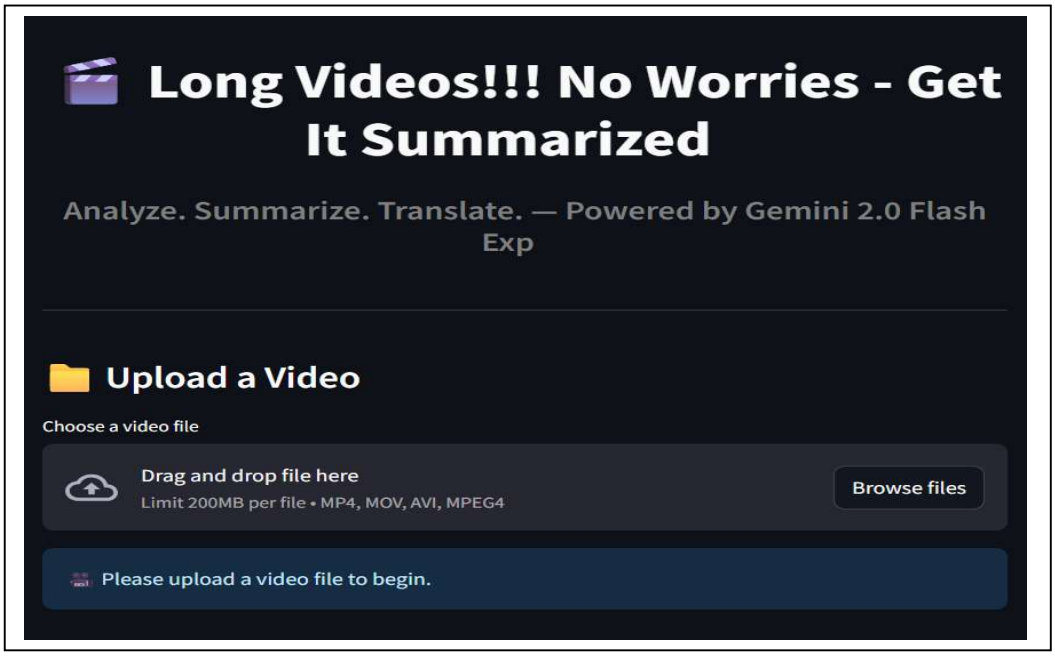
```
•
•  # Core Imports
•  import streamlit as st
•  from phi.agent import Agent
•  from phi.model.google import Gemini
•  from phi.tools.duckduckgo import DuckDuckGo
•  from google.generativeai import upload_file, get_file
•  import google.generativeai as genai
•  from deep_translator import GoogleTranslator
•  import time
•  import tempfile
```

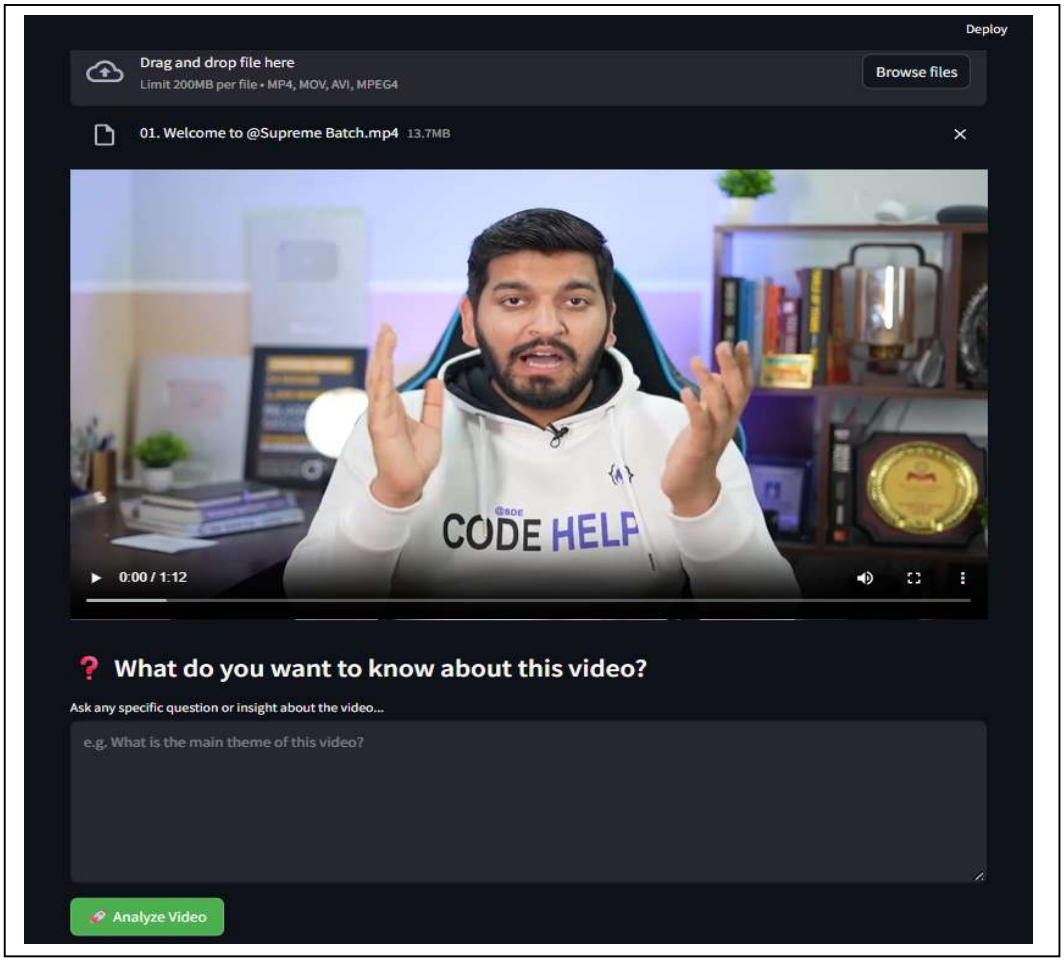
```
from pathlib import Path
from dotenv import load_dotenv
import os

# Load API Key
load_dotenv()
API_KEY = os.getenv("GOOGLE_API_KEY")
if API_KEY:
    genai.configure(api_key=API_KEY)

# Set Page Config
st.set_page_config(
    page_title="🤖 AI Video Summarizer",
    page_icon="🤖",
    layout="wide",
)

st.markdown("""

    <style>
    body {
        background-color:#f0f0f0;
    }
    .main {
        background-color:#999999;
        padding: 2rem;
        border-radius: 12px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    .stButton button {
        background-color: #4CAF50;
        color: white;
        border-radius: 8px;
        padding: 0.5rem 1rem;
    }
    .stTextArea textarea {
        height: 180px !important;
    }

    </style>
""", unsafe_allow_html=True)

# Title and Header
st.markdown("<h1 style='text-align: center;'>🎬 Long Videos!!!   No Worries - Get It Summarized</h1>", unsafe_allow_html=True)
st.markdown("<h4 style='text-align: center; color: gray;'>Analyze. Summarize. Translate. — Powered by Gemini 2.0 Flash Exp</h4>", unsafe_allow_html=True)
st.markdown("---")

# Session State
```

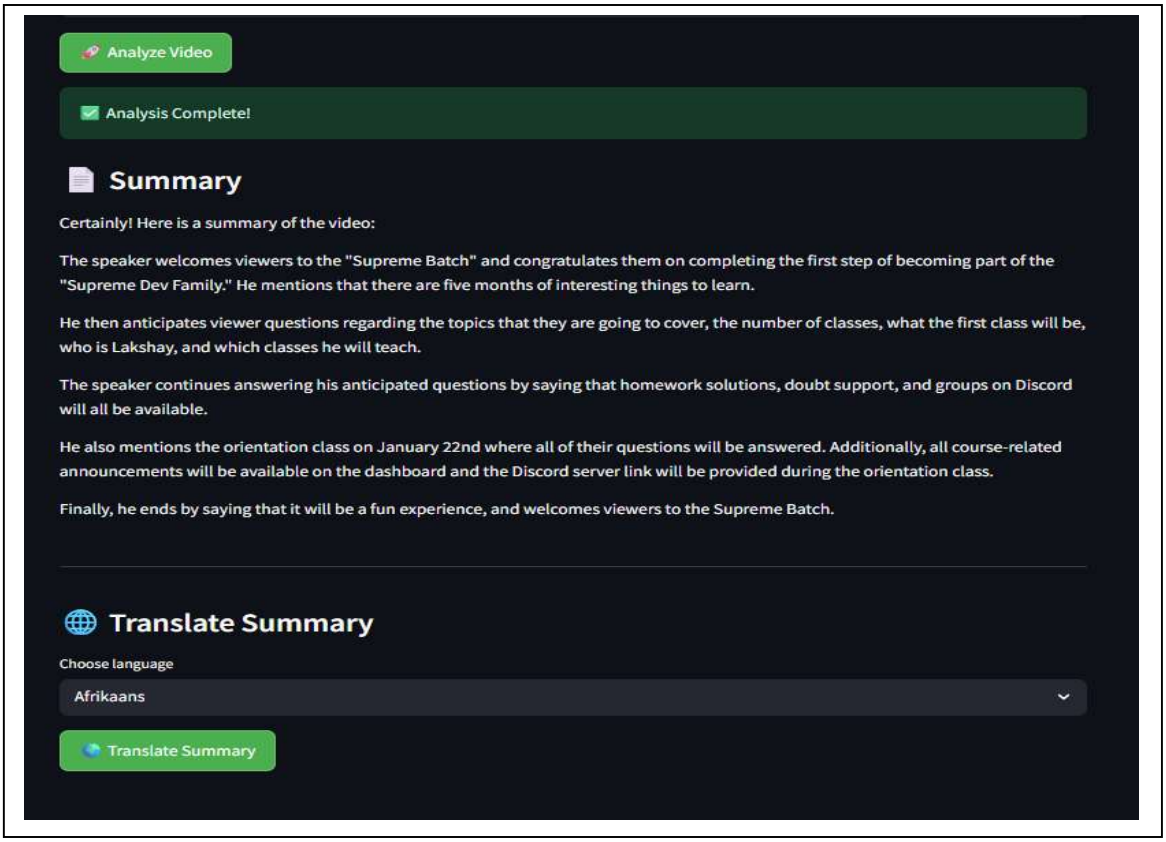
```python
if 'original_summary' not in st.session_state:
    st.session_state.original_summary = None

# Initialize AI Agent
@st.cache_resource
def initialize_agent():
    return Agent(
        name="Video AI Summarizer",
        model=Gemini(id="gemini-2.0-flash-exp"),
        tools=[DuckDuckGo()],
        markdown=True,
    )
agent = initialize_agent()


# Upload Video
st.subheader("📁 Upload a Video")
video_file = st.file_uploader("Choose a video file", type=['mp4', 'mov', 'avi'])

if video_file:
    with tempfile.NamedTemporaryFile(delete=False, suffix='.mp4') as temp_video:
        temp_video.write(video_file.read())
        video_path = temp_video.name
    st.video(video_path, format="video/mp4")

    st.subheader("❓ What do you want to know about this video?")
    user_query = st.text_area(
        "Ask any specific question or insight about the video...",
        placeholder="e.g. What is the main theme of this video?",
    )

    if st.button("🚀 Analyze Video"):
        if not user_query:
            st.warning("Please enter a question before analyzing the video.")
        else:
            try:
                with st.spinner("🔍 Processing video, please wait..."):
                    processed_video = upload_file(video_path)
                    while processed_video.state.name == "PROCESSING":
                        time.sleep(1)
                        processed_video = get_file(processed_video.name)

                    prompt = f"""
                    Analyze the uploaded video for content and context.
                    Respond to the following query using video insights and web
knowledge:
                    {user_query}
                    Give a clear, concise, and user-friendly answer.
                    """

```

```
                    response = agent.run(prompt, videos=[processed_video])
                    st.session_state.original_summary = response.content

                st.success("✅ Analysis Complete!")
                st.subheader("📝 Summary")
                st.markdown(response.content)

            except Exception as error:
                st.error(f"An error occurred: {error}")
            finally:
                Path(video_path).unlink(missing_ok=True)

    if st.session_state.original_summary:
        st.markdown("---")
        st.subheader("🌐 Translate Summary")
        languages = GoogleTranslator().get_supported_languages()
        lang_dict = {lang.capitalize(): lang for lang in languages}
        selected_lang = st.selectbox("Choose language", list(lang_dict.keys()),
index=0)

        if st.button("🌍 Translate Summary"):
            try:
                translated = translate_text(st.session_state.original_summary,
lang_dict[selected_lang])
                st.success("✅ Translation Successful!")
                st.subheader(f"🗣️ Translated Summary ({selected_lang})")
                st.markdown(translated)
            except Exception as error:
                st.error(f"Translation error: {str(error)}")

else:
    st.info("🤖 Please upload a video file to begin.")

# Translation helper function
def translate_text(text, dest_language):
    try:
        translator = GoogleTranslator(source='auto', target=dest_language)
        return translator.translate(text)
    except Exception as e:
        return f"Error in translation: {e}"
```

- Web Application
- AI Agent with video reasoning
- Multilingual Summary Output
- UI for video input and user prompts


Long Videos!!! No Worries - Get It Summarized
Analyze. Summarize. Translate. — Powered by Gemini 2.0 Flash Exp
Upload a Video
Choose a video file
Drag and drop file here
Limit 200MB per file • MP4, MOV, AVI, MPEG4
Browse files
Please upload a video file to begin.


Deploy
Drag and drop file here
Limit 200MB per file • MP4, MOV, AVI, MPEG4
Browse files
01. Welcome to @Supreme Batch.mp4 13.7MB
CODE HELP
0:00 / 1:12
What do you want to know about this video?
Ask any specific question or insight about the video...
e.g. What is the main theme of this video?
Analyze Video



VII. Results and Discussion

The application successfully generated accurate, user-friendly summaries based on various queries. It also demonstrated translation capability to languages like Hindi, French, German, and more. The agent framework ensured real-time, intelligent interaction with user input. This proves the potential of Agentic AI in multimedia education, accessibility, and research applications.

VIII. Conclusion

This project showcases a practical implementation of Agentic AI that democratizes access to video content by providing smart summaries and translations. It is a scalable solution for educational, corporate, and research sectors. Future work includes enhancing the UI, adding sentiment analysis, and supporting real-time video summarization.

IX. References

1. Google AI Studio: https://makersuite.google.com/
2. Streamlit Documentation: https://docs.streamlit.io/
3. Deep Translator: https://pypi.org/project/deep-translator/
4. Phi Agent Framework: https://github.com/nomic-ai/phi
5. DuckDuckGo Tools for AI