VideoAutoMaker: A Fully Automated AI-Powered Video Generation Pipeline

NP_123
Rice University
np123greatest@gmail.com

Abstract

*We introduce **VideoAutoMaker**, an automated AI-powered video generation pipeline that transforms a given script into complete video segments with synchronized visuals, TTS audio, and metadata. VideoAutoMaker streamlines video content creation for platforms such as YouTube, TikTok, and Bilibili. The system integrates large language models (LLMs), text-to-speech synthesis, stock video search, and text-to-video generation to produce coherent, platform-ready output. We present our system architecture, implementation details, experimental results, and future directions. The source code for VideoAutoMaker is publicly available at* `https://github.com/NPgreatest/VideoAutoMaker`.

1. Introduction

Creating engaging video content requires effort across scriptwriting, narration, visual matching, and editing. **VideoAutoMaker** aims to automate the entire pipeline using modular AI components. Given a user-input script and theme, the system generates corresponding audio narration, retrieves or synthesizes matching video segments, and outputs a ready-to-publish video with subtitles and metadata.

This pipeline enables efficient creation of educational, narrative, and promotional content, dramatically reducing the time and skill needed to produce high-quality videos.

2. Related Work

Recent advances in AI content generation have led to a surge of tools for synthetic video, audio, and multimodal media creation. We categorize related work into three major areas: avatar-based systems, retrieval/generative video models, and AI infrastructure for compositional pipelines.

Avatar and TTS-Driven Video Systems

Platforms like **Synthesia** [11] and **HeyGen** provide human-avatar video synthesis from scripts, integrating TTS with face animation. However, they often lack scene control, creative storytelling, or open-source flexibility. Projects like **Wav2Lip** [7] align speech to talking faces but require real footage or avatars.

Video Generation and Retrieval

RunwayML [9], **Pika Labs**, and **Gen-2** by Runway have popularized text-to-video diffusion models, enabling short generative video clips from prompts. **Pexels** and **Unsplash** provide retrieval APIs for stock footage, used in systems such as **Vizard** [12]. More research-focused efforts include **CogVideo** [4], **Make-A-Video** [10], and **VideoCrafter** [13], demonstrating high-fidelity generative capability using diffusion or transformer-based backbones.

Our work integrates both retrieval-based and generative methods by dynamically switching modes per video block, guided by scene semantics.

LLM-Based Storytelling and Scene Understanding

HuggingFace Transformers [6] and **DeepSeek-V3** [3] provide general-purpose LLMs that are capable of generating structured narration and scene descriptions. **AllenAI**'s pipelines [1] demonstrate how structured reasoning can guide downstream visual selection or synthesis.

Our system extends this by allowing per-line decision logic using LLMs, injecting control into a generative pipeline that would otherwise be passive.

Multimodal Compositional Pipelines

Projects like **Open-Sora** [**?**] and **Meta's Make-A-Scene** [5] explore multimodal inputs to condition video synthesis. Unlike monolithic generators, our pipeline emphasizes modularity and flexibility—users can plug in custom TTS (e.g., GPT-SoVITS [2]), video backends (SiliconFlow, Wan2.1), or subtitle processors (Whisper [8]) with ease.

Distinction from Prior Work

Unlike existing systems that optimize for single-mode synthesis (avatar or prompt-to-video), **VideoAutoMaker** connects narration, LLM reasoning, video sourcing, and rendering into a fully modular and automated pipeline. Its

design encourages experimentation, cost control, and creative flexibility.

3. Use Cases and Applications

- **Education:** Automatically generate lectures or explainers with visuals and narration.
- **YouTube Shorts:** Create catchy video bites from news, scripts, or tweets.
- **Language Learning:** Create character-driven dialogue videos with subtitles.
- **Storytelling:** Narrate fictional content with cinematic visuals using minimal user input.

Discuss how users can fine-tune the pipeline for different scenarios by adjusting prompt templates or selecting specific characters.

4. System Design

The system is organized as follows:

1. **Input:** User submits a script and video theme.
2. **Project Setup:** Generates a structured JSON project file.
3. **Audio Generation:** Switches to the correct GPT-SoVITS fine-tuned model and generates speech for each line.
4. **Video Clip Decision:** Uses an LLM to decide whether to query Pexels (stock search) or generate video via a text-to-video model.
    - If Pexels: Sends an LLM-generated search query to fetch stock video.
    - If T2V: Uses the prompt to submit a video generation task to SiliconFlow.
5. **Post-processing:** Records timestamps, generates .srt subtitles.
6. **Composition:** Uses ffmpeg to assemble audio, video, and subtitles into a final `.mp4`.

5. Implementation

The backend is written in Python and uses FastAPI to expose a REST API. Modules include:

- `project_manager.py` for project creation and JSON state management

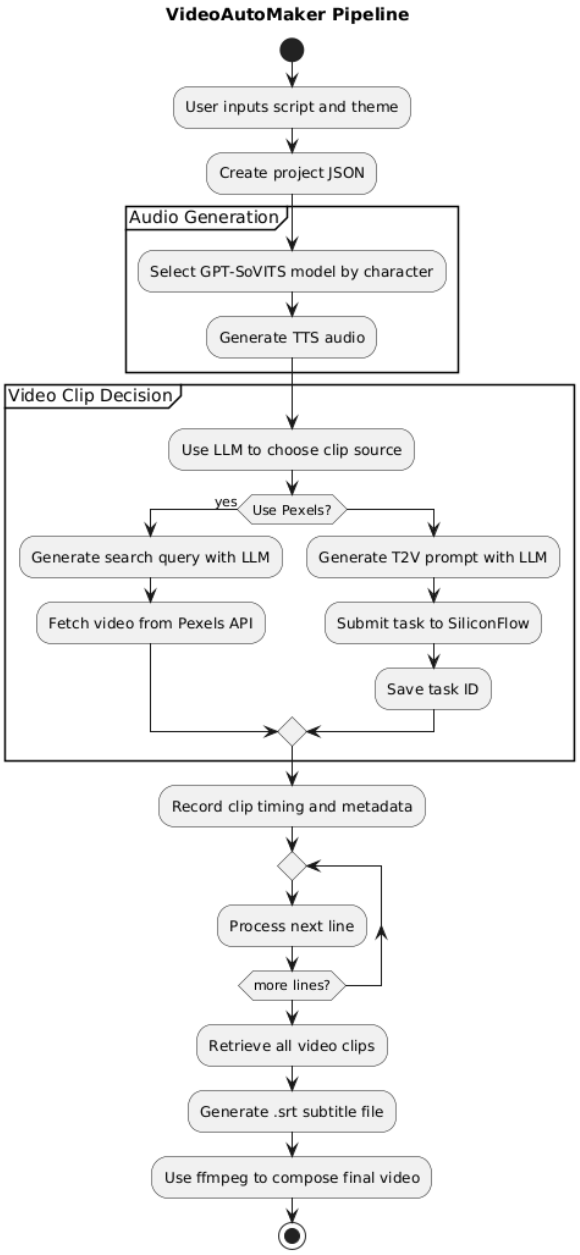

Figure 1. Pipeline overview. LLM decides per-block whether to retrieve stock video or generate custom clip, synchronized with TTS audio.

- `MediaProcessor` for per-block audio/video generation
- GPT-SoVITS API for audio synthesis with dynamic speaker switching
- Pexels API and LLM-generated search queries for visual matching
- Optionally: text-to-video generation submitted to a re-

mote queue (SiliconFlow)

- ffmpeg composition of final video

The frontend (previously Gradio) has been replaced by a FastAPI-based REST interface. Projects can now be managed fully via curl or frontend integration.

6. Cost Analysis

To evaluate the operational cost of generating a 10-minute video with **VideoAutoMaker**, we analyze expenses across text-to-video generation, LLM-based scene reasoning, and text-to-speech synthesis.

Assumptions

- A 10-minute video contains ∼150 lines of narration (1.5 lines per 6s).
- Each line uses DeepSeek-V3 (LLM) for analysis.
- 40% of lines require text-to-video generation using Wan2.1.
- TTS is performed locally using GPT-SoVITS on a 6GB RTX 3060 GPU.
- Electricity rate: $0.12/kWh (US average).
- RTX 3060 power draw during inference: ∼90W.

Cost Breakdown

1. Video Generation (Wan2.1)
Wan2.1 charges 1.5 per generated video (1 task per clip).
Total clips = 40% of 150 lines = 60 clips.
Cost: 1.5 × 60 = 90

2. LLM Analysis (DeepSeek-V3)
DeepSeek charges 8 per million tokens.
Assume 25 tokens per line × 150 lines = 3,750 tokens.
Cost: 8 × (3,750 / 1,000,000) = 0.03

3. TTS (GPT-SoVITS on RTX 3060)
TTS runs locally with negligible API cost, but GPU electricity usage applies.
Assume 1x real-time TTS: 10 minutes of audio takes 10 minutes of compute.
Power usage = 90W × (10/60) hours = 0.015 kWh
Cost: 0.015 × $0.12 $0.0018 (0.013)

Total Estimated Cost (10-Min Video)

- **Text-to-Video:** 90
- **LLM Token Usage:** 0.03
- **TTS (Local RTX 3060):** 0.013

Total: 90.04 (= $12.43)
This estimate shows that the dominant cost lies in video generation (Wan2.1), while local TTS and LLM inference contribute negligibly per video.

7. Results

We used the system to generate videos on topics like science explainers and story narration.

- **Quantitative:** Each project takes 2-3 minutes per minute of video content. Most blocks complete audio+video generation in under 20s.
- **Qualitative:** Fine-tuned character voices improve immersion. LLM-generated video queries result in relevant, high-quality visuals.

Figure 2. Example output: narrator-driven video generated from a paragraph-level script with matching clips.

8. Future Work

We plan to enhance VideoAutoMaker with:

- Subtitle smoothing and alignment via Whisper
- Music/BGM layering and automatic beat sync
- Full automation of YouTube/Bilibili publishing
- Web dashboard for project editing and preview

9. Comparison with Existing Pipelines

To better position **VideoAutoMaker** among existing automated video generation platforms, we compare its core features with two prominent tools: RunwayML and Pika Labs. Table 1 highlights key capabilities across systems.

Unlike proprietary platforms that offer only closed APIs or limited customization, **VideoAutoMaker** combines both generative and retrieval-based approaches, integrates LLM-based decision making, and enables local execution for

Feature	RunwayML	Pika Labs	VAM
Audio Generation	–	–	✓
Character Voices	–	–	✓
Text-to-Video	✓	✓	✓
LLM Scene Reasoning	–	–	✓
Cost Control (Hybrid API/Local)	–	–	✓
Customizable Backend	–	–	✓
Open Source	–	–	✓

Table 1. Comparison of features across modern video generation platforms.

cost-efficient TTS generation. This flexibility makes it suitable for research, content creators, and application developers alike.(VAM = VideoAutoMaker)

Unlike proprietary platforms that offer only closed APIs or limited customization, VideoAutoMaker combines both generative and retrieval-based approaches, integrates LLM-based decision making, and enables local execution for cost-efficient TTS generation. This flexibility makes it suitable for research, content creators, and application developers alike.

10. Conclusion

VideoAutoMaker presents a practical AI-powered pipeline for turning structured text into media-rich videos. By connecting LLM reasoning, character-based TTS, and flexible video generation/retrieval, our system enables rapid and high-quality content production.

The modular structure supports future plug-ins, fine-tuning, and scaling. We hope this inspires further development in automated creative pipelines.

Acknowledgments

We thank the contributors to GPT-SoVITS, Pexels, Open-Sora, and the open-source AI community. Their tools made this project possible.

References

[1] Allen Institute for AI. Allenai research pipelines. https://allenai.org, 2023. Accessed: 2025-04-17.

[2] O. S. Contributors. Gpt-sovits: Prompt-controllable zero-shot voice synthesis. https://github.com/RVC-Project/GPT-SoVITS, 2024. Accessed: 2025-04-17.

[3] DeepSeek AI. Deepseek-v3: General-purpose large language model. https://github.com/deepseek-ai/DeepSeek-V3, 2024. Accessed: 2025-04-17.

[4] H. X. et al. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868, 2022.

[5] O. Gafni et al. Make-a-scene: Scene-based text-to-image generation with human priors. arXiv preprint arXiv:2203.13131, 2022.

[6] Hugging Face. Transformers by hugging face. https://huggingface.co/transformers/, 2023. Accessed: 2025-04-17.

[7] K. Prajwal, R. Mukhopadhyay, et al. Wav2lip: Accurate lip syncing for speech. arXiv preprint arXiv:2008.10010, 2020.

[8] A. Radford et al. Whisper: Robust speech recognition via large-scale weak supervision. https://openai.com/research/whisper, 2022. Accessed: 2025-04-17.

[9] Runway. Runwayml: Creative tools powered by machine learning. https://runwayml.com, 2023. Accessed: 2025-04-17.

[10] A. Singer, A. Polyak, M. Tsimpoukelli, et al. Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792, 2022.

[11] Synthesia. Synthesia ai video generation platform. https://www.synthesia.io/, 2023. Accessed: 2025-04-17.

[12] Twelve Labs. Vizard: Ai-powered stock video generator. https://www.vizard.ai/, 2023. Accessed: 2025-04-17.

[13] C. Wu et al. Videocrafter: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2303.13439, 2023.