

Automated Text-to-Video System

Objective

Develop a fully automated pipeline that converts a raw text prompt into a complete short-form video.

The primary focus is on system architecture, backend automation, and R&D using exclusively free and open-source tools.

Important: Interns are required to research and select the most suitable, up-to-date open-source resources themselves; the examples provided are for reference only and not mandatory.

Workflow

The system should follow a modular “Chain of Generation” approach:

1. Script & Scene Parsing

- Process input text into segmented scenes, scene descriptions, and narration scripts.
- Use local or self-hosted LLMs for text understanding and structuring.

2. Visual Generation

- Generate relevant images or video clips for each scene.
- Use open-source models (examples: Stable Diffusion, ModelScope). Interns may select the best available alternatives.

3. Audio Synthesis

- Convert narration scripts into speech using high-quality TTS engines.

- Examples: Bark, Coqui TTS, Piper. Interns may explore latest open-source TTS options.

4. Assembly & Synchronization

- Programmatically stitch visuals and audio together.
- Include subtitles/captions where appropriate.
- Use Python libraries like MoviePy or FFmpeg, or equivalent open-source tools.

Constraints

- Zero-Cost Requirement: Only free and open-source libraries, models, and tools are allowed.
- Local / Self-Hosted: Prefer models that can run locally or on free-tier environments.
- Frontend Focus: A CLI or minimal frontend is sufficient.
- Full Automation: The pipeline must run end-to-end without manual intervention.
- Resource Selection: Interns are required to research and select the most suitable, up-to-date open-source resources themselves; the examples provided are illustrative only.

Deliverables

- Source Code: Clean, well-documented Python repository.
- Dependencies: requirements.txt or environment.yml for easy setup.

- Sample Output: At least three ~30-second videos generated by the pipeline.
- Technical Documentation: README explaining system design, model choices, and usage instructions.