ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may562

AutoVidGen: A Fully Automated Text-to-Video Generation Pipeline Using AI and Multimedia Tools

Mohammed Lubna Firdous[1]; Thameena Nousheen[2]; Dr. K. Rajitha[3]; K. Shirisha[4]; Dr. K. Sreekala[5*]

[1;2;3;4;5]Department of Computer Science and Engineering Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India

Corresponding Author: Dr. K. Sreekala[5*]

Publication Date: 2026/05/25

Abstract: Short-form video has become one of the most effective ways to explain ideas, promote learning material, and share information on social media platforms. However, creating even a short vertical video usually requires script writing, voice recording, subtitle preparation, stock media selection, editing, and final rendering. This paper presents AutoVidGen, a Python-based automated text-to-video generation pipeline that converts a single user-provided topic into a complete vertical video. The system combines a large language model for narration script generation, neural text-to-speech for voiceover creation, Whisper-based transcription for timed captions, language-model-assisted keyword extraction for visual search, the Pexels API for stock video retrieval, and MoviePy with FFmpeg support for final video composition. The final output is rendered as a 1080 x 1920 MP4 video at 30 frames per second, making it suitable for platforms such as YouTube Shorts, Instagram Reels, and TikTok. The system is implemented with both command-line and Streamlit interfaces, allowing users to generate videos with minimal technical effort. Experimental use of the prototype shows that the pipeline can quickly produce usable educational and informational videos, while the final quality depends on topic clarity, keyword relevance, stock footage availability, and caption rendering reliability.

Keywords: *Text-to-Video Generation; Generative AI; Large Language Model; Text-to-Speech; Whisper; Automatic Subtitles; Stock Video Retrieval; MoviePy; Short-Form Video; Content Automation.*

How to Cite: Mohammed Lubna Firdous; Thameena Nousheen; Dr. K. Rajitha; K. Shirisha; Dr. K. Sreekala (2026) AutoVidGen: A Fully Automated Text-to-Video Generation Pipeline Using AI and Multimedia Tools. *International Journal of Innovative Science and Research Technology*, 11(5), 1653-1656. https://doi.org/10.38124/ijisrt/26may562

I. INTRODUCTION

Video has become a primary medium for digital communication because it can combine speech, visuals, timing, emotion, and context in a compact form. Educational channels, social media creators, marketing teams, and students increasingly depend on short videos to explain concepts and reach wider audiences. At the same time, video production remains a multi-step activity. A creator must normally write a script, record narration, locate suitable visual clips, prepare captions, arrange the media on a timeline, adjust the aspect ratio, and export the final video. This workflow can be difficult for beginners and time-consuming even for experienced users.

Recent progress in artificial intelligence makes it possible to reduce this manual effort. Large language models can generate topic-specific narration, neural text-to-speech systems can produce natural voiceovers, speech recognition models can create timestamped captions, and media processing libraries can assemble audio, text, and video into a final output. Instead of treating these capabilities as separate tools, AutoVidGen connects them into a single practical pipeline.

The main objective of this work is to design and implement a system that accepts only a topic from the user and generates a complete vertical video. The system focuses on short-form videos because that format is widely used on mobile-first platforms. The proposed pipeline does not attempt to synthesize every frame from scratch; instead, it uses AI for planning, narration, voice, and captioning while retrieving relevant stock videos for the visual layer. This approach makes the system lighter, faster, and easier to run on a student-level development setup.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may562

II. LITERATURE SURVEY

Text-to-video generation has gained significant attention in recent years. Early generative video research explored the use of recurrent networks, generative adversarial networks, and motion-content separation to model temporal patterns. Vondrick et al. introduced work on generating videos with scene dynamics, while Tulyakov et al. proposed MoCoGAN to separate motion and content during video synthesis. These studies helped establish the foundation for video generation but were limited by low resolution, unstable training, and weak control over semantic content.

The development of transformer architectures and diffusion-based generative models improved text alignment and visual quality. Models such as Make-A-Video, latent diffusion video synthesis, DirecT2V, LAVIE, and StreamingT2V show that natural language can guide video generation with increasing realism. However, such models often require large computational resources, extensive training data, and specialized infrastructure. These requirements make them difficult to deploy in a lightweight academic prototype.

Parallel progress has occurred in supporting technologies. Transformer-based language models can generate coherent scripts and extract structured information from text. Neural text-to-speech systems such as WaveNet, Tacotron, and modern cloud or edge voices improve the quality of synthetic narration. Whisper and related automatic speech recognition models provide accurate transcription with timestamps, which is essential for synchronized subtitles. AutoVidGen builds on these practical advances by orchestrating available AI services and multimedia tools rather than training a new end-to-end video synthesis model.

- *Problem Statement*

The existing process of creating short informational videos requires several separate tools and a large amount of manual coordination. Users who want to create a video from a simple topic must usually prepare the narration, search for footage, record or synthesize voice, create subtitles, edit clips, and export the final result in the required aspect ratio. This process increases production time and creates a barrier for students, educators, and small creators who may not have advanced editing knowledge.

The problem addressed in this work is the lack of a simple and integrated system that can transform a plain topic into a complete short-form vertical video. The system should reduce manual editing effort, produce synchronized captions, select relevant visuals, and render the final video in a format ready for social media use.

- *Proposed System*

AutoVidGen is proposed as a modular pipeline for automated video creation. The user enters a topic through either the command-line interface or the Streamlit web interface. The backend then generates a narration script, converts the script into speech, creates captions from the generated audio, extracts visual search keywords, downloads suitable portrait-oriented stock clips, and combines all components into a final MP4 video.

The proposed system is intentionally divided into independent modules. The script generator is responsible for narration text, the text-to-speech module handles voiceover creation, the caption module generates timestamped subtitle segments, the keyword module extracts searchable visual phrases, the Pexels module retrieves stock footage, and the video assembler produces the final render. This separation makes the system easier to test, debug, and extend.

- *System Architecture*

The architecture follows a sequential data flow. First, the input topic is passed to the script generation module, which uses the Groq large language model to produce an 80 to 120 word conversational narration. The same script is used in two branches: one branch creates audio through Edge TTS, and the other extracts visual keywords for stock media search. The generated audio is processed by Whisper to create caption segments with start and end timestamps.

The keyword list is sent to the Pexels Video API, which downloads relevant stock clips. The assembler module then loads the voiceover, captions, and downloaded videos. Each clip is resized and cropped to a 9:16 vertical frame. The clips are concatenated, looped, or trimmed to match the voiceover duration. Finally, the audio and captions are overlaid and the output is encoded as a 1080 x 1920 MP4 video at 30 frames per second.

III. METHODOLOGY

The methodology begins with topic input. A user enters a topic such as a science concept, current educational theme, or general knowledge idea. The script generation module sends this topic to the language model with instructions to create a short, engaging, and visually rich narration. The prompt avoids scene directions and focuses on spoken content so that the result can be directly converted into a voiceover.

After the script is generated, Edge TTS converts it into an MP3 voiceover using a neural voice. The system then uses Whisper to transcribe the generated audio. Although the script is already known, transcription is still useful because it produces time-aligned caption segments. These timestamps allow subtitles to appear at the correct moments in the final video.

For visual selection, the system asks the language model to extract five concrete search queries from the script. This step converts abstract narration into media-searchable phrases. The Pexels API is then queried for portrait-oriented videos, and the selected clips are downloaded into a temporary folder. In the final stage, MoviePy processes the video clips, attaches the audio, places subtitles, and exports the final video. Temporary files are cleaned after each run so that the next generation starts from a fresh state.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may562

IV. IMPLEMENTATION

The project is implemented in Python. The main command-line workflow is handled by main.py, while app.py provides a Streamlit-based web interface. Both entry points call the same backend modules, which helps maintain consistent behavior across the CLI and web application.

The config.py file loads environment variables, validates the required Groq and Pexels API keys, creates the temp and output directories, and configures FFmpeg and ImageMagick-related settings. The script_generator.py module calls the Groq API for narration generation. The tts.py module uses edge-tts to create the voiceover file. The captions.py module uses the local Whisper model to produce timed captions. The keywords.py module generates visual search keywords, pexels.py downloads stock video clips, and video_assembler.py performs resizing, concatenation, caption overlay, audio synchronization, and final rendering.

The final output is saved in the output directory as final_video.mp4. The video uses H.264 video encoding, AAC audio, 30 FPS, and a portrait resolution of 1080 x 1920 pixels. These choices make the generated videos compatible with common short-video platforms.

V. RESULTS AND DISCUSSION

The prototype successfully demonstrates that a complete short-form video can be generated from a single topic using a coordinated AI and multimedia workflow. The system reduces the number of manual steps normally required for video creation and produces a ready-to-share MP4 with voiceover, visuals, and burned-in captions.

During testing, the most important factor affecting output quality was the clarity of the input topic. Specific topics produced stronger scripts and more accurate visual keywords. Broad or vague topics sometimes led to generic stock footage. The relevance of Pexels search results also affected visual quality because the system depends on available stock videos. Caption quality was generally reliable because Whisper generated timestamped segments from the actual voiceover audio.

Compared with manual editing, the proposed system saves time and lowers the skill barrier. Instead of using separate applications for writing, voice generation, subtitling, media search, and editing, the user interacts with one pipeline. However, the system still has limitations. It does not perform deep semantic shot matching, it depends on external APIs, and it may require ImageMagick or FFmpeg configuration for smooth rendering on Windows systems.

VI. APPLICATIONS

AutoVidGen can be used for educational explainers, quick social media content, project demonstrations, awareness videos, product concept videos, and prototype content generation. It is especially useful for students and educators who want to create short visual explanations without spending significant time on editing tools. The system can also help creators test multiple topic ideas quickly before investing time in polished manual production.

VII. LIMITATIONS

The system relies on external services for language generation and stock media retrieval, so internet connectivity and API availability are required. Stock video results may not always match the exact meaning of the narration. The current prototype uses a fixed caption style and a fixed vertical output format. It also focuses on short videos and does not yet support advanced scene planning, custom visual themes, multilingual narration, or user-controlled editing of each generated segment.

VIII. CONCLUSION AND FUTURE WORK

This paper presented AutoVidGen, a fully automated text-to-video generation pipeline that converts a user-provided topic into a vertical short-form video. By combining language generation, neural text-to-speech, Whisper-based captions, stock video retrieval, and MoviePy-based composition, the system reduces manual effort and makes video creation more accessible. The implementation shows that practical text-to-video automation can be achieved by orchestrating existing AI and multimedia tools in a carefully designed pipeline.

Future improvements can include semantic shot selection, multilingual voice generation, improved caption customization, support for different aspect ratios, real-time preview controls, user-editable scene timelines, and integration with generative video models for cases where suitable stock footage is not available.

REFERENCES

[1]. M. Singer, S. Sheynin, A. Polyak, et al., "Make-A-Video: Text-to-Video Generation without Text-Video Data," arXiv preprint arXiv:2209.14792, 2022.
[2]. A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017.
[3]. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[4]. A. Radford, J. W. Kim, C. Hallacy, et al., "Learning Transferable Visual Models From Natural Language Supervision," International Conference on Machine Learning, 2021.
[5]. A. van den Oord, S. Dieleman, H. Zen, et al., "WaveNet: A Generative Model for Raw Audio," arXiv preprint arXiv:1609.03499, 2016.
[6]. Y. Wang, R. Skerry-Ryan, D. Stanton, et al., "Tacotron: Towards End-to-End Speech Synthesis," Interspeech, 2017.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may562

[7]. A. Radford, J. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," International Conference on Machine Learning, 2023.
[8]. Z. Tulyakov, M. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing Motion and Content for Video Generation," IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[9]. C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating Videos with Scene Dynamics," Advances in Neural Information Processing Systems, 2016.
[10]. W. Wang, Y. Zhang, Z. Liu, et al., "DirecT2V: Large Language Models as Frame-Level Directors for Zero-Shot Text-to-Video Generation," IEEE/CVF International Conference on Computer Vision, 2023.