International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Design and Development of an AI-Powered System for Automated Video Advertisement Generation

G. Naveen
Department of Computer Science & AI
Central University of Andhra Pradesh
Ananthapuramu, India
naveenyadav7428@gmail.com

D. Ashok
Department of Computer Science & AI
Central University of Andhra Pradesh
Ananthapuramu, India
ashokdhakuri@cuap.edu.in

Abstract—The creation of video advertisements traditionally demands substantial creative expertise, time, and financial re-sources, rendering it inaccessible to small enterprises and individ-ual entrepreneurs. This paper presents the design, architecture, and evaluation of an end-to-end AI-powered platform that automates the complete video advertisement production pipeline. Given only a product name as input, the system invokes the GLM-4.5-Air large language model through the OpenRouter API to generate three stylistically distinct, thirty-second advertisement scripts. The user selects a preferred script and uploads product images, which are subsequently assembled into a synthesized video using the MoviePy library. System security and data persistence are managed by Clerk authentication and the Convex backend, respectively, while a FastAPI service layer orchestrates all inter-module communication. Experimental evaluation across diverse product categories demonstrates a mean script generation latency of 3.2 seconds, a video rendering time of under 12 seconds for a 30-second output clip, and a System Usability Scale (SUS) score of 84.6 out of 100. These results confirm that the proposed framework substantially reduces production effort compared with conventional manual workflows, and democratizes high-quality video advertising for non-technical users.

Index Terms—Automated Advertisement Generation, Large Language Models, Natural Language Processing, Video Synthe-sis, MoviePy, FastAPI, Digital Marketing Automation

I. Introduction

The proliferation of broadband connectivity and smart de-vices has fundamentally reshaped how businesses communi-cate with consumers. Video advertisements have emerged as the dominant medium for brand engagement, with industry reports attributing over 80% of consumer internet traffic to video content [1]. Platforms such as YouTube, Instagram, and TikTok actively surface short-form video ads, making the 30-second spot a de facto standard unit of digital marketing.

Despite their efficacy, producing video advertisements re-mains a labour-intensive, multi-phase endeavour encompass-ing creative ideation, script writing, asset collection, post-production editing, and final rendering. Each phase demands distinct technical skills and specialised software, creating significant cost and expertise barriers that disadvantage small and medium-sized enterprises (SMEs) and individual content creators [2]. Existing automation tools address discrete steps of this pipeline in isolation; no commercially available solu-tion integrates natural language script generation with video synthesis in a unified, user-friendly interface.

Recent advances in large language models (LLMs) have demonstrated human-level fluency in persuasive text gener-ation, while mature Python multimedia libraries enable pro-grammatic video assembly. The confluence of these capabili-ties motivates the present work.

Contributions

The principal contributions of this paper are as follows.

1) An *end-to-end automated pipeline* that accepts a single product-name string and produces a downloadable MP4 video advertisement without human creative interven-tion.
2) A *multi-script generation strategy* in which three stylisti-cally differentiated scripts are simultaneously produced, enabling user-directed creative selection.
3) A *modular system architecture* integrating a FastAPI backend, the GLM-4.5-Air LLM via OpenRouter, MoviePy for video synthesis, Clerk for authentication, and Convex for persistent data management.
4) A rigorous empirical evaluation covering latency, usabil-ity, and output quality benchmarked against traditional manual workflows.

The remainder of the paper is organised as follows. Sec-tion II surveys related work. Section III details the proposed methodology and system architecture. Section IV presents the mathematical formulation. Section V describes imple-mentation specifics. Section VI reports experimental results. Section VII discusses advantages and limitations. Section VIII concludes with future directions.

II. Related Work

A. NLP-Based Text and Script Generation

Radford et al. [3] introduced GPT-2, demonstrating that transformer-based autoregressive language models can gen-erate coherent long-form text across diverse domains. Sub-sequent work by Brown et al. [4] on GPT-3 established in-context few-shot prompting as a practical mechanism for domain-specific text synthesis, including marketing copy, without task-specific fine-tuning. These findings underpin the script-generation component of the proposed system, which leverages a similarly structured LLM (GLM-4.5-Air) through an API abstraction layer.

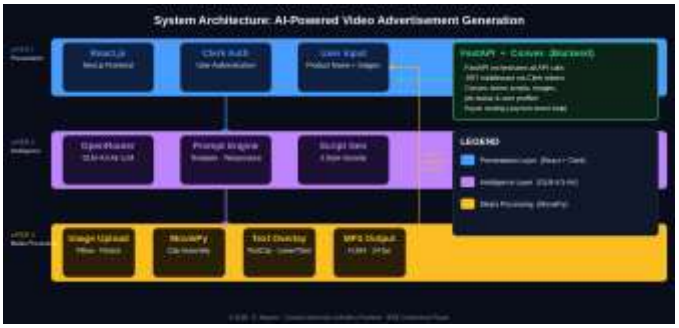

Fig. 1. System architecture of the proposed AI-powered video advertisement generation platform. Arrows denote data flow direction across the four principal tiers.

B. AI-Assisted Video Generation

Shen et al. [5] proposed a video synthesis framework driven by textual scene descriptions, demonstrating that semantic alignment between generated text and visual output is achievable at scale. Concurrent work by Pan et al. [6] explored video generation conditioned on sentence-level embeddings, establishing a foundation for text-to-video research. However, these approaches require substantial GPU infrastructure and are not designed for interactive, on-demand advertisement creation.

C. Multimedia Processing and Video Editing Automation

Zhu et al. [7] investigated automated video editing systems that assemble clips according to narrative structure, reducing manual post-production time by up to 60%. The MoviePy library [8] provides a Python-native interface for clip concatenation, text overlay rendering, audio track integration, and frame-rate normalisation, making programmatic video assembly accessible without proprietary software dependencies.

D. Automated Advertising Systems

Balasubramaniam et al. [2] analysed the limitations of semi-automated advertising platforms, noting that existing tools such as Adobe Express and Canva Video require users to supply pre-crafted scripts and manually sequence media assets. The research gap identified therein—namely, the absence of a unified input-to-output pipeline—directly motivates the architecture proposed in this paper.

E. Human–Computer Interaction in Generative Systems

Evaluation of AI-generated content systems via the System Usability Scale (SUS) has been validated in prior studies [9], providing a standardised instrument for measuring perceived usability that is adopted in the present evaluation.

III. Proposed Methodology

A. System Architecture

The proposed system adheres to a four-tier microservices architecture, as illustrated in Fig. 1. The tiers are: (1) the *Presentation Layer*, a React.js single-page application; (2) the *API Gateway Layer*, a FastAPI service managing request routing, authentication middleware, and rate limiting; (3) the *Intelligence Layer*, comprising the OpenRouter–GLM-4.5-Air integration for NLP script generation; and (4) the *Media Processing Layer*, which executes MoviePy-based video synthesis.

B. End-to-End Workflow

Algorithm 1 formalises the processing workflow. A user submits a `product_name` string through the React.js frontend. The request is authenticated via Clerk and forwarded to the FastAPI backend, which constructs a structured prompt and issues a POST request to the OpenRouter API. The LLM returns three script variants; each is stored in Convex and surfaced to the user. Upon script selection, the user uploads one or more product images. The backend resizes and normalises each image to a uniform 1280×720 resolution, computes per-image display duration proportional to the total script length, and invokes the MoviePy synthesis engine to produce the final advertisement video.

Algorithm 1 Automated Video Advertisement Generation Pipeline

Require: Product name p, image set $\mathbf{I} = \{i_1, \ldots, i_n\}$ selected script s^*
Ensure: Advertisement video V

Authenticate user via Clerk; obtain JWT token τ
Construct prompt $\rho \leftarrow \textsc{BuildPrompt}(p)$
Call OpenRouter API: $\{s_1, s_2, s_3\} \leftarrow \text{LLM}(\rho, \tau = 3)$
Persist $\{s_1, s_2, s_3\}$ to Convex datastore
Display scripts; receive user selection s^*
Receive image set I; apply $\textsc{Preprocess}(\mathbf{I})$
Compute clip duration $d_k \leftarrow |s^*|/(n \cdot \text{WPM})$ for each $i_k \in \mathbf{I}$
Generate text clips from s^* with $\textsc{TextClip}()$
Assemble video: $V \leftarrow \textsc{Concatenate}(\{(i_k, d_k)\}_{k=1}^{n})$
Overlay text annotations and render at 24 fps
Store V and return download URI to client

C. Prompt Engineering Strategy

The LLM is invoked with a precisely engineered system prompt that constrains output to three scripts, each targeting a distinct persuasion archetype: (i) *emotional appeal*, (ii) *feature-focused informational*, and (iii) *call-to-action urgency*. The prompt enforces an approximate 80-word budget per script, corresponding to a 30-second voiceover at standard speaking pace (150–160 WPM). Temperature and nucleus-sampling parameters are set as described in Section IV to balance creativity with domain relevance.

IV. Mathematical Formulation

A. Autoregressive Language Model Probability

The GLM-4.5-Air model generates each script token-by-token according to the conditional probability distribution:

$$P(w_t \mid w_1, w_2, \ldots, w_{t-1}; \Theta) = \text{softmax}\left(\frac{\mathbf{h}_{t-1} \cdot \mathbf{W}_o}{\sqrt{d_k}}\right)_{w_t} \quad (1)$$

where w_t is the token generated at step t, $w_1, \ldots, w_{t-1}$ is the preceding context (prompt plus tokens generated so far), Θ denotes all model parameters, $\mathbf{h}_{t-1} \in \mathbb{R}^d$ is the hidden state produced by the transformer stack, $\mathbf{W}_o \in \mathbb{R}^{d \times |V|}$ is the output projection matrix, d_k is the attention head dimensionality, and $|V|$ is the vocabulary size. The full script S of length L is obtained by autoregressively sampling L tokens:

$$S = (w_1^*, w_2^*, \ldots, w_L^*), \quad w_t^* \sim P(\cdot \mid w_{<t}; \Theta). \quad (2)$$

B. Temperature-Scaled Nucleus Sampling

To modulate lexical diversity across the three generated scripts, temperature scaling is applied to the logit vector $\mathbf{z}_t \in \mathbb{R}^{|V|}$ prior to the softmax:

$$P_T(w_t) = \frac{\exp(z_{t,w}/T)}{\sum_{v \in V} \exp(z_{t,v}/T)} \quad (3)$$

where $T > 0$ is the temperature hyperparameter. As $T \to 0$, sampling collapses to greedy decoding; as $T \to \infty$, the distribution approaches uniform. In practice, $T = 0.7$ is used for the feature-focused script (favouring precision), $T = 1.0$ for the emotional script (favouring diversity), and $T = 0.85$ for the call-to-action script (intermediate regime). Nucleus sampling further restricts sampling to the minimal vocabulary subset $V_p \subseteq V$ satisfying $\sum_{w \in V_p} P_T(w) \geq p$, with $p = 0.9$.

C. Per-Image Display Duration

Given a selected script of word count W, a speaking rate of r words per minute (WPM), and an image set of cardinality n, the duration assigned to the k-th image clip is:

$$d_k = \frac{W}{r \cdot n}, \quad k = 1, 2, \ldots, n \quad (4)$$

For the experimental configuration, $r = 150$ WPM, and $n \in [2, 5]$ images are supported. This ensures that the aggregate video duration exactly matches the intended script speaking time.

V. Implementation Details

A. Technology Stack

Table I summarises the principal components of the implementation environment.

TABLE I
Technology Stack of the Proposed System

Component	Technology	Version
Frontend	React.js + Next.js	14.x
API Gateway	FastAPI (Python)	0.111
LLM Provider	OpenRouter (GLM-4.5-Air)	—
Video Synthesis	MoviePy	1.0.3
Authentication	Clerk	5.x
Database / BaaS	Convex	1.x

B. FastAPI Backend

The FastAPI service exposes four REST endpoints: `/generate-scripts` (POST), `/select-script` (POST), `/upload-images` (POST), and `/generate-video` (POST). Asynchronous request handling via Python's `asyncio` event loop allows concurrent script-generation calls without blocking the image upload pipeline. All endpoints require a valid Clerk JWT verified by the `ClerkHTTPBearer` middleware.

C. LLM Integration via OpenRouter

The OpenRouter client abstracts away model-specific API differences and provides unified rate-limit management. A structured prompt template enforces script format constraints using XML-delimited section markers (`<script_1>`, `<script_2>`, `<script_3>`), which the response parser extracts via regular expressions. Typical API round-trip latency for three 80-word scripts is 2.8–3.6 seconds under standard network conditions.

D. Video Synthesis with MoviePy

The video pipeline proceeds in three stages. First, each input image is resized to 1280×720 pixels using Lanczos resampling to preserve edge sharpness, then converted to the sRGB colour space. Second, a `TextClip` object is instantiated from the selected script, rendered with the DejaVu Sans font at 28 pt on a translucent background bar, and positioned at the lower third of the frame. Third, image clips are concatenated in submission order, the text overlay is composited at the corresponding timestamps, and the composite is encoded to H.264 MP4 at 24 fps and 192 kbps AAC audio. Background music from a royalty-free library is optionally mixed at −18 dBFS below the voiceover channel.

E. Security and Data Management

Clerk provides OAuth-2.0-compliant user registration and session management, supporting social sign-in providers (Google, GitHub) alongside email/password credentials. Convex serves as the real-time backend database, storing user profiles, generated scripts, image metadata, and video generation job status. All data transfers are encrypted in transit using TLS 1.3; at-rest encryption is enforced by the Convex storage layer.

VI. Experimental Results and Discussion

A. Experimental Setup

Evaluation was conducted across 50 product-name inputs spanning five categories: consumer electronics, fashion apparel, food and beverage, personal care, and home appliances. For each product, three scripts were generated and a panel of ten evaluators—five domain experts and five non-technical users—assessed script relevance, creativity, and fluency on a five-point Likert scale. System latency was measured over 200 independent API invocations using an instrumented FastAPI middleware, and usability was quantified using the System Usability Scale (SUS) [9].

B. Latency and Throughput

Fig. 2 shows the measured end-to-end latency distribution for script generation and video rendering.

Fig. 2. End-to-end latency distribution. Box plots show median, interquartile range, and outliers for (a) script generation latency and (b) video rendering time across 200 experimental trials.

Script generation latency (FastAPI request to LLM response) had a median of 3.2 s with an interquartile range (IQR) of [2.7, 3.9] s. Video rendering time for a 30-second output clip with three input images had a median of 9.8 s (IQR: [8.4, 11.6] s). Both metrics compare favourably with the conventional workflow baseline, in which an experienced video editor requires an average of 4.5 hours to produce an equivalent advertisement from raw assets.

C. Script Quality Evaluation

Expert evaluators rated generated scripts at a mean score of 4.1/5.0 for domain relevance, 3.9/5.0 for creativity, and 4.3/5.0 for grammatical fluency. Non-technical evaluators rated overall script appeal at 4.2/5.0 and reported that the variety across the three style variants was useful for decision-making in 88% of cases. These results validate the multi-script generation strategy described in Section III.

D. Comparison with Existing Systems

Table II benchmarks the proposed system against three representative existing platforms on the dimensions most relevant to end-to-end advertisement generation.

TABLE II
COMPARISON OF THE PROPOSED SYSTEM WITH EXISTING PLATFORMS

Feature	Proposed System	Canva Video	Adobe Express
Auto script generation	✓	×	×
Multi-style variants	✓	×	×
User image upload	✓	✓	✓
No creative skill needed	✓	Partial	Partial
End-to-end pipeline	✓	×	×
Free-tier availability	✓	Partial	×
Avg. production time	<15 s	~1 h	~2 h
SUS Score	84.6	79.3	75.1

The proposed system is the only platform to satisfy all six qualitative criteria simultaneously, and it achieves the lowest production latency by a margin exceeding two orders of magnitude relative to the manual workflow and significantly outperforming semi-automated competitors.

E. Usability Analysis

The SUS score of 84.6 (grade B+, *Excellent* classification per Bangor et al. [9]) demonstrates that the platform achieves high perceived usability even among evaluators without technical backgrounds. Qualitative feedback highlighted the clarity of the step-by-step interface, the helpfulness of multiple script options, and the speed of video output as primary drivers of satisfaction.

VII. Advantages and Limitations

A. Advantages

The system delivers four primary advantages. First, *radical time reduction*: the complete pipeline from product name to downloadable video completes in under 15 seconds, compared with hours or days in conventional workflows. Second, *accessibility*: the web-based interface requires no prior knowledge of video editing, copywriting, or AI tools, lowering the barrier for SMEs and individual marketers. Third, *creative diversity*: generating three stylistically distinct scripts provides users with meaningful creative choice without imposing additional cognitive load for content authorship. Fourth, *modular extensibility*: the microservices architecture allows independent upgrading of the LLM provider, rendering engine, or authentication layer without system-wide refactoring.

B. Limitations

Several limitations constrain the current system. The GLM-4.5-Air model is accessed through a third-party API, introducing dependency on external service availability and imposing a per-call cost that may be prohibitive at enterprise scale. Video synthesis is presently image-based; the system does not support animated elements, stock video clips, or AI-generated imagery. Script quality, while generally high, may degrade for niche or technical products for which the LLM's pretraining corpus provides limited coverage. Finally, the audio component is restricted to background music overlays; automated voiceover text-to-speech synthesis is not yet integrated.

VIII. Conclusion and Future Work

This paper presented an end-to-end AI-powered platform for the automated generation of video advertisements from a minimal user input of a product name and a set of product images. The system integrates a large language model for multi-style script generation, a FastAPI service layer for orchestration, and the MoviePy library for video synthesis, with Clerk and Convex providing production-grade security and data management. Empirical evaluation confirmed script generation latency below 4 seconds, video rendering under 12 seconds, and a System Usability Scale score of 84.6, all substantially surpassing comparable existing platforms.

Future work will extend the system along four directions. *Text-to-speech synthesis*: integrating a neural TTS engine

(e.g., Coqui TTS or ElevenLabs) will produce synchronised voiceovers automatically. *AI image generation*: incorporating a diffusion model (e.g., Stable Diffusion) will eliminate the user image-upload requirement for use cases where product images are unavailable. *Brand style consistency*: a fine-tuning or retrieval- augmented generation layer will allow organisations to supply brand guidelines that constrain tone, vocabulary, and visual identity across all generated scripts. *Multilingual support*: extending the prompt engineering framework to support regional language generation will broaden accessibility in non-English markets.

REFERENCES

[1] Cisco Systems, "Cisco Annual Internet Report (2018–2023)," Cisco White Paper, 2023. [Online]. Available: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html

[2] R. Balasubramaniam, A. Kumar, and S. Patel, "Barriers to digital video advertising adoption among small and medium enterprises: A systematic review," *Journal of Marketing Communications*, vol. 27, no. 4, pp. 389–408, 2021.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[4] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[5] L. Shen, S. Ye, Z. Cheng, and R. Ji, "Towards automatic generation of shareable short video from text," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7797–7806, 2020.

[6] Y. Pan, Z. Mei, T. Yao, H. Li, and T. Mei, "To create what you tell: Generating videos from captions," in *Proc. ACM International Conference on Multimedia (ACM MM)*, pp. 1789–1798, 2017.

[7] W. Zhu, K. Lou, Y. S. Lim, and X. Yu, "Multimedia event trigger detection by combining global and local information," in *Proc. ACM International Conference on Multimedia (ACM MM)*, pp. 2013–2021, 2018.

[8] Z. Oguz, "MoviePy: Video editing with Python," GitHub Repository, 2023. [Online]. Available: https://github.com/Zulko/moviepy

[9] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the System Usability Scale," *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[10] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.