A RAG-Enhanced Multimodal Knowledge Synthesis System for Specialized Video Analysis and Management

1st Haozhe Zhang
School of Computing
Neusoft Institute
Foshan, China
735194816@qq.com

2nd Ruofei Wang*
School of Computing
Neusoft Institute
Foshan, China
wangruofei@nuit.edu.cn

3rd Jin Lin
School of Computing
Neusoft Institute
Foshan, China
linjin@nuit.edu.cn

4th Bowen Li
School of Computing
Neusoft Institute
Foshan, China
2481990466@qq.com

5th Hongwei Zhang
School of Computing
Neusoft Institute
Foshan, China
3352238132@qq.com

6th Bingkun Zhou
School of Computing
Neusoft Institute
Foshan, China
Python_zbk666@outlook.com

***Abstract*—Specialized knowledge videos are valuable resources for professional learning, yet converting long-form audiovisual content into accurate and reusable structured notes remains challenging due to cross-modal fragmentation, dense terminology, and hallucinations in general-purpose large language models. To address these issues, this paper presents a RAG-enhanced multimodal knowledge synthesis system for specialized video analysis and management. The system integrates video keyframe extraction, ASR-based transcription, temporal alignment, retrieval-augmented semantic calibration, and structured note generation within a layered Spring AI-based architecture. In addition, JDK 21 virtual threads are adopted to improve concurrency and service responsiveness. Experimental results show that the proposed method improves terminology accuracy from 72.0% to 95.4% after introducing the RAG-based calibration mechanism. It also achieves ROUGE-L of 0.86 and BLEU-4 of 0.82 in structured note generation, while maintaining stable throughput of about 40 requests per second under 200 concurrent users. These results demonstrate the effectiveness of the proposed system in multimodal understanding, knowledge alignment, and high-concurrency scientific video analysis.**

Keywords-multimodal knowledge generation; retrieval-augmented generation; Spring AI; video semantic extraction; professional terminology calibration; vector database

I. INTRODUCTION

As professional knowledge becomes increasingly digitized, audiovisual resources have become essential for documenting complex disciplinary systems and supporting individual knowledge construction[1]. However, extracting structured content from unstructured, information-dense videos remains challenging because of inadequate contextual modeling and semantic degradation in long-context processing[2]. General-purpose large language models, limited by their pretraining corpora, often produce domain-specific hallucinations, while end-to-end extraction methods struggle to ensure terminological accuracy and professional consistency[3].

From a software engineering perspective, this paper presents a RAG-enhanced multimodal system for specialized video analysis and knowledge synthesis. The framework integrates timestamp-synchronized multimodal perception, retrieval-augmented semantic calibration, and structured note generation within a layered Spring AI-based architecture, while JDK 21 virtual threads improve responsiveness and concurrency robustness.

The main contributions of this work are threefold. First, we propose a unified end-to-end framework for professional video understanding that combines video keyframe extraction, ASR-based transcription, temporal alignment, and structured knowledge generation. Second, we design a knowledge alignment mechanism that integrates vector retrieval with prompt-based semantic calibration to reduce terminology errors and suppress hallucinations in domain-specific note generation. Third, we implement the system with Spring AI and pgvector-based storage, enabling real-time streaming interaction under concurrent workloads. Experimental results further demonstrate significant improvements in terminology accuracy and note quality, confirming the practical value of the proposed system in knowledge-intensive video analysis scenarios.

II. RELATED WORK

A. Multimodal Synthesis and Scientific Analysis

In recent years, research on video content understanding and text generation has focused on "semantic extraction" and "system implementation". Li et al.[4] proposed a visual-language relational Transformer to improve cross-modal consistency in video-to-text generation, while Yi-Fan et al.[5] demonstrated the value of AI-driven video-to-text recognition in transforming unstructured audiovisual materials into reusable knowledge assets. At the system architecture level, Suryotrisongko et al.[6] developed a microservice backend using Spring Boot, providing a useful engineering reference for decoupling complex audiovisual analysis tasks.

However, existing work primarily addresses general information scenarios and does not fully resolve challenges specific to professional videos, such as dense terminology, strong coupling between visual presentation and spoken narration, and the structural complexity of long-duration content. Mythily et al.[7] highlighted Spring Boot's suitability for scalable backends, yet an integrated design covering the full pipeline from video parsing and multimodal alignment to structured output remains lacking.

Consequently, this paper unifies video keyframes, ASR-generated text, and knowledge fragments within a synchronized temporal framework to establish a comprehensive multimodal generation pipeline tailored for professional audiovisual analysis, as illustrated in Fig. 1.

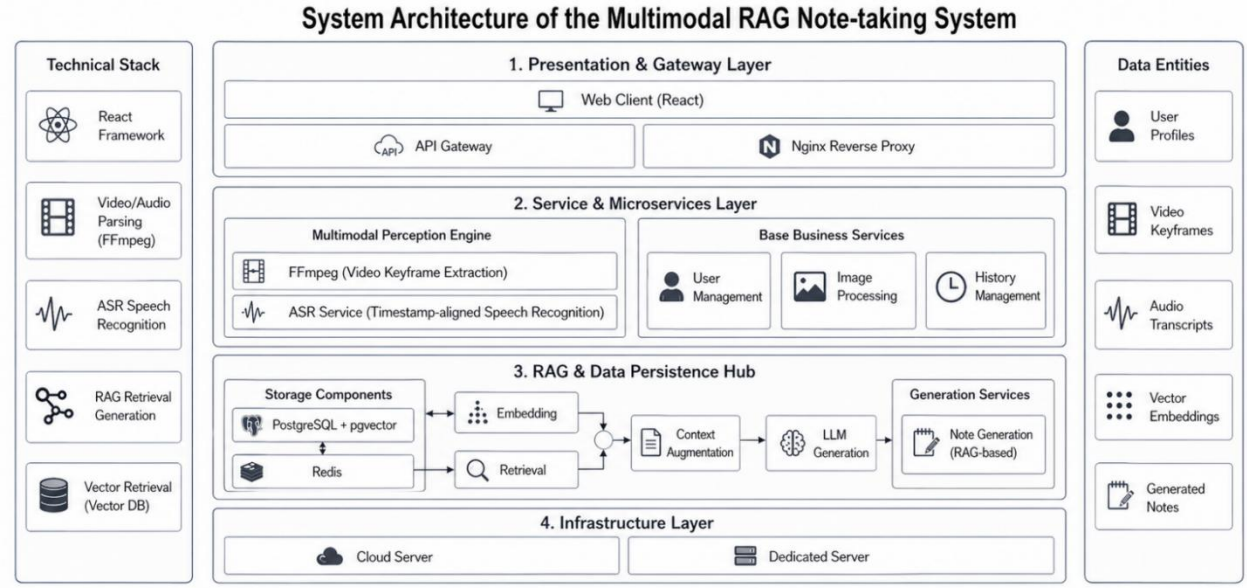

Figure 1. System Architecture Diagram

B. *Retrieval Enhancement Generation and Semantic Calibration*

In knowledge-intensive tasks, Lewis et al.[8] introduced the RAG framework, establishing the foundational "retrieval first, generation later" paradigm. Subsequent work by Zhang et al.[9] and Niinimaki et al.[10] examined the influence of index structures on generation accuracy from database and vector recall perspectives. At the Java engineering level, Walls[11] and Anthapu et al.[12] illustrated integration pathways for Spring AI and large model orchestration, offering practical foundations for model invocation and knowledge injection in domain-specific applications.

Despite these advances, limited attention has been paid to continuously correcting generated content using retrieval results in terminology-dense contexts. Anasuri et al.[13] emphasized the importance of closed-loop monitoring in AI-driven software design patterns for improving reliability. Inspired by prior work, this paper introduces a retrieval-calibrated generation loop to constrain outputs with authoritative domain knowledge and reduce semantic hallucinations.

C. *High-performance concurrency and engineering robustness in AI systems*

Suryotrisongko et al.[6] built a Spring Boot microservice backend guided by domain-driven design principles, demonstrating through REST interfaces and black-box testing that loosely coupled service decomposition enhances the stable execution of complex business processes. This architectural pattern offers a valuable reference for constructing robust AI-driven systems.

Nevertheless, most prior studies concentrate on traditional web services or generic AI applications, with insufficient exploration of I/O blocking, link instability, and feedback continuity when combining multimodal parsing, large language model invocation, and vector retrieval. Building upon recent work on cloud-native Java deployment[14], event-driven fallback orchestration for LLM-DBMS pipelines[15], and Agentic AI optimization for high-throughput microservices[16], this paper integrates JDK 21 virtual threads with asynchronous retry and service decoupling strategies. The resulting industrial-grade implementation balances throughput, call resilience, and semantic consistency for large-scale audiovisual knowledge extraction scenarios.

III. METHODS

The proposed system adopts a layered microservices architecture for end-to-end multimodal knowledge synthesis, spanning video parsing, multimodal alignment, retrieval-augmented generation, and structured note output. Built on JDK 21, Spring Boot 3.4.x, and Spring AI, the framework consists of four layers: Multimodal Perception and Preprocessing, Retrieval-Augmented Inference and Generation, Hybrid Storage and High-Concurrency Management, and Multimodal Synchronous Interaction and Output. As illustrated in Fig. 2, these components jointly transform raw scientific video content into structured, retrieval-grounded notes.

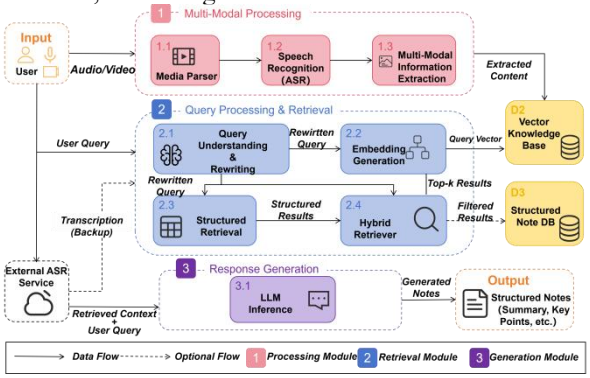

Figure 2. RAG-Based Multimodal Architecture

A. Multimodal Perception and Preprocessing Layer

This layer converts unstructured professional videos into standardized multimodal features for downstream processing. FFmpeg-based dynamic keyframe extraction is used to identify representative visual anchors according to scene transitions, while ASR services transcribe the audio stream into timestamped sentence segments. The extracted keyframes and ASR outputs are then aligned within a unified temporal coordinate system to establish a synchronized multimodal representation.

Formally, let the extracted keyframe sequence be denoted as $K = \{(k_i, t_i)\}_{i=1}^{N}$, where k_i represents the i-th keyframe and t_i denotes its timestamp. Let the ASR transcription sequence be represented as $S = \{(s_j, \tau_j^{start}, \tau_j^{end})\}_{j=1}^{M}$, where s_j is the j-th speech segment with start and end timestamps. The temporal alignment between visual and textual modalities is defined as

$$A(k_i) = \{s_j | \tau_j^{start} \le t_i \le \tau_j^{end}\} \quad (1)$$

where $A(k_i)$ denotes the set of speech segments aligned with keyframe k_i. This alignment establishes a unified multimodal representation for downstream retrieval and structured note generation.

B. Retrieval-Augmented Inference and Generation Layer

This layer coordinates LLM inference, vector retrieval, and the integration of external knowledge. Such a design is consistent with recent studies on combining vector databases with large language models for knowledge-intensive generation tasks[17]. DeepSeek-V3 is encapsulated through the Spring AI ChatModel interface to support unified model invocation and streaming generation. To reduce terminology errors and hallucinations in domain-specific videos, the system adopts a retrieval-augmented calibration loop in which segmented ASR text is embedded, matched against an authoritative knowledge base, and fused into the generation prompt.

Given a segmented query q, the embedding model projects it into a dense vector representation $e_q \in \mathbb{R}^{\mathbb{d}}$. Let the knowledge base contain document embeddings $E = \{e_1, e_2, ..., e_n\}$. The semantic relevance between the query and each candidate document is computed by cosine similarity:

$$\mathrm{sim}(e_q, e_i) = \frac{e_q \cdot e_i}{\|e_q\| \|e_i\|} \quad (2)$$

The retrieval module then selects the Top-k most relevant knowledge fragments as

$$R_k(q) = TopK_{e_i \in E} sim(e_q, e_i) \quad (3)$$

To enforce terminology consistency and reduce hallucinations, the final generation prompt is formulated as

$$P = [I, \mathrm{Q}, R_{k(q)}, C] \quad (4)$$

where I denotes the system instruction, Q denotes the current query or transcript segment, $R_k(q)$ denotes the retrieved authoritative context, and C denotes the semantic calibration constraint. The generated results are then organized into structured Markdown notes and streamed to the client in real time.

The overall workflow of retrieval-augmented inference and real-time note streaming is illustrated in Fig. 3. The generated results are then organized into structured Markdown notes and streamed to the client in real time.

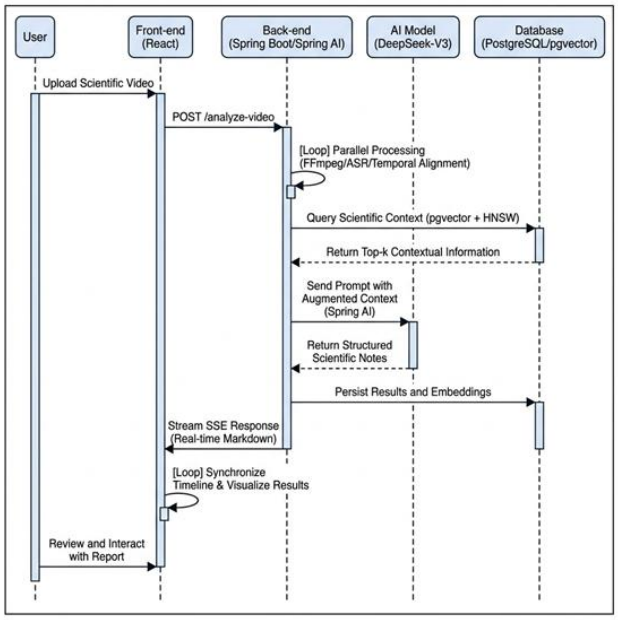

Figure 3. End-to-End Sequence Flow of Video Analysis, Retrieval Augmentation, and Real-Time Note Streaming

C. Hybrid Storage and High-Concurrency Management Layer

This layer provides storage and concurrency support for multimodal knowledge synthesis. PostgreSQL serves as the primary persistence medium, and the pgvector extension is used to manage dense vector representations together with relational business data in a unified database instance. In addition, Redis is employed to maintain asynchronous task states and transient recognition results, thereby supporting low-latency response in online processing scenarios.

To improve system responsiveness under I/O-intensive workloads, the framework adopts JDK 21 virtual threads instead of a conventional thread-pool model. This mechanism reduces scheduling overhead during blocking operations and provides stable support for concurrent multimodal processing. The effectiveness of this design is further validated in the experimental evaluation.

D. Multimodal Synchronous Interaction and Output Layer

This layer is responsible for presenting generated knowledge in a synchronized and user-interactive manner. Structured Markdown notes are streamed to the client through the Server-Sent Events (SSE) protocol, allowing users to inspect intermediate results before the full pipeline finishes. The final output supports LaTeX equations, code highlighting, and structured scientific note rendering.

To support multimodal interaction, note blocks, visual keyframes, and speech transcription nodes are bound to the video timeline, enabling bidirectional navigation between playback progress and structured knowledge. As shown in Fig. 4, the prototype interface integrates video playback, timeline navigation, multimodal evidence, and AI-synthesized research notes within a unified interaction space, thereby improving the interpretability and usability of generated scientific notes.

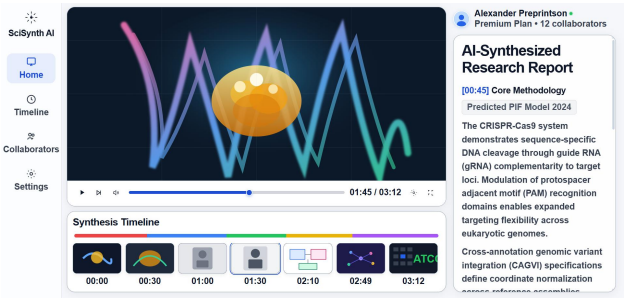

Figure 4. Prototype Interface for Timeline-Grounded Scientific Video Analysis and AI-Synthesized Research Notes

IV. EXPERIMENTS

To verify system stability, functional integrity, and high-concurrency capability in professional knowledge processing scenarios, this study constructed a comprehensive evaluation covering unit, functional, and performance testing. Experiments were conducted in a controlled LAN environment using an Intel Xeon Gold 6248R processor with 32GB memory, JDK 21, and Spring Boot 3.4.3. The test client connected via a 1000Mbps switch to eliminate public Internet interference, while DeepSeek-V3 API calls operated on a stable link with an average round-trip latency of 120 ± 10 ms. This setup enables quantitative evaluation of architectural feasibility and engineering robustness in intelligent knowledge synthesis scenarios[18].

A. Unit Testing

Core backend modules were validated with JUnit 5 and Mockito to verify structured output parsing, exception handling, retry logic, and Top-k retrieval correctness.

B. Functional Testing

Core functional completeness and usability were verified by simulating real-world user workflows. Test results are summarized in Table I.

TABLE I. KEY FUNCTIONALITY TEST RESULTS

Functional Module	Test Case	Expected Result	Test Result
Access Control	Regular user attempts to access admin backend.	Access denied with redirection to homepage.	pass
Core Video Streaming	Upload 15-minute technical lecture (.mp4).	Parsing pipeline starts, structured document generated within 30-45s.	pass
RAG Terminology Correction	Upload video with obscure terminology.	Terminology in generated notes corrected to authoritative definitions.	pass
Real-time Rendering	View notes with LaTeX formulas and code blocks.	Front-end successfully renders formulas and syntax-highlighted code.	pass
Historical record retrieval	Search by date range and keyword.	Returns historical parsed records with pagination support.	pass

The results confirm the completeness of the end-to-end workflow from video upload to structured knowledge persistence and real-time rendering.

C. Performance Testing

To evaluate robustness under combined high I/O and computational loads, stress tests were conducted on the core multimodal synthesis interface with varying concurrent user scales. Key performance indicators are shown in Table II.

TABLE II. STRESS TEST RESULT

Concurrent Users	Average Response Time	95th Percentile	Throughput
50	510	780	46.5
100	1020	1650	42.8
200	2050	3200	39.2

Due to JDK 21's efficient virtual-thread scheduling mechanism, server CPU utilization remained at 40%-50% under 200 concurrent connections, with no blocking phenomena from thread exhaustion or context switching. Average response time exhibited quasi-linear growth as concurrency increased, primarily constrained by cloud inference service queuing latency and API rate-limiting. Overall throughput converged to approximately 40 req/s, verifying stable multimodal parsing output in controlled environments and meeting engineering requirements for high-load professional knowledge processing scenarios.

D. Verification of AI-Generated Note Accuracy

Semantic accuracy in professional domains constitutes a critical effectiveness indicator. This study selected 50 high-information-density videos from MIT OpenCourseWare spanning chemistry, physics, and medicine as a benchmark test set. Standard NLP metrics ROUGE-L and BLEU-4 were employed to quantitatively evaluate structural consistency and semantic similarity. Results are presented in Table III.

TABLE III. NOTE-GENERATING QUALITY AND RAG PERFORMANCE METRICS

Assessment Task	Performance indicators	DeepSeek-V3 (without RAG)	This system (with RAG)
Text summarization quality	ROUGE-L Score	0.74	0.86
semantic overlap	BLEU-4 Score	0.68	0.82
Technical terminology correction	Terminology Accuracy	72.0%	95.4%
Search relevance	RAG Hit Accuracy	-	93.8%
Test dataset size	50 Research Videos	Validated	Validated

Experimental analysis shows the structured summary achieved ROUGE-L of 0.86, demonstrating accurate capture of core knowledge points and logical structure. BLEU-4 of 0.82 further validates fluency and professionalism in language organization. Notably, terminology accuracy for less common concepts in cutting-edge disciplines improved substantially, largely due to the suppression of semantic hallucinations through the integration of an authoritative knowledge base. By injecting retrieved Top-k context into prompts, the system achieves accurate terminology correction with traceable sources via RAG citation mechanism, satisfying stringent professional domain accuracy requirements. All evaluation metrics are calculated against manually annotated

references with independent verification and consistency validation.

E. Ablation Study

To further quantify the contribution of each core component, an ablation study was conducted by progressively disabling the key modules of the proposed framework, namely video analysis, RAG retrieval, and the calibration mechanism. The evaluation metrics include terminology accuracy, ROUGE-L, BLEU-4, multimodal alignment score, and user satisfaction, as summarized in Table IV.

Five system variants were evaluated: the full system, a version without video analysis, a version without RAG retrieval, a version without the calibration mechanism, and a baseline configuration using only ASR and direct LLM generation. All variants were tested on the same 50-video benchmark to ensure fair comparison across configurations. The evaluation metrics include terminology accuracy, ROUGE-L, BLEU-4, multimodal alignment score, and user satisfaction.

TABLE IV. ABLATION STUDY RESULTS

Configuration	Term Acc.	ROUGE-L	BLEU-4	MM Align.	User Sat.
Full system	95.4%	0.86	0.82	0.91	4.6/5.0
Without video analysis	94.8%	0.84	0.81	0.52	3.8/5.0
Without RAG retrieval	72.0%	0.74	0.68	0.88	3.2/5.0
Without calibration	78.5%	0.78	0.73	0.89	3.5/5.0
Baseline (ASR + LLM)	68.3%	0.71	0.65	0.48	2.9/5.0

The ablation results show that RAG retrieval is the most critical component for terminology correction and semantic quality: removing it reduces terminology accuracy from 95.4% to 72.0% and lowers ROUGE-L and BLEU-4 from 0.86/0.82 to 0.74/0.68. Removing the calibration mechanism also causes a clear decline, indicating that retrieval alone is insufficient without prompt-level semantic guidance. In contrast, removing video analysis mainly affects multimodal alignment rather than terminology accuracy. These results confirm that video analysis, retrieval augmentation, and semantic calibration jointly contribute to the overall effectiveness of the proposed framework.

V. DISCUSSION

The experimental results confirm that the proposed system effectively improves the quality of structured note generation for specialized videos, particularly in terminology-intensive scenarios. The substantial increase in terminology accuracy from 72.0% to 95.4% demonstrates the importance of retrieval-augmented semantic calibration in constraining large language model outputs with authoritative domain knowledge. In addition, the ablation study shows that RAG retrieval contributes most directly to terminology correction and semantic fidelity, while video analysis mainly improves multimodal alignment and user interaction quality.

From a systems perspective, the proposed architecture also proves practical from an engineering perspective. The integration of Spring AI, PostgreSQL/pgvector, Redis, and JDK 21 virtual threads enables stable end-to-end processing, real-time streaming interaction, and robust performance under concurrent workloads. These results indicate that the framework is not only effective in knowledge synthesis quality but also feasible for deployment in real-world knowledge-intensive applications.

Nevertheless, several limitations remain. First, the current evaluation dataset is limited to 50 benchmark videos from selected scientific domains, which may not fully capture the diversity of professional audiovisual content. Second, although the proposed method significantly reduces terminology errors, the quality of generated notes still depends on the coverage and authority of the external knowledge base. Third, the present framework focuses mainly on note generation and synchronized review, while deeper reasoning, cross-video knowledge linking, and user-adaptive summarization remain underexplored.

Future work will therefore focus on expanding the evaluation dataset, enriching the domain knowledge repository, and improving multimodal reasoning across longer and more diverse videos. In addition, more advanced interaction mechanisms and adaptive retrieval strategies will be investigated to further enhance the interpretability, scalability, and decision-support capability of the system.

VI. CONCLUSIONS

This paper presented a RAG-enhanced multimodal knowledge synthesis system for specialized video analysis and management. By integrating Spring AI, multimodal perception, temporal alignment, retrieval-augmented semantic calibration, and structured note generation within a layered architecture, the proposed system effectively improves the conversion of complex audiovisual resources into structured and reusable professional knowledge. Experimental results show that the system improves terminology accuracy from 72.0% to 95.4%, achieves ROUGE-L of 0.86 and BLEU-4 of 0.82, and maintains stable throughput under concurrent workloads.

These results confirm that the proposed framework is effective not only in reducing hallucinations and improving terminology consistency, but also in delivering strong engineering robustness for knowledge-intensive applications. Future work will focus on larger-scale cross-domain evaluation, richer knowledge base construction, and deeper multimodal reasoning to further enhance the system's interpretability, scalability, and practical value.

ACKNOWLEDGMENT

This work was supported by the 2025 National Undergraduate Innovation and Entrepreneurship Competition project "ZhiGeng Intelligent Supervision Central System" (Project No.: S202512574005).

REFERENCES

[1] I. Osunbunmi, B. P. -J. Shih, S. Cutler and C. Perez, "(WIP) Developing a Generative AI Literacy Module for Engineering Graduate Teaching Assistants: A Model for Engineering Education," 2025 IEEE Frontiers in Education Conference (FIE), Nashville, TN, USA, 2025, pp. 1-5, doi: 10.1109/FIE63693.2025.11328161.

[2] Andy Peng; Alex Strick van Linschoten; Duarte O.Carmo, DeepSeek in Practice: From basics to fine-tuning, distillation, agent design, and prompt engineering of open source LLM, Packt Publishing, 2025.

[3] G. Solaimalai, "AI-Driven Enterprise Architecture for Scalable Cloud Applications," 2025 International Conference on Recent Innovation in Science Engineering and Technology (ICRISET), CHENNAI, India, 2025, pp. 1-8, doi: 10.1109/ICRISET64803.2025.11252450.

[4] T. Li, H. Wang, Q. Li and Z. Ni, "Vision-Language Relational Transformer for Video-to-Text Generation," in IEEE Transactions on Multimedia, vol. 27, pp. 4584-4596, 2025, doi: 10.1109/TMM.2025.3535394.

[5] L. Yi-Fan, L. M. Irfan, and H. Wu-Yuin, "Enhancing EFL Writing Through AI-Driven Video-to-Text Recognition in Authentic Learning Contexts," in 2025 IEEE International Conference on Advanced Learning Technologies (ICALT), 2025, pp. 166-168.

[6] H. Suryotrisongko, D. P. Jayanto and A. Tjahyanto, "Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot," Procedia Computer Science, 2017.

[7] M. Mythily et al., "An analysis of the significance of Spring Boot in the market," 2022.

[8] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[9] Y. Zhang, S. Liu and J. Wang, "Are There Fundamental Limitations in Supporting Vector Data Management in Relational Databases? A Case Study of PostgreSQL," 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 2024, pp. 3640-3653, doi: 10.1109/ICDE60146.2024.00280.

[10] M. Niinimaki, S. R. Shrestha and A. Udofia, "Comparison of Vector Database Management Systems for Retrieval Augmented Generation," 2025 29th International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2025, pp. 13-17, doi: 10.1109/ICSEC67360.2025.11298006.

[11] C. Walls, Spring AI in Action, Manning Publications, 2025.

[12] Ravindranatha Anthapu; Siddhant Agarwal; Dr. Jim Webber; Dr. Julian Risch, Building Neo4j-Powered Applications with LLMs: Create LLM-driven search and recommendations applications with Haystack, LangChain4j, and Spring AI, Packt Publishing, 2025.

[13] S. Anasuri, G. P. Rusum and K. K. Pappula, "AI-Driven Software Design Patterns: Automation in System Architecture," Int. J. Artificial Intelligence, Data Science, and Machine Learning, 2023.

[14] A. Aladiyan, "Integrating Spring Boot with Cloud Services for Scalable Java Applications with the test results of AI Implementation," 2024 International Conference on Communication, Computing and Energy Efficient Technologies (I3CEET), Gautam Buddha Nagar, India, 2024, pp. 17-22, doi: 10.1109/I3CEET61722.2024.10993799.

[15] E. Shahab and S. Taghipour, "Resilient LLM-DBMS Pipelines via Event-Driven Fallback Orchestration," 2025 IEEE International Conference on Collaborative Advances in Software and COmputiNg (CASCON), Toronto, ON, Canada, 2025, pp. 627-630, doi: 10.1109/CASCON66301.2025.00113.

[16] S. Padhi, "Agentic AI for Microservices: Autonomous Optimization of High-Volume Financial Transactions in Cloud Native Environments," in 2025 IEEE Cloud Summit, 2025, pp. 165-169, doi: 10.1109/Cloud-Summit64795.2025.00033.

[17] P. Sharma, A. S. Pundir, G. D. Singh and R. Gupta, "Integration of Vector Databases with Large Language Models (LLMs)," 2025 International Conference on Next Generation Information System Engineering (NGISE), Ghaziabad, Delhi (NCR), India, 2025, pp. 1-5, doi: 10.1109/NGISE64126.2025.11085087.

[18] J. Shah, S. R. Gade, D. Ponna and K. A. Patel, "Cloud-Native AI and Generative AI on AWS: A Systematic Review and a Proposed Unified Model," 2026 7th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), Goathgaun, Nepal, 2026, pp. 1710-1717, doi: 10.1109/ICMCSI67283.2026.11412536.