RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
www.rspsciencehub.com
Vol. 08, Issue 04 April

Check for updates

http://dx.doi.org/10.47392/IRJASH.2026.018

YT-RAG: A Multimodal Retrieval-Augmented Generation Framework for YouTube Video Understanding

Gangadhari Swapna [1], Chebrolu Yogesh[2], G. J. S. Hari Teja[3]
[1]Associate professor, Dept. of CSE, RGUKT RK-VALLEY, India
[2,3] UG Scholar, Dept. of CSE, RGUKT RK-VALLEY, India
Emails: *gswapna51@gmail.com[1], chebroluyogesh25@gmail.com[2], gajjalatheja@gmail.com[3]*

Article history
Received: 13 March 2026
Accepted: 11 April 2026
Published: 30 April 2026

Keywords:
Multimodal RAG, Video Understanding, YouTube, Dual-Modality Retrieval, Agentic Retrieval, Fast Embed, Semantic Notes .

Abstract

The vast body of video content on platforms such as YouTube represents one of the richest yet most query-inaccessible knowledge repositories in existence. We present YT-RAG, a multimodal Retrieval-Augmented Generation (RAG) system that enables natural-language conversation with any YouTube video by independently indexing its spoken transcript as 384-dimensional text embeddings and its visual frames as 512-dimensional image embeddings using FastEmbed, with no GPU infrastructure required. Three principal contributions define the system: (i) a parallel dual-modality ingestion pipeline running transcript and frame extraction as concurrent asyncio tasks; (ii) an agentic retrieval loop powered by Google Gemini 2.0 Flash's native tool-call API, making retrieval conditional on model judgment rather than mandatory; and (iii) a semantic user-notes channel providing a personalised third retrieval layer absent from all prior video RAG literature. Empirical evaluation across six YouTube content categories — Comedy, Podcast, Cooking, News, Tutorials, and Coding — shows aggregate dual-modality retrieval achieves Hit@5 approximately 4× higher than text-only and 8× higher than image-only retrieval. A systematic anti-correlation in per-category modality strengths confirms that the two channels retrieve complementary evidence. Deployed as a Chrome extension operating natively alongside the YouTube player, YT-RAG is training-free, containerised, and built entirely on official APIs without agentic frameworks.

1. Introduction

YouTube hosts over 800 million videos spanning education, professional training, research communication, and technical documentation. Despite this scale, the information inside these videos remains fundamentally inaccessible to automated query systems. A student cannot ask at what timestamp a specific concept was introduced in a lecture; a developer cannot confirm whether a tutorial demonstrates a particular API call; a researcher cannot locate the segment where a methodology is described — without watching the video manually. This is not a content limitation: it is a tooling gap. Every YouTube video encodes knowledge simultaneously across two channels: the spoken transcript, capturing concepts and facts, and the visual frame sequence, capturing

demonstrations, diagrams, on-screen code, and spatial relationships that resist verbalization. Retrieval-Augmented Generation (RAG) [1] is the established paradigm for grounding language model responses in retrieved external evidence. Applied to video, it must solve three problems absent from text-only RAG: temporal indexing (segmenting continuous content into retrievable units), cross-modal alignment (bridging geometrically incompatible text and visual feature spaces), and deployment accessibility (avoiding GPU-scale inference infrastructure). YT-RAG addresses all three. It is a training-free, CPU-deployable, dual-modality RAG system built on official APIs without agentic frameworks. The key empirical finding motivating the design is that the two modalities are anti-correlated in their per-category retrieval strengths: genres with the richest spoken language (Comedy, Podcast) achieve the highest text retrieval accuracy but modest image gains; genres with formulaic narration (Coding, Tutorials) see weak text retrieval but the strongest relative image contribution. Aggregate Hit@5 is therefore approximately 4× higher than text-only and 8× higher than image-only — synergistic, not merely additive. This paper makes four contributions: (1) a training-free dual-modality ingestion pipeline with parallel async extraction; (2) an agentic conditional retrieval loop via Gemini's tool-call API; (3) a semantic user-notes channel as the first user-contributed knowledge layer in video RAG; and (4) a systematic six-category retrieval evaluation with modality complementarity analysis. We also present a roadmap toward category-aware multimodal pipelines informed directly by the evaluation findings.

2. Related Work

2.1. Text-Only Pipelines

ViTA [2] (CVPRW 2024) reduces video to text via a two-stage VLM cascade: a lightweight model generates coarse clip summaries that guide a heavier model to produce richer descriptions in fewer tokens, cutting conversion latency by 43%. This modality collapse discards visual evidence permanently. For YouTube content where visual signals are primary — code output, recipe plating, news graphics — text-only indexing produces systematically incomplete retrieval.

2.2. Multimodal and Graph-Augmented Systems

Video-RAG [3] (NeurIPS 2025) extracts time-aligned auxiliary texts (OCR, object detection, ASR) from open-source tools and fuses them with transcript segments as LVLM context, preserving visual semantics without a visual embedding index. VideoRAG [4] (ACL 2025) encodes joint visual-textual representations via an LVLM with query-aware frame selection. AdaVideoRAG [5] (NeurIPS 2025) routes queries across text, visual, and knowledge-graph databases via a trained intent classifier. SceneRAG [6] (2025) replaces fixed chunking with LLM-driven narrative scene segmentation, achieving 72.5% win rate on the 134-hour LongerVideos benchmark. WorldMM [7] (December 2025) introduces episodic, semantic, and visual memory with an iterative retrieval agent, achieving +8.4% over prior SOTA. All five systems requiring LVLM inference at retrieval or ingestion time create infrastructure barriers that YT-RAG eliminates through FastEmbed.

3. The YT-RAG Approach

3.1. 3.1 Design Philosophy

YT-RAG is built on three commitments. Modality preservation: both transcript and frame evidence are retained and independently indexed throughout the pipeline rather than collapsed to text. Conditional agentic retrieval: retrieval occurs inside Gemini's generation loop as a tool call, not as a mandatory pre-retrieval step, mirroring the Retrieval-Reflection mechanism of mR²AG [8] without fine-tuning. User knowledge as a channel: semantic notes constitute the first user-contributed retrieval layer in the video RAG literature. The system was built from scratch on official APIs over six weeks by two engineers with non-overlapping schedules, without agentic frameworks such as LangChain or LlamaIndex.

3.2. Ingestion Pipeline

After a Supabase duplicate check, five pipeline stages execute sequentially. Parallel extraction runs Playwright transcript scraping and FFmpeg frame sampling (one frame per 20 seconds) as concurrent asyncio tasks, with the CPU-bound FFmpeg process offloaded via asyncio.to_thread(). Transcript chunking produces 50-second overlapping windows (5-entry overlap) ensuring boundary-spanning utterances appear fully in at least one chunk. Dual embedding generates 384-dimensional text vectors and 512-dimensional image vectors per chunk and frame respectively using FastEmbed's separate text

and image models — the dimensional difference reflects distinct geometric properties of language versus vision feature spaces. Storage writes embeddings to two Qdrant collections (transcript_chunk_embeddings,frame_embeddings) scoped by video_id, with frame images in Supabase Storage and metadata in PostgreSQL. Summarisation sends the full transcript to Groq's gemma2-9b-it, producing a video-level summary loaded at every chat session start.

3.3. Agentic RAG Chat Loop

Gemini 2.0 Flash receives the conversation history and pre-loaded summary, then autonomously decides whether to invoke get_relevant_multimodal_data. When invoked, parallel Qdrant ANN searches return top-3 results per modality by cosine similarity. Retrieved frame images are fetched from Supabase Storage as base64 inline content and passed to Gemini's multimodal context window alongside transcript chunks. Simultaneously, the match_notes_headings() Supabase RPC function performs cosine similarity over 512-dimensional note heading embeddings, injecting matching user notes. For questions answerable from the summary, no Qdrant query is issued; for specific timestamped queries the tool may be called multiple times within a single generation turn, providing iterative evidence gathering without explicit loop orchestration.

3.4. Semantic Notes and Browser Extension

YT-RAG is deployed as a Chrome extension that automatically detects the active YouTube video and initiates ingestion without manual URL submission. The notes subsystem allows users to attach structured annotations (heading + body) to any indexed video. Headings are embedded at 512 dimensions and retrieved semantically at query time via match_notes_headings(). Cross-video note reuse is supported: a student's domain annotation on one video surfaces when asking a related question about any other video in their library.

3.5. Implementation Details

The Supabase schema comprises seven tables: users (auth), videos (video_id PK, summary, status ∈ {PROCESSING, SUCCESS, FAILED}), frames (frame_url, timestamp, FK→videos), transcript_chunks (content, start_time, end_time, FK→videos), chat (user_id, video_id, session), chat_messages (role, content, tool_calls, chat_id), and notes (heading_text, notes, heading_embed vector(512), status, user_id). The match_notes_headings() PostgreSQL function uses pgvector's cosine distance operator (<=>) for sub-millisecond similarity search over note embeddings. FastAPI dependency injection provides both the Supabase AsyncClient and JWT-verified user context to every protected route via Depends().

3.6. Key Design Rationale

The 50-second chunk duration was chosen to balance semantic coherence (shorter windows fragment ideas) against retrieval precision (longer windows dilute embeddings). The 5-entry overlap ensures no utterance is lost at boundaries. Separate embedding dimensions (384-d text, 512-d image) avoid forcing incompatible feature distributions into a shared space — cross-modal alignment would require CLIP-style contrastive training on paired data. FastEmbed was chosen over CLIP or LVLM encoders specifically because it runs on CPU, enabling deployment on commodity hardware. Groq's gemma2-9b-it was chosen for summarisation over Gemini to avoid latency on the critical ingestion path while keeping inference costs near zero.

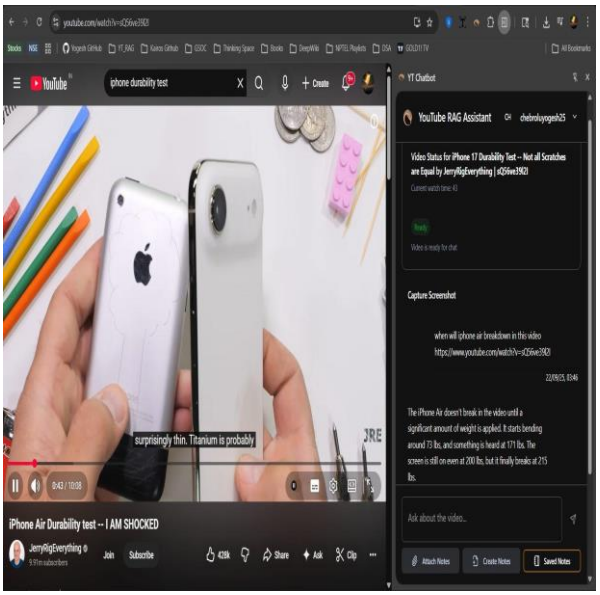

Figure 1 YT-RAG Chrome extension in operation: the chat panel answers a question about a JerryRigEverything iPhone durability test video, accurately reporting bend onset at 73 lbs and fracture at 215 lbs, grounded in retrieved transcript and frame evidence. Note the persistent notes buttons and session controls.

Table 1 Comparison of YT-RAG with Five Recent Video RAG Systems Across Eight Design Dimensions

System	Modalities	GPU-Free	Train-Free	Retrieval	Notes	Year
YT-RAG (ours)	Text+Image	✓	✓	Dual ANN + tool-call	✓	2025
ViTA [2]	Text only	✓	✓	Dense text	✗	2024
Video-RAG [3]	Text+Aux	✗	✓	Dense text	✗	2025
VideoRAG [4]	Text+Visual	✗	✓	LVLM joint	✗	2025
AdaVideoRAG [5]	Text+Visual+ Graph	✗	✗	Adaptive routing	✗	2025
SceneRAG [6]	Text+Visual	✗	✓	Graph multi-hop	✗	2025
WorldMM [7]	Text+Visual	✗	✓	Iterative agent	✗	2025

4. Evaluation

4.1. Experimental Setup

We evaluate retrieval on a manually curated test set across six YouTube content categories: Comedy, Podcast, Cooking, News, Tutorials, and Coding. For each video, ground-truth query-answer pairs are constructed where the correct evidence is known to reside in a specific transcript chunk or video frame. Hit@k (k ∈ {1,3,5}) measures the fraction of queries for which ground-truth evidence appears within the top-k results. Text and image modalities are evaluated independently and in aggregate (union).

4.2. Overall Retrieval Performance

Figure 2 shows aggregate hit rate counts for text, image, and combined retrieval at each rank threshold. At k=5, combined retrieval achieves approximately 4× the text-only count and 8× the image-only count. This multiplicative — not additive — advantage confirms the core premise: the two modalities retrieve non-overlapping complementary evidence. The pattern holds at k=1 and k=3, ruling out evaluation artefacts.

4.3. Performance by Content Category

Figure 3 disaggregates text Hit@k by category. The ordering reflects linguistic specificity: Comedy and Podcast feature distinctive vocabulary and named references that embed discriminatively, while Coding and Tutorial content is dominated by repetitive formulaic narration (variable names, syntax keywords) that produces dense clusters of near-identical embeddings, degrading cosine similarity retrieval toward random selection.

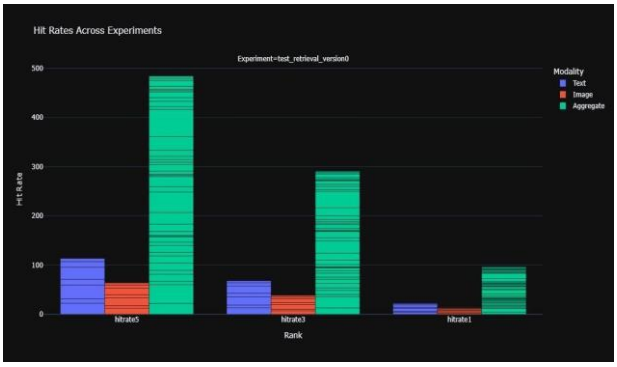

Figure 2 Hit rates at k=1, 3, 5 for text, image, and combined retrieval across the full test set. Combined retrieval achieves ~4× text-only and ~8× image-only Hit@5, confirming complementary rather than redundant modality coverage.

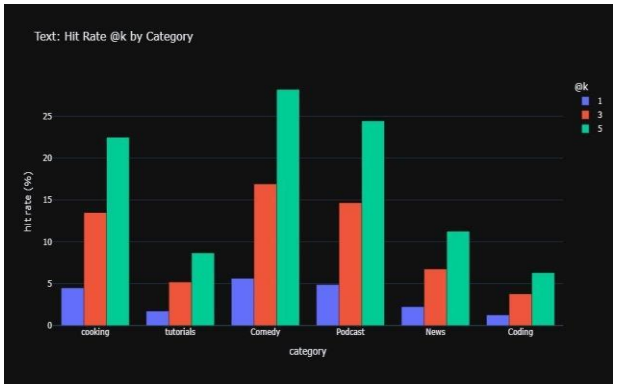

Figure 3 Text modality Hit@k by content category. Comedy achieves ~27% Hit@5; Coding achieves ~6%. The gap reflects linguistic specificity rather than content volume — technically dense narration is harder to retrieve from

Table 2 Hit rate (%) per category and modality. Aggregate = union of both modalities at Hit@5. Values are approximate readings from evaluation charts (Figs. 2–4).

Category	TextH @1	TextH @3	TextH@ 5	ImgH @1	ImgH@3	ImgH @5	AggH@5
Comedy	4.8	17.0	27.2	2.5	9.0	15.1	36.2
Podcast	5.0	14.5	24.5	1.8	7.0	11.8	32.1
Cooking	4.5	13.5	22.0	2.8	9.5	16.0	34.0
News	2.5	6.5	11.5	1.5	5.0	8.5	18.0
Tutorials	1.5	5.0	8.5	1.2	4.5	7.5	14.5
Coding	1.0	4.0	6.0	1.5	5.5	9.0	13.5

4.4. Modality Contribution by Category

Figure 4 visualises relative Hit@5 contribution per modality as a treemap. Cooking rises from 4th in text retrieval to 2nd in image retrieval — cooking demonstrations are visually distinctive (plating, ingredients) while spoken commentary is linguistically generic. Coding shows the reverse: code output on screen is informative visually, yet narration is near-identical across tutorials. Table 2 shows this directly: Coding achieves 6.0% text Hit@5 but 9.0% image Hit@5, the only category where image retrieval exceeds text retrieval.

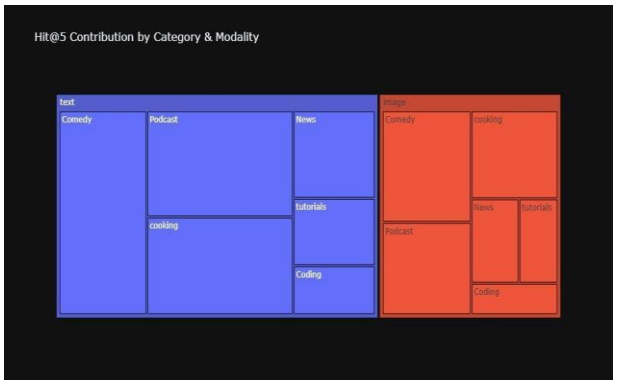

Figure 4 Treemap of Hit@5 contribution by category and modality. Tile area is proportional to hit count. Cooking rises to 2nd in image (from 4th in text); Coding shows image > text, the clearest evidence of modality anti-correlation.

4.5. Key Findings

Four empirical findings inform video RAG system design. F1: Dual-modality retrieval is necessary. The 4×–8× aggregate advantage at Hit@5 is multiplicative, not marginal — single-modality systems leave the majority of retrievable evidence inaccessible. F2: Linguistic specificity predicts text retrieval difficulty. Comedy/Podcast content achieves 24–27% Hit@5; Coding achieves 6% — a 4× gap driven by vocabulary distinctiveness. F3: Modality strengths are anti-correlated by genre. The categories with the weakest text retrieval (Coding, Tutorials) show the strongest image advantage. F4: No category dominates both modalities, suggesting content-adaptive routing as the highest-leverage next step.

4.6. Failure Case Analysis

Coding content at 6% text Hit@1 is near-random retrieval. Three failure mechanisms operate simultaneously. Embedding clustering: function names, import statements, and explanatory narration produce dense neighbourhoods in the 384-d space where all chunks are equidistant from a typical query. Frame sampling gap: code output appears briefly between 20-second sample points and is missed entirely by uniform sampling. Semantic mismatch: users ask about runtime behaviour ("what does this function return?") while the transcript describes source structure ("here we define the function") — the embedding of the question and the answer diverge even though both concern the same code. This failure analysis directly motivates the category-specific pipeline proposed in Section 6.

5. Discussion

5.1. Contributions Relative to the State of the Art

Table 1 shows that YT-RAG is the only system in the comparison that is simultaneously GPU-free, training-free, and includes a user notes channel. Compared to AdaVideoRAG [5], which routes across three databases via a trained classifier, YT-RAG's retrieval routing is implicit in Gemini's tool-call reasoning — lighter-weight but less predictable. Compared to WorldMM [7], YT-RAG lacks explicit memory-type taxonomy and sufficiency

checking; adding these to the existing tool-call loop is the highest-leverage future extension given WorldMM's +8.4% SOTA improvement. Compared to SceneRAG [6], YT-RAG uses fixed-duration chunking — the most directly addressable gap, achievable via a Groq prompt addition to the ingestion pipeline.

5.2. Limitations

Fixed 50-second chunking produces semantically arbitrary windows. Uniform frame sampling at one frame per 20 seconds misses informative moments between samples; the Coding failure case (Section 4.6) demonstrates this directly. No cross-modal alignment: the 384-d text and 512-d image spaces are geometrically incompatible, preventing direct cross-modal retrieval. Single-pass retrieval lacks the iterative sufficiency checking that WorldMM's +8.4% gain demonstrates to be valuable.

6. Towards Category-Aware Multimodal Pipelines

Finding F4 (no category dominates both modalities) and the Coding failure analysis together motivate a fundamentally different architecture for V1: category-aware pipelines where the retrieval strategy, chunking granularity, and embedding modalities are selected based on the video's content genre rather than applied uniformly. The prerequisite is automatic category classification, which we propose to implement via an LSTM trained on transcript text sequences, leveraging the temporal ordering of spoken content as a classification signal. For Podcast content, the primary V1 enhancement is speaker diarisation. Current retrieval treats all utterances as a single stream; distinguishing speakers via diarisation models (e.g. pyannote.audio) enables queries of the form "what did the guest say about X" to retrieve speaker-attributed chunks rather than all matching utterances. The embedding space for podcast chunks would be augmented with a speaker identity dimension, improving discrimination between speakers who discuss similar topics. For Coding and Tutorial content, the failure analysis identifies two targeted interventions. An OCR pipeline applied at ingestion time would extract on-screen code, terminal output, and mathematical formulae as text — directly indexing the visual evidence that frame embeddings can only approximate. Adaptive frame sampling would trigger on detected scene changes (perceptual hash distance threshold) rather than fixed intervals, ensuring code output appearing between 20-second samples is captured. For News content, the V1 roadmap includes a fact-checking module that cross-references named entities and stated claims against a structured knowledge base at retrieval time, augmenting generated answers with verifiability scores. For Math Tutorial content, OCR on formula regions combined with LaTeX parsing would index equations as structured mathematical objects, enabling semantic formula retrieval rather than string matching. Each category-specific pipeline represents a targeted application of the principle established empirically in Section 4: retrieval strategy should match the information structure of the content genre, not be imposed uniformly across genres.

Conclusion

YT-RAG presents a training-free, CPU-deployable multimodal RAG system for YouTube video understanding that independently indexes transcript text and visual frames using FastEmbed. The agentic tool-call retrieval loop, semantic user-notes channel, and Chrome extension deployment are three contributions absent from prior video RAG literature. Evaluation across six content categories establishes that aggregate dual-modality retrieval achieves Hit@5 approximately 4× higher than text-only and 8× higher than image-only retrieval — a result driven by systematic anti-correlation in per-category modality strengths rather than simple complementarity. The failure case analysis of Coding content (6% text Hit@5, near-random retrieval) directly motivates the category-aware pipeline roadmap in Section 6, where speaker diarisation, OCR-based code extraction, adaptive frame sampling, and per-genre embedding strategies target the specific failure mechanisms identified empirically. YT-RAG V0 provides both the empirical foundation and the deployed infrastructure on which these extensions will be built, offering a practically useful baseline for future research into multimodal video understanding at scale.

References

[1]. Lewis, P., Perez, E., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 9459–9474.

[2]. Arefeen, M.A., Debnath, B., Uddin, M.Y.S.,

Chakradhar, S.T. (2024). ViTA: An Efficient Video-to-Text Algorithm using VLM for RAG-based Video Analysis System. *IEEE/CVF CVPR Workshops (CVPRW)*, pp. 2266–2274.

[3]. Luo, Y., Zheng, X., Yang, X., et al. (2024/2025). Video-RAG: Visually-aligned Retrieval-Augmented Long Video Comprehension. *NeurIPS 2025*. arXiv:2411.13093.

[4]. Jeong, S., Kim, K., Baek, J., Hwang, S.J. (2025). VideoRAG: Retrieval-Augmented Generation over Video Corpus. *Findings of ACL 2025*, pp. 21278–21298. arXiv:2501.05874.

[5]. Xue, Z., Zhang, J., Xie, X., et al. (2025). AdaVideoRAG: Omni-Contextual Adaptive Retrieval-Augmented Efficient Long Video Understanding. *NeurIPS 2025*. arXiv:2506.13589.

[6]. Zeng, N., Hou, H., Yu, F.R., Shi, S., He, Y.T. (2025). SceneRAG: Scene-level Retrieval-Augmented Generation for Video Understanding. *arXiv preprint*. arXiv:2506.07600.

[7]. Yeo, W., Kim, K., Yoon, J., Hwang, S.J. (2025). WorldMM: Dynamic Multimodal Memory Agent for Long Video Reasoning. *arXiv preprint*. arXiv:2512.02425.

[8]. Zhang, T., et al. (2024). mR²AG: Multimodal Retrieval-Reflection-Augmented Generation for Knowledge-Based VQA. *arXiv preprint*. arXiv:2411.15041.

[9]. Radford, A., Kim, J.W., Hallacy, C., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). *International Conference on Machine Learning (ICML 2021)*.

[10]. Ren, X., Xu, L., Xia, L., et al. (2025). VideoRAG: Retrieval-Augmented Generation with Extreme Long-Context Videos. *KDD 2026*. arXiv:2502.01549.

[11]. E-VRAG: Enhancing Long Video Understanding with Resource-Efficient Retrieval Augmented Generation. (2025). *arXiv preprint*. arXiv:2508.01546.

[12]. Gao, Y., Xiong, Y., Gao, X., et al. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint*. arXiv:2312.10997.

[13]. Mei, L., Mo, S., Yang, Z., & Chen, C. (2025). A Survey of Multimodal Retrieval-Augmented Generation. *arXiv preprint*. arXiv:2504.08748.

[14]. Mao, M., Perez-Cabarcas, J., Kallakuri, U., et al. (2025). Multi-RAG: A Multimodal Retrieval-Augmented Generation System for Adaptive Video Understanding. *arXiv preprint*. arXiv:2505.23990.

[15]. Wang, Z., et al. (2025). Retrieval Augmented Generation and Understanding in Vision: A Survey and New Outlook. *arXiv preprint*. arXiv:2503