Multimodal LLM Evaluation with Advanced Multimodal RAG Pipeline using Video Input

Multimodale LLM-Auswertung mit erweiterter multimodaler RAG Pipeline unter Verwendung von Videoeingang

Masterarbeit
im Rahmen des Studiengangs Robotik und Autonome Systeme der Universität zu Lübeck

Vorgelegt von
Aly Elzoghol

Ausgegeben und betreut von
Prof. Dr. Philipp Rostalski

Mit Unterstützung von
Dr. Niko Seubert

Diese Arbeit ist im Rahmen von Arbeiten bei der Firma Robert Bosch GmbH entstanden.

Lübeck, den 15. May 2025

Acknowledgment

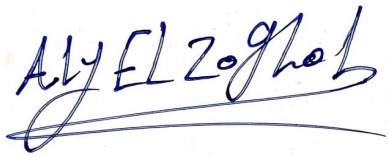
This research has been accomplished at PS-DC/PJ in BOSCH GmbH. I would like to express my gratitude, especially to Dr. Niko Seubert, and Mr. Ruben Kapp for their constant support and efforts. The chance I have been given was more than I hoped for and I have learned a lot through this journey. Also, I want to thank my university supervisors, Prof. Dr. Philipp Rostalski, and Georg Wolf for their valuable guidance and support, which enabled me to successfully complete my thesis.

Lübeck, den 15. May 2025

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne die Benutzung anderer als der angegebenen Hilfsmittel selbst ständig verfasst habe; die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe des Literaturzitats gekennzeichnet.

Lübeck den 15. May 2025
Aly Elzoghol



Zusammenfassung

Multimodale Modelle sind KI-Modelle, die in der Lage sind, mehrere Modalitäten wie Text, Bilder, Video und Audio gleichzeitig zu verarbeiten, um sinnvolle Ausgaben zu generieren. Diese Modelle basieren auf multimodalen Frameworks, die Daten aus verschiedenen Quellen integrieren und so ein kontextbezogeneres und umfassenderes Verständnis der Eingaben ermöglichen.

In dieser Arbeit wird die Herausforderung untersucht, Videoeingaben effizient mit verschiedenen multimodalen großen Sprachmodellen (MM-LLMs) zu verarbeiten. Zu diesem Zweck wurde eine Video Retrieval-Augmented Generation (Video-RAG)-Pipeline entwickelt, die es MM-LLMs ermöglicht, eine große Anzahl vorverarbeiteter und vorgelagerter Videos innerhalb einer domänenspezifischen Vektordatenbank zu analysieren. Zwei multimodale RAG Pipelines (MM-RAG) wurden bewertet, die unterschiedliche Arten visueller Eingaben nutzen: (i) die Verarbeitung roher Einzelbilder mit einem Vision Transformer (ViT-g-14) und (ii) die Verarbeitung semantischer Beschreibungen einzelner Frames mithilfe eines Text-Embedding-Modell (text-embedding-3-large). Beide Ansätze beinhalten Kompromisse zwischen Speicherbedarf und Rechenaufwand, was für die Optimierung realer Anwendungen entscheidend ist.

Beide MM-RAG Pipelines wurden mit verschiedenen MM-LLMs getestet, wobei letztlich das GPT-4o-Modell bessere Leistungen als Modelle aus den LLaMA-, Qwen2- und Llava-Familien zeigte. Zur Validierung der Pipeline-Leistung wurden zwei reale Anwendungsfälle genutzt, um die Hyperparameter zu optimieren und eine abschließende Evaluierung durchzuführen. Ein Anwendungsfall bestand in der Analyse einer Videoaufnahme eines chirurgischen Trainingsverfahrens. Ein strukturierter Satz von Frage-Antwort-Paaren (QA) wurde vorbereitet, um die Fähigkeit des Modells zu bewerten, Fragen aus verschiedenen Kategorien – darunter Handlungen, Objekte, Gründe und Anomalien – zu beantworten. Ein weiterer Anwendungsfall konzentrierte sich auf die Fehlerbehebung bei Hardwareproblemen anhand einer Reihe von Anleitungsvideos, die von erfahrenen Technikern aufgenommen wurden. Eine gleichmäßige Frame-Sampling-Methode wurde eingesetzt, bei der ein geeigneter Frames-per-Second (FPS) Wert feinjustiert wurde, um eine ausreichende Informationsgewinnung für jeden Anwendungsfall sicherzustellen. Die Ergebnisse zeigten die Fähigkeit der Pipeline, komplexe und kontextuell herausfordernde Anfragen zu

verarbeiten.

Es wurde eine Video-RAG Pipeline entwickelt, bei der semantische Beschreibungen aufeinanderfolgender Frames aus Videoinhalten extrahiert und in einer Vektordatenbank gespeichert wurden. Dies ermöglichte ein effektives Frage-Antwort-System auf Basis sequenzieller Frame-Informationen und verbesserte die Fähigkeit des Modells, relevante Details aus multimodalen Videodaten abzurufen.

Abstract

Multi-modal models are AI models capable of simultaneously processing multiple modalities, such as text, images, video, and audio, to generate meaningful outputs. These models rely on multi-modal frameworks that integrate data collected from various sources, providing a more context-specific and comprehensive understanding of the input.

In this research, the challenge is to efficiently process video inputs using different multi-modal large language models (MM-LLMs). To address this, a Video Retrieval-Augmented Generation (Video-RAG) pipeline was developed, enabling MM-LLMs to process a large number of pre-processed and pre-stored videos within a domain-specific vector database. Two multi-modal RAG (MM-RAG) pipelines were evaluated, utilizing different types of visual input: (i) raw frame processing with a vision transformer (ViT-g-14) and (ii) frame-wise semantic description processing using a text embedding model (text-embedding-3-large). Each approach presents a trade-off between storage requirements and computational efficiency, which is crucial for optimizing real-world implementations.

Both MM-RAG pipelines were tested using various MM-LLMs, ultimately concluding that the GPT-4o model outperformes other models from the LLaMA, Qwen2, and Llava families. To validate the performance of the pipeline, two real-world use cases were employed for the pipeline's hyperparameter fine-tuning and final evaluation. One such use case involved analyzing a video recording of a surgical procedure training. A structured set of question-answer (QA) pairs was prepared to assess the model's ability to answer questions across different categories, including actions, objects, reasons, and anomalies. Another use case focused on hardware troubleshooting, utilizing a series of instructional videos recorded by expert engineers to guide technicians. A uniform frame sampling approach was applied by fine-tuning a suitable Frames-Per-Second (FPS) value to ensure sufficient information retrieval for each use case. The results demonstrated the pipeline's capability to handle complex and contextually challenging queries.

A Video-RAG pipeline was developed where semantic descriptions of sequential frames were extracted from video content and stored in a vector database. This enabled effective question-answering based on sequential frame information, improving the model's ability to retrieve relevant details from multimodal video data.

Contents

1. Introduction **1**
1.1. Motivation 1
1.2. Related Work 2
1.3. Thesis outline 4

2. Research Theory **5**
2.1. GenAI and LLMs 5
2.2. Embedding Models 7
2.2.1. Text Embedding Model 8
2.2.2. Vision Transformer 10
2.3. Multimodal LLMs 10
2.3.1. GPT-4o 11
2.3.2. LLaMA3.2-Vision-90B 12
2.3.3. LLaVA-OneVision-7B 13
2.3.4. LLaVA-NeXT-7B 13
2.3.5. Qwen2-VL-7B 14
2.4. Text-RAG 15
2.4.1. Baseline-RAG 16
2.4.2. Advanced-RAG 16
2.5. Multimodal-RAG 18
2.5.1. Image-RAG 18
2.5.2. Video-RAG 19
2.6. RAGAS Evaluation Metrics 21
2.6.1. Text-RAG Evaluation Metrics 21
2.6.2. Multimodal-RAG Evaluation Metrics 23

3. Models and Methods **25**
3.1. Image-RAG Pipeline 25
3.1.1. Raw-Image Input 25
3.1.2. Image-Description Input 25
3.2. Video-RAG Pipeline 28
3.2.1. Key-Frames Extraction 28
3.2.2. Key-Frame Processing 30
3.2.3. Redundancy Removal 32
3.2.4. Temporal Information 38

4. Evaluation Setup **41**
4.1. Academic Multimodal Test Set 41
4.1.1. Raw-Image Pipeline 41
4.1.2. Image-Description Pipeline 42
4.2. Real Use Case Video Test Set 42
4.2.1. Surgical Procedure Training 43
4.2.2. Test-Bench Hardware Troubleshooting 44

5. Evaluation Results **47**
5.1. Academic Multimodal Test Set 47
5.1.1. Raw-Image Pipeline 47
5.1.2. Image-Description Pipeline 48
5.2. Real Use Case Video Test Set 49
5.2.1. Surgical Procedure Training 49
5.2.2. Test-Bench Hardware Troubleshooting 54
5.3. Cost-Efficiency Analysis of Multimodal LLMs 55

6. Discussion **57**
6.1. Key-Frame Processing 57
6.2. MM-LLMs Performance 58
6.3. Semantic Description 59
6.4. Key-Frames Extraction 60
6.5. Temporal Information 60
6.6. Real Use Case Videos 61

7. Conclusion **63**

8. Future Work **65**

A. Appendix **67**

A.1. Software Dependencies 67

A.2. Hardware Resources Cost Calculations 68

A.3. Audio Transcript Result 69

List of Figures **71**

List of Tables **77**

Bibliography **79**

1. Introduction

1.1. Motivation

The rapid advancement of Large Language Models (LLMs) has spurred substantial interest in extending their capabilities to video understanding. This has led to the emergence of Large Video-Language Models (LVLMs), which demonstrate strong performance in low-interaction video comprehension. However, reasoning over videos with high interaction density and extended temporal dependencies remains a persistent and unsolved challenge.

A key difficulty lies in effectively modeling temporal information across sequential frames in videos. To address this, several research approaches have been proposed. For example, LLaVA-Video expands the token capacity of existing LLMs and adapts their architectures to better interpret video data. Despite its effectiveness, this approach requires extensive pretraining on large-scale datasets, and there remains a discrepancy between the conditions under which models are fine-tuned and those encountered during real-world deployment.

Recent research has explored the integration of Retrieval-Augmented Generation (RAG) into video-language modeling. RAG improves generative capabilities by retrieving relevant external information, thereby enriching the model's responses through access to contextual and temporal information. In the context of video, early approaches have adapted this method by treating long video content as plain text, applying RAG to retrieve related semantic descriptions before passing the information to proprietary models.

In this work, Video-RAG is applied to two real-world scenarios in the medical and industrial domains to evaluate the capability of Multimodal LLMs (MM-LLMs) and the Video-RAG framework in leveraging contextual and temporal information from video inputs.

In summary, Video-RAG [35] addresses key limitations in existing video-language

modeling approaches by combining the strengths of retrieval-based augmentation with lightweight auxiliary data integration. It delivers enhanced multimodal reasoning, preserves essential visual semantics, and achieves performance comparable to state-of-the-art proprietary systems all while relying solely on open-source models like LLaVA-Video [32] or Gemini 1.5 [55]. This represents a significant step toward scalable, efficient, and accessible multimodal video understanding.

1.2. Related Work

Multimodal LLMs [69] extend traditional LLM [72] [8] architectures by enabling models to process and generate information across multiple modalities, including text, images, videos, and audio. This multimodal functionality is achieved through the integration of specialized unimodal components into a unified framework. Typically, such models consist of modality-specific encoders, a pre-trained language model core, and modality-aligned output generators. These components are connected via projectors mapping functions that facilitate the transformation of input and output representations across different data types.

Despite their generative power, LLMs often face challenges when responding to domain-specific or long-tail queries, especially when required to recall factual knowledge [24]. To address this, Retrieval-Augmented Generation (RAG) [18] [13] has emerged as a powerful solution that augments generative models with access to external knowledge sources. RAG combines a vector-based retrieval mechanism with generative modeling to fetch relevant contextual information from large-scale knowledge bases, thereby enhancing output accuracy and informativeness.

Extending the principles of RAG, Multimodal Retrieval-Augmented Generation (MM-RAG) [46] [1] incorporates retrieval and generation across heterogeneous data types, such as images and audio, in addition to text. MM-RAG frameworks maintain an external memory or knowledge base containing multimodal content, enabling models to perform retrieval using a combination of modalities.

Video Retrieval-Augmented Generation (Video-RAG) [35] has emerged as a transformative approach in processing and understanding long-form video content by integrating retrieval mechanisms with generative models. Beyond general applications, Video-RAG has demonstrated significant potential in specialized domains [47] such as surgical training, industrial maintenance, and educational technology.

In the medical field, particularly in surgical training, Video-RAG facilitates automated assessment of surgical skills by analyzing procedural videos [67]. For instance, frameworks like Video Semantic Aggregation (ViSA) [30] segment surgical videos into semantically meaningful parts, enabling the evaluation of technical proficiency across different surgical phases. This approach not only aids in objective skill assessment but also provides explanatory visualizations to understand the decision-making process of the model . Additionally, systems have been developed to segment surgical procedures into predefined phases, annotating each segment with relevant descriptors, thereby assisting in training machine learning models to detect similar patterns in other surgical videos.

In industrial settings, Video-RAG enhances equipment maintenance and knowledge management by providing contextual assistance based on historical data [56] [38]. For example, AI-driven assistants utilize Video-RAG to retrieve and present relevant information from maintenance logs, manuals, and past repair videos, offering step-by-step troubleshooting guidance. This not only reduces downtime but also preserves institutional knowledge, making it accessible to newer technicians. Furthermore, interactive industrial knowledge management systems have been developed, employing Video-RAG to facilitate complex query responses and improve decision-making processes within organizations [9].

In the realm of education, Video-RAG has been instrumental in developing intelligent tutoring systems that adapt to learners' needs [51]. By retrieving pertinent video segments and generating explanatory content, these systems provide personalized learning experiences. For instance, in surgical education, platforms utilize Video-RAG to offer trainees access to a curated library of procedural videos, enabling them to study specific techniques and receive feedback on their performance [60]. Such applications have been shown to enhance learning outcomes by providing contextually relevant information tailored to individual learning paths [19].

Evaluating the performance of RAG-based systems requires an assessment of both retrieval accuracy and generation quality. Common evaluation criteria include faithfulness, relevance, contextual alignment, and informativeness of the model's outputs. Benchmarking typically involves the use of domain-specific datasets that test the end-to-end system across both retrieval and response generation components. One increasingly adopted method involves self-evaluation by LLMs, where models assess the quality of their own outputs. While this provides a scalable and often effective

proxy for human judgment, manual human evaluation remains the gold standard particularly in high-stakes domains where factual precision is critical. In the context of multimodal evaluation, vision-language models (VLMs) [7] [31] have been shown to correlate well with human assessments, thereby validating their effectiveness in scenarios requiring cross-modal reasoning.

1.3. Thesis outline

The choice of an optimal MM-LLM is a crucial factor in determining the overall performance of the Video-RAG pipeline. Various models from the LLaMA, Qwen2, and LLaVA families were evaluated in Chapter 3. To validate the effectiveness of the pipeline, real-world use cases were utilized, including the analysis of surgical procedure training videos to extract contextual and visual information, as well as detect anomalies within the video inputs. Structured Question-Answer (QA) evaluations, as detailed in Chapter 4, assessed the model's ability to generate precise and contextually relevant responses across various categories, including actions, objects, reasons, and anomalies.

This research is driven by the increasing demand for efficient, scalable, and domain-specific video retrieval and reasoning systems. By developing and evaluating a Video-RAG pipeline that incorporates multimodal processing techniques, this study seeks to advance the field of AI-driven video understanding. The proposed methodology not only aimed to improve the efficiency of multimodal large language models (MM-LLMs) in handling large-scale video data but also to offer a structured approach to enhance retrieval quality and interpretability in critical applications such as medical procedure education and automated processes. The evaluation results and experimental setup for various componements of the pipeline are presented in Chapter 5.

2. Research Theory

2.1. GenAI and LLMs

Generative Artificial Intelligence (GenAI) holds significant importance in the domain of intelligent systems. This category of AI is closely associated with Large Language Models (LLMs), exhibiting the capability to generate human-like language responses. GenAI models are capable of producing original content, including images, music, and text. By leveraging extensive datasets, these models employ advanced machine learning algorithms to identify patterns and generate coherent outputs [37]. Common techniques utilized in GenAI include recurrent neural networks (RNNs) [50] [48] and generative adversarial networks (GANs) [20]. Moreover, the transformer architecture [58] represented by the "T" in ChatGPT [33] is a foundational component of this technology.

For instance, an image generation model may be trained on millions of images and illustrations to learn the visual patterns and characteristics that define different categories of visual content. Similarly, models trained for music and text generation utilize extensive datasets of musical compositions or textual corpora, respectively. Notable examples of GenAI models include DALLE [45] by OpenAI, which is trained on a diverse set of images and is capable of generating unique and detailed visuals based on textual prompts.

GenAI presents numerous advantages, such as fostering creativity through the generation of diverse, context-aware content, facilitating natural human-computer interaction, and enhancing problem-solving efficiency via intelligent insights. Its versatility allows it to be effectively applied across various industries. Furthermore, GenAI models can improve iteratively as they are exposed to more data and user interactions.

LLMs [8] represent a prominent subclass of GenAI. These models consist of billions of parameters trained on extensive textual data for tasks involving natural language

understanding and generation. For example, LLaMA3.2-Vision-90B [39] comprises 90 billion parameters. LLMs are capable of achieving deep contextual comprehension and maintaining coherence across interactions due to the incorporation of memory mechanisms within their architecture. These mechanisms allow for the storage and retrieval of relevant information, enabling the generation of contextually accurate and coherent responses.

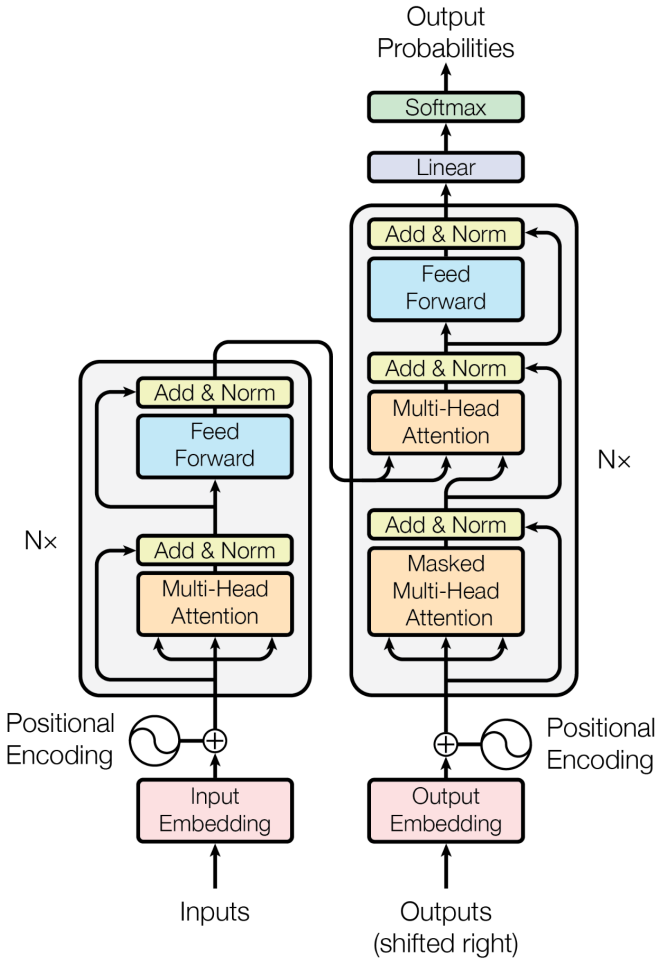


Figure 2.1.: The Transformer model serves as the foundational architecture for all large language models. It employs stacked self-attention mechanisms and point-wise fully connected layers in both the encoder and decoder, as illustrated in the left and right halves of the model architecture. Graph is referenced from [58].

The Transformer model architecture shown in Figure 2.1 follows an encoder-decoder structure. The encoder maps an input sequence of symbol (where each symbol may represent a word or sub-word) representations $\mathbf{x} = (x_1, \ldots, x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, \ldots, z_n)$. Given $\mathbf{z}$, the decoder generates an output sequence $\mathbf{y} = (y_1, \ldots, y_m)$ of symbols, one element at a time. At each step, the model operates in an auto-regressive manner [21], consuming the previously generated symbols as additional input for predicting the next token. The Transformer implements this architecture using stacked self-attention mechanisms and point-wise, fully connected layers in both the encoder and decoder components, shown in the left and right halves of Figure 2.1, respectively.

Representative examples of LLMs include GPT-3 and GPT-4 (Generative Pre-trained Transformers) [2] by OpenAI, PaLM2 (Pre-trained Auto-Regressive Language Model 2) [6] by Google AI, and LLaMA1.3 (Large Language Model Meta AI 1.3) [57] by Meta. These models are employed in a wide range of natural language processing tasks, such as multimodal content generation, text completion, summarization, translation, question answering, and code generation.

2.2. Embedding Models

Word embeddings are dense vector representations of individual words in a text, capturing both the contextual meaning and the relationships with surrounding words. The dimensionality of these real-valued vectors can be predefined, allowing semantic relationships between words to be represented more effectively than in traditional Bag-of-Words [16] models. In essence, words with similar meanings or those that frequently occur in similar contexts tend to have closely aligned vector representations, reflecting their semantic proximity in the embedding space as illustrated in Figure 2.2.

Semantic meaning refers to the interpretation of words, phrases, or sentences in terms of their conceptual or contextual significance, as opposed to their literal or syntactic form. This approach emphasizes capturing the underlying intent or message conveyed by the text rather than its surface-level structure. For instance, the sentences “The cat is on the mat” and “The mat has a cat on it” convey equivalent semantic content despite differing in word order. A well-trained embedding model should produce similar vectors for these inputs, thereby reflecting their conceptual

equivalence.

The dimensionality of an embedding model for example, 3072 is determined by model designers based on the underlying architecture and the specific training objectives. This dimensionality reflects a trade-off between expressive capacity and computational efficiency. Higher-dimensional embeddings are capable of capturing more subtle relationships and complex patterns within the data, thereby enabling a richer semantic representation of text. However, increasing dimensionality also incurs greater computational and memory costs, and beyond a certain threshold, it may yield diminishing returns in terms of performance [66]. Therefore, optimizing the dimensionality is essential to achieve a balance between representational richness and resource efficiency. The following subsections provide a detailed explanation of the differences between vision-based and text-based embedding models.

2.2.1. Text Embedding Model

The text-embedding-3-large [61] model employs a substantially higher embedding dimensionality of 3072, providing notable advantages in semantic representation. The expanded vector space enables more fine-grained encoding of linguistic features and relationships within textual data, making it particularly effective when working with large and heterogeneous datasets. Higher-dimensional embeddings facilitate improved differentiation between closely related yet semantically distinct inputs—an essential capability in applications requiring nuanced language understanding, such as question-answering systems, semantic search, and natural language inference. For example, in a question-answering system, the queries "What is the capital of France?" and "What is the population of France?" share many words but require very different types of answers. Low-dimensional embeddings may struggle to capture this distinction due to limited representational capacity. In contrast, higher-dimensional spaces allow the model to encode finer semantic nuances, grouping geographic questions separately from statistical ones. This enables more accurate query interpretation and routing to relevant knowledge sources. Thus, the selection of embedding dimensionality represents a critical architectural decision, requiring a balance between representational expressiveness and computational efficiency [65]. While higher dimensional embeddings generally improve performance, they also impose greater computational and memory demands.

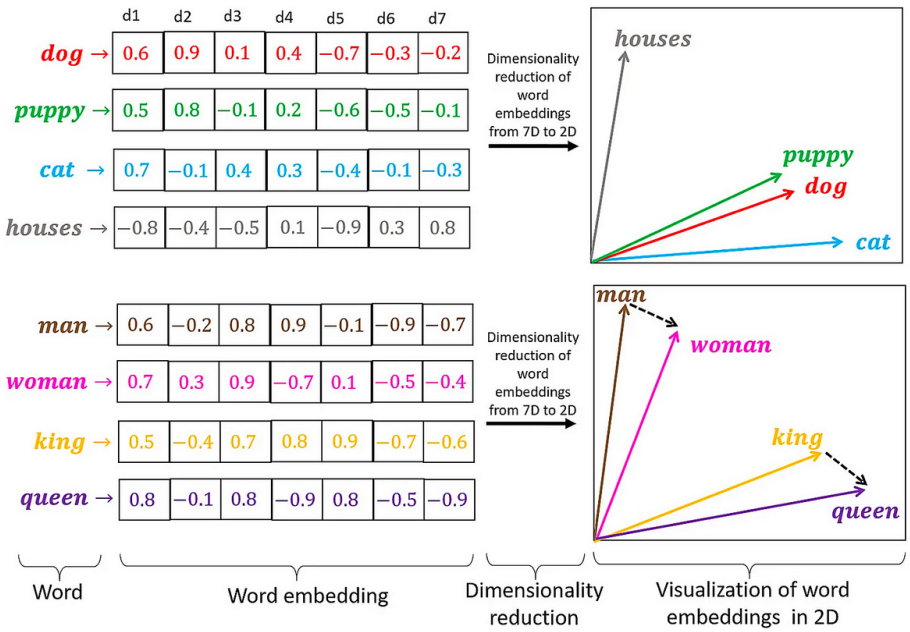


Figure 2.2.: Embedding Vector Space: A common example illustrating the properties of semantic embeddings is that the vector difference between "man" and "woman" is approximately equivalent to that between "king" and "queen." Similarly, semantically related word pairs such as "dog" and "puppy" tend to have closely aligned vector representations, reflecting their conceptual similarity.

2.2.2. Vision Transformer

The Vision Transformer (ViT) architecture [12] [3] is a deep learning model specifically adapted for processing image-based inputs. The ViT-g-14 variant employs an embedding dimensionality of 1024, which influences its capacity for internal representation. A reduced embedding dimensionality limits the model's ability to encode fine-grained details, potentially leading to a loss of representational granularity. However, this reduction also improves computational efficiency and can still yield high-quality representations for a wide range of vision tasks. Although lower-dimensional embeddings may be somewhat less expressive or discriminative particularly when applied to longer or more complex inputs. the ViT architecture is engineered to retain strong performance in visual domains, effectively balancing representational depth with efficiency. Rather than processing raw pixels directly or relying on convolutional operations, the ViT architecture, shown in Figure 2.3, divides an input image (e.g., 224x224 pixels) into fixed-size patches (e.g., 16x16). Each patch is then flattened and projected linearly into an embedding vector. This sequence of patch embeddings is treated analogously to word tokens in natural language processing, significantly reducing the input sequence length compared to pixel-level processing. Since Transformers are inherently permutation-invariant and do not capture order on their own, ViTs incorporate either learned or fixed positional embeddings to the patch embeddings. This addition enables the model to preserve spatial relationships between patches, an essential requirement for vision-related tasks. Following the transformation of an input image into a sequence of patch embeddings augmented with positional encodings, the resulting sequence is processed through the standard components of the Transformer architecture. These include multi-head self-attention layers, feedforward multilayer perceptrons (MLPs), layer normalization, and residual connections. Collectively, these mechanisms endow the model with a high degree of representational power and expressive capacity, enabling effective learning and inference in vision-related tasks.

2.3. Multimodal LLMs

Recent advances in multimodal (MM) pre-training [68] [10] have significantly improved performance across a wide range of downstream tasks. However, as mod-

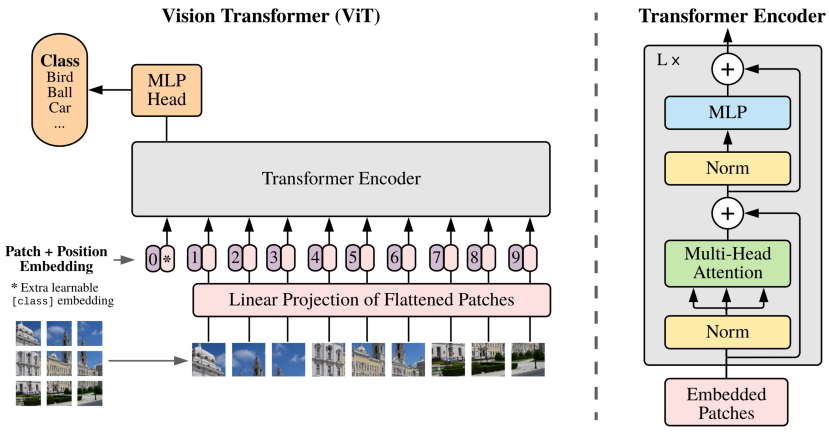


Figure 2.3.: Vision Transformer Archeticture : split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the ViT is taken from [12].

els and datasets continue to increase in scale and complexity, traditional MM approaches demand substantial computational resources, particularly when trained from scratch. Given that MM research inherently involves the integration of multiple data modalities, a more efficient strategy has emerged: leveraging pre-trained unimodal foundation models, especially Large Language Models (LLMs). This paradigm not only reduces computational overhead but also enhances the overall efficiency of MM pre-training, giving rise to a novel direction in AI research—Multimodal Large Language Models (MM-LLMs). In this section, we examine five representative MM-LLM architectures and explore their respective design methodologies.

2.3.1. GPT-4o

GPT-4o [2] is a state-of-the-art multimodal model developed by OpenAI, designed to process and generate text, images, and audio within a unified architecture. Unlike its predecessor GPT-4V [64], which utilizes separate encoders for different modalities, GPT-4o is trained end-to-end on text, vision, and audio inputs using a single transformer-based model. All input types are converted into a shared token space, allowing seamless integration of modalities during inference. Visual data, such as im-

ages, are likely processed using a patch-based embedding approach similar to Vision Transformers (ViTs), while audio inputs are tokenized through a learned frontend inspired by Whisper [5] [42]. In multimodal models that process both text and images, image tokens refer to numerical representations or discrete units of data that encode segments of an image, analogous to how words are tokenized in text-based models. These tokens are then fed into the same decoder-only transformer used for text, enabling joint reasoning across modalities. This architecture significantly reduces latency and computational overhead compared to modular systems, while improving performance in tasks like visual question answering (VQA), speech-based interaction, and image-to-text generation. GPT-4o demonstrates real-time audio processing capabilities, achieving response latencies as low as 232 milliseconds using OpenAI instances. Its design reflects a shift toward more fluid and integrated multimodal intelligence, marking a key advancement in unified AI systems.

2.3.2. LLaMA3.2-Vision-90B

The LLaMA3.2-Vision series [39] comprises instruction-tuned multimodal models with 11B and 90B parameter variants, optimized for tasks involving image reasoning, visual recognition, image captioning, and visual question answering. These models accept both image and text inputs while generating textual outputs, and they surpass many open-source and proprietary counterparts on established benchmarks. For text-only tasks, the model officially supports many languages—English, German, French, Italian, and Spanish while also being trained on a wider multilingual corpus. However, for multimodal tasks, English remains the primary language supported.

The LLaMA-Vision-90B model is a large-scale multimodal architecture that integrates a 90-billion-parameter LLaMA language model with a vision encoder, enabling joint processing of visual and textual information. A notable innovation in this model is the implementation of a dual-token strategy, where each visual input is represented by two distinct tokens: a context token summarizing the overall image and a content token capturing fine-grained visual details. This approach aims to reduce computational overhead, particularly in processing long-form visual content, by minimizing the number of visual tokens without sacrificing richness. While not yet validated on video benchmarks, such a model could provide improved scalability

and efficiency across a range of image understanding tasks.

2.3.3. LLaVA-OneVision-7B

LLaVA-OneVision is a family of open large multimodal models (LMMs) introduced through rigorous experimentation with data curation, architectural innovations, and visual representation strategies [27]. Empirical results suggest that LLaVA-OneVision significantly advances performance on single-image and multi-image reasoning tasks. While not designed for video input, the model shows early potential for generalizing to video-derived frame analysis, leveraging strong image-based pretraining. This highlights its robustness in handling long-form visual inputs, though further work is needed for true video comprehension.

The LLaVA-OneVision-7B model integrates a vision encoder with a 7-billion-parameter language model, enabling unified vision-language understanding through a single-stream architecture. It employs a projection layer to align visual embeddings with the language model's input space, facilitating seamless multimodal interaction. This architecture offers notable advantages, including improved generalization across diverse vision-language tasks and reduced system complexity compared to dual-stream approaches.

2.3.4. LLaVA-NeXT-7B

LLaVA-NeXT-Interleave extends the LLaVA-NeXT framework [28] by introducing mechanisms for advanced multi-image understanding. Its architecture includes a vision encoder, an intermediate projector, and a large language model. Through enhanced training techniques, the model generalizes single-image expertise to more complex multi-image tasks. It builds upon LLaVA-NeXT-Image, which is pretrained on image-caption pairs and fine-tuned for single-image tasks. The interleaved multi-image instruction tuning is conducted using the M4-Instruct dataset, enhancing the model's ability to handle diverse visual inputs. Evaluation results show that LLaVA-NeXT-Interleave outperforms previous open-source models on both in-domain and out-of-domain benchmarks, demonstrating robust multimodal reasoning capabilities. In several evaluations, such as Mantis-Eval and BLINK, it achieves performance comparable to GPT-4V.

The LLaVA-NeXT-7B model is an advanced vision-language architecture specifically designed to tackle abstract visual reasoning (AVR) tasks. Building upon the LLaVA framework, it incorporates a 7-billion-parameter language model integrated with a vision encoder. A key innovation of LLaVA-NeXT-7B lies in its data synthesis and post-training process, which enables the model to learn complex reasoning patterns step by step.

The model is trained using synthetically generated instruction-following data that simulates detailed reasoning. This approach involves prompting a more capable language model, such as GPT-4V, with images to generate structured outputs including step-by-step rationales, explanations, and dialogues grounded in visual content. The resulting synthetic data encompasses multi-turn question answering, multi-step reasoning, and explanatory chains, which are used to fine-tune the model.

The post-training process includes multiple stages: (i) Instruction tuning, where diverse vision-language instruction pairs are employed; (ii) Conversational fine-tuning, emphasizing dialogue-style interaction over single-turn outputs; and (iii) Rationale-focused training, encouraging the model to produce outputs that justify answers, enhancing interpretability and depth of reasoning. This approach allows the model to surpass both open-sourced (e.g., Qwen-2-VL-72B) and closed-sourced (e.g., GPT-4o) powerful VLMs on representative AVR benchmarks. Notably, LLaVA-NeXT-7B achieves this enhanced reasoning capability without compromising its performance on general multimodal comprehension tasks.

2.3.5. Qwen2-VL-7B

The Qwen2-VL [62] series consists of models of 3 sizes, which are Qwen2-VL-2B, Qwen2-VL-7B and Qwen2-VL-72B. Notably, Qwen2-VL employs a 675M parameter ViT across various-sized LLMs, ensuring that the computational load of the ViT remains constant regardless of the scale of the LLM. Figure 2.4 illustrates the comprehensive structure of Qwen2-VL which integrates vision encoders and language models. A Vision Transformer (ViT) [12] with approximately 675 million parameters, adept at handling both image and video inputs.

The Qwen2-VL-7B model employs a unified multimodal architecture that integrates same vision encoder with a 7-billion-parameter transformer-based language model, enhanced by innovations such as Multimodal Rotary Position Embedding (M-

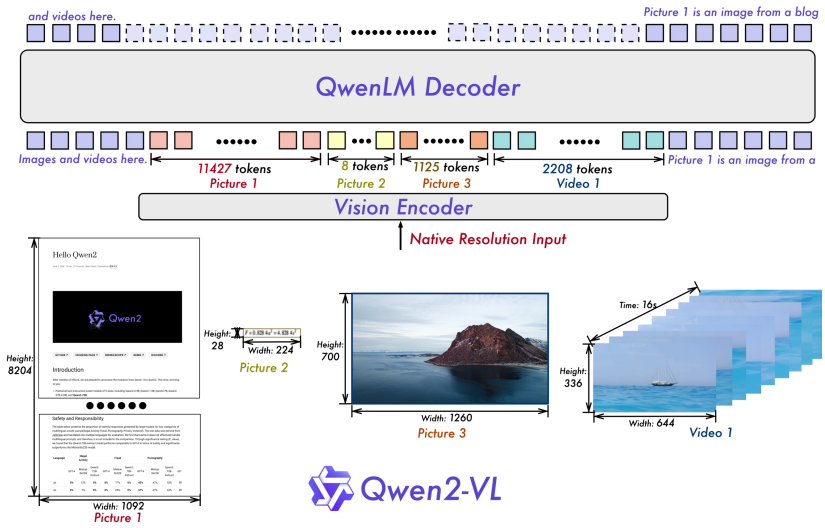


Figure 2.4.: Qwen-VL model Archeticture: Qwen2-VL is capable of accurately identifying and comprehending the content within images, regardless of their clarity, resolution, or extreme aspect ratios. Graph is taken from [62]

RoPE) [52], Naive Dynamic Resolution (NDR) [11], and Unified Image and Video Understanding employing a mixed training regimen incorporating both image and video data, ensuring proficiency in image understanding and video comprehension. These components allow the model to flexibly handle images and videos of varying resolutions while preserving positional and spatial relationships across modalities. This design enables efficient visual reasoning and strong generalization across diverse multimodal tasks, with competitive performance even against significantly larger models.

2.4. Text-RAG

Retrieval-Augmented Generation (RAG) [18] enhances large language models (LLMs) by integrating external knowledge sources, thereby improving the factual accuracy and contextual relevance of generated outputs. This is particularly beneficial in domain-specific or dynamically evolving contexts where up-to-date and specialized information is critical.

A fundamental component of RAG systems is chunking, which structures large

text corpora into semantically coherent segments. This facilitates efficient similarity-based retrieval and improves the alignment between user queries and retrieved content. Despite their effectiveness, conventional RAG systems exhibit limitations. They often rely on flat data structures, hindering their capacity to model complex inter-entity relationships. Furthermore, limited contextual reasoning can result in fragmented responses when queries span multiple interrelated concepts. For instance, in response to a multifaceted query such as: "How does the rise of electric vehicles impact urban air quality and public transportation infrastructure?", traditional RAG systems may retrieve relevant information on each component independently but fail to synthesize a coherent, interconnected explanation.

In the text-only RAG configuration, textual data is extracted from PDF documents and embedded using OpenAI's text-embedding-3-large model. These embeddings were stored in a vector database to enable similarity-based retrieval. Retrieved passages were concatenated with the user query and passed to a multimodal LLM for response generation. This setup allowed for an evaluation of text-based retrieval efficacy in addressing domain-specific inquiries. The following subsections provide a detailed explanation of various RAG technologies.

2.4.1. Baseline-RAG

Naive RAG [18] functions as a fundamental baseline model within Retrieval-Augmented Generation (RAG) systems, designed to address existing limitations in information retrieval and text generation. This approach, shown in Figure 2.5, involves partitioning raw text into discrete, manageable segments, which are subsequently stored in a vector database through the application of text embeddings. This method serves as a foundational benchmark against which the performance and effectiveness of more sophisticated RAG models can be evaluated.

2.4.2. Advanced-RAG

Advanced Retrieval-Augmented Generation (Advanced-RAG) [13] refers to a set of enhanced techniques that extend the standard RAG framework to address limitations related to retrieval precision, context relevance, and generative fidelity. While traditional RAG pipelines retrieve and integrate external documents based on simple chunking and basic retrieval mechanisms, advanced variants adopt more so-

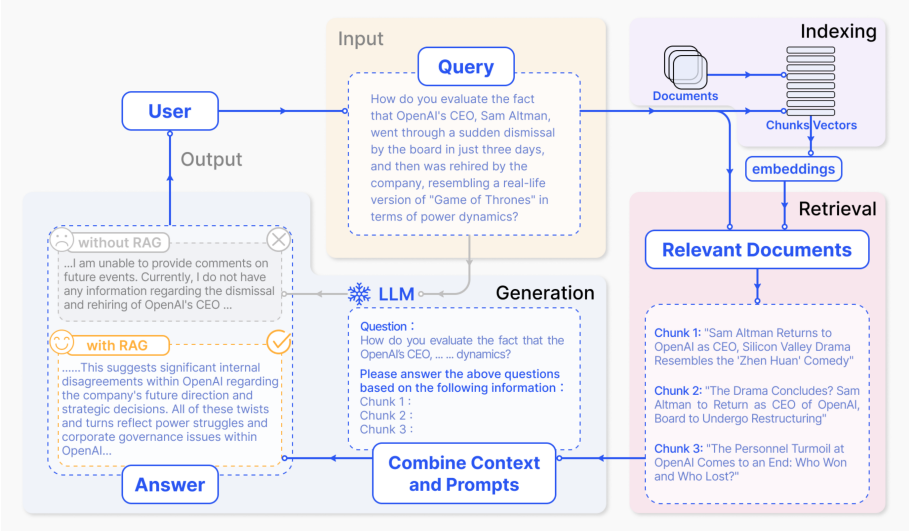


Figure 2.5.: Text-RAG Baseline Architecture [18]

phisticated strategies to improve the alignment between retrieved content and the generative task.

One such enhancement is semantic chunking [41], which segments source documents not by fixed token lengths but based on semantic coherence (e.g., topic shifts, discourse boundaries, or scene transitions). This approach preserves contextual integrity in retrieved passages, leading to more meaningful grounding during generation. In parallel, advanced prompting strategies [4] [59] leverage dynamic query formulation, instruction tuning, or multi-turn reasoning prompts to better guide the retriever and generator, especially in complex or multi-hop tasks.

Another innovation is ensemble RAG [29], which combines multiple retrieval and generation pathways—such as using different retrievers (e.g., dense vs. sparse), aggregating outputs from multiple generators, or voting mechanisms—to enhance robustness and reduce hallucination. Additionally, integration with re-ranking models or retrieval fusion techniques (e.g., HyDE, FiD-RAG) [17] [26] [22] further refines the quality of retrieved evidence before generation.

Collectively, these techniques enable Advanced-RAG systems to better handle noisy or large-scale corpora, adapt to diverse domains, and produce more accurate, grounded, and coherent outputs [13]. As such, Advanced RAG represents a critical step toward more reliable and domain-aware retrieval-augmented generation frame-

works.

Prompt Optimization

Prompt optimization [40] [53] in RAG systems focuses on designing prompts that effectively guide the language model to leverage retrieved information. This includes structuring the prompt to balance context length, retrieved content, and task instructions. Techniques such as prompt tuning, prefix tuning, and instruction prompting help improve performance by aligning the model's behavior with the retrieval results. Recent work explores query rewriting and retrieval-aware prompting, which adapt the input or prompt format based on the retrieved documents' structure and relevance. Optimizing prompts is especially important when dealing with multi-document or noisy retrieval outputs, where poorly framed prompts can lead to hallucinations or irrelevant generations.

2.5. Multimodal-RAG

Multimodal Retrieval-Augmented Generation (Multimodal-RAG) [36] [63] [46] extends the RAG framework by incorporating multiple data modalities such as text, images, video, and audio into both the retrieval and generation stages. This enables the model to generate richer, context-aware responses grounded in diverse modalities. It is particularly effective in tasks requiring visual or auditory grounding, such as image-based question answering or video summarization. By unifying retrieval and reasoning across modalities, Multimodal-RAG enhances both factual accuracy and interpretability in complex, real-world scenarios [71] [25] [54].

2.5.1. Image-RAG

Image Retrieval-Augmented Generation (Image-RAG) is a hybrid approach that integrates visual understanding with retrieval-based techniques to enhance the performance of multimodal tasks such as image captioning, visual question answering, and image-grounded dialogue. Rather than relying solely on the pre-trained knowledge of a vision-language model like CLIP [43], Image-RAG introduces an external retrieval mechanism that enables the model to access and incorporate supplementary information relevant to the input image.

The process typically involves three key components: (i) an image encoder that extracts semantic features from the input image, (ii) a retrieval module that uses these features to query an external knowledge base (e.g., image-text datasets, documents, or metadata), and (iii) a generative module that synthesizes a response by conditioning on both the visual input and the retrieved content. This architecture allows for contextually rich and factually grounded outputs, especially in cases where the visual signal alone is ambiguous or insufficient.

By leveraging external information, Image-RAG systems can dynamically adapt to new domains and rare scenarios without requiring extensive re-training. This approach builds upon the principles of text-based Retrieval-Augmented Generation (RAG) and extends them to the visual modality, thereby enabling more robust and informed multimodal reasoning.

2.5.2. Video-RAG

Video Retrieval-Augmented Generation (Video-RAG) [47] [70] [35] extends the RAG framework to the video domain, combining temporal visual understanding with external knowledge retrieval to enhance long video comprehension tasks such as summarization, question answering, and event understanding. Given the complexity and length of video content, standalone models often struggle to retain and reason over all relevant temporal information. Video-RAG addresses this limitation by retrieving external, contextually aligned information such as transcripts, documents, or multimodal snippets ased on cues extracted from video frames or subtitles.

The typical Video-RAG pipeline, shown in Figure 2.6, involves extracting keyframes or segment-level features using a video encoder, which are then used to formulate a query for an external retriever. The retrieved content is subsequently combined with the video representation and passed to a language model or multimodal generator to produce coherent, grounded outputs. This architecture enables the model to efficiently handle long-range dependencies and augment its reasoning with up-to-date or domain-specific knowledge [23].

Video-RAG has demonstrated effectiveness in tasks that require both fine-grained temporal reasoning and factual accuracy. By fusing visual, textual, and retrieved external modalities, it significantly improves the model's capacity to understand and generate language grounded in complex video data.

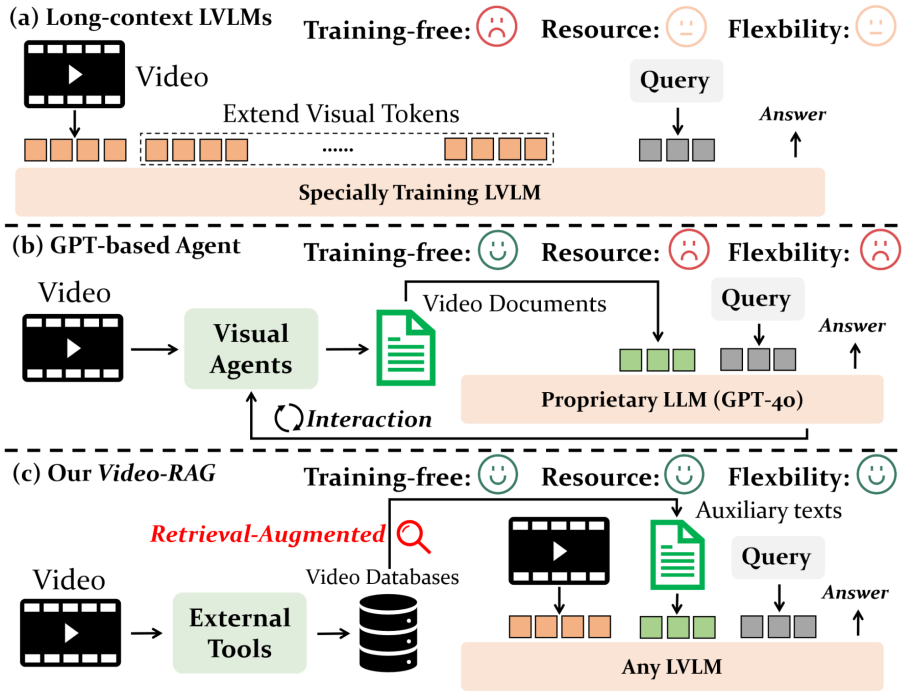


Figure 2.6.: Video-RAG architecture: Illustration of different video understanding approaches alongside Video-RAG. Video-RAG provides a resource-efficient, training-free pipeline that is easily compatible with any LVLM. By leveraging RAG, it retrieves auxiliary texts for input, leading to notable performance enhancement. The graph is referenced from [35].

2.6. RAGAS Evaluation Metrics

Retrieval-Augmented Generation Assessment (RAGAS) [14] [15] [44] is a framework for evaluation of Retrieval-Augmented Generation (RAG) pipelines. This evaluation strategy employs large language models (LLMs) to assess the faithfulness and relevance of the generated answers with respect to the ground truth. However, relying on an LLM as the critic model introduces a degree of uncertainty in the evaluation process, given the subjective nature and potential biases of the model.

2.6.1. Text-RAG Evaluation Metrics

Each of the following metrics was scored using functions defined by the RAGAS framework [15]. Each function employs a critique large language model (LLM) agent, guided by a task-specific prompt that outlines the evaluation criteria and scoring methodology. The agent compares the generated response with the corresponding ground truth to determine an evaluation score.

Faithfulness

Faithfulness, as shown in equation 2.1, measures the factual consistency of the generated answer against the provided context.

$$\text{Faithfulness score} = \frac{|\text{Number of claims in the generated answer that can be inferred}|}{|\text{Total number of claims in the generated answer}|} \tag{2.1}$$

Answer Relevance

Answer Relevance equation 2.2 focuses on how relevant the generated answer is to the given query. The Answer Relevancy is defined as the mean cosine similarity of the original question to a number of artifical questions, which where generated (reverse engineered) by LLM based on the answer.

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N}\cos(E_{g_i}, E_o) \tag{2.2}$$

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N}\frac{E_{g_i}\cdot E_o}{\|E_{g_i}\|\|E_o\|} \tag{2.3}$$

Where:

- E_{g_i} is the embedding of the generated question i.
- E_o is the embedding of the original question.
- N is the number of generated questions, which is 3 by default.

Context Recall

Context Recall equation 2.4 measures the extent to which the retrieved context aligns with the ground truth answer.

$$\text{Context Recall} = \frac{|\text{ground truth that can be attributed to context}|}{|\text{Total number of claims in the ground truth}|} \tag{2.4}$$

Context Precision

Context Precision equation 2.5 evaluates if all the ground truth relevant items available in the contexts are ranked higher or not.

$$\text{Context Precision} = \frac{|\sum_{k=1}^{K} Precision@k \times u_k|}{|\text{Total number of relevant items in the top K results}|} \tag{2.5}$$

Where:

- The numerator term
$$\left|\sum_{k=1}^{K} Precision@k \times u_k\right|$$
represents the sum of the precision at each rank k, weighted by the binary relevance indicator u_k, where $u_k = 1$ if the item at rank k is relevant, and $u_k = 0$ otherwise. This effectively accumulates the precision scores of only the relevant items among the top K retrieved results.
- K is the total number of chunks in contexts and $u_k \in 0, 1$ is relevant indicator at rank k.

Context Utilization

Context Utilization equation 2.6 evaluates if all the answers relevant items available in the contexts are ranked higher or not.

$$\text{Context Utilization} = \frac{|\sum_{k=1}^{K} Precision@k \times u_k|}{|\text{Total number of relevant items in the top K results}|} \quad (2.6)$$

Where:

- K is the total number of chunks in contexts and $u_k \in 0, 1$ is relevant indicator at rank k.

2.6.2. Multimodal-RAG Evaluation Metrics

Multimodal Faithfulness

Multimodal Faithfulness metric measures the factual consistency of the generated answer against both visual and textual context. It is calculated from the answer, retrieved textual context, and visual context. The answer is scaled to a (0,1) range, with higher scores indicating better faithfulness. The generated answer is regarded as faithful if all the claims made in the answer can be inferred from either the visual or textual context provided. To determine this, faithfulness can be inferred by comparing the model's response to a known correct answer. A faithfulness score of 1 is assigned if the model's answer either exactly matches or is semantically equivalent to the ground truth, whereas a score of 0 is given if the response contradicts the correct answer.

Multimodal Relevance

Multimodal Relevance metric measures the relevance of the generated answer against both visual and textual context. It is calculated from the user input, response, and retrieved contexts (both visual and textual). The answer is scaled to a (0,1) range, with higher scores indicating better relevance. The generated answer is regarded as relevant if it aligns with the visual or textual context provided. To determine this, the response is directly evaluated against the provided contexts, and the relevance score is either 0 or 1.

Semantic Similarity

The concept of Answer Semantic Similarity pertains to the assessment of the semantic resemblance between the generated answer and the ground truth. This evaluation is based on the ground truth and the answer, with values falling within the range of 0 to 1. A higher score signifies a better alignment between the generated answer and the ground truth. Measuring the semantic similarity between answers can offer valuable insights into the quality of the generated response. This evaluation utilizes a bi-encoder model to calculate the semantic similarity score.

3. Models and Methods

In the following chapter, the Image-RAG and Video-RAG pipelines are described in detail in Section 3.1 and Section 3.2, including the implementation of each sub-component within both architectures. Various key-frame extraction and processing techniques are outlined, along with an explanation of the research methodology employed to experiment with different pipeline hyperparameters across a range of domain-specific video use cases.

3.1. Image-RAG Pipeline

Two different Image Retrieval Augmented Generation (Image-RAG) pipelines were implemented using two methods, as shown in Figure 3.1, with two distinct inputs: Raw-Image and Image-Semantic Description. These inputs are stored in separate databases. Further details are provided in the following sections.

3.1.1. Raw-Image Input

As shown in Figure 3.1, raw images are fed into the 'ViT-g-14' model, a multi-modal embedding model, to generate embeddings for both images and text. These embeddings are then ingested into two separate collections within a multi-vector database. Subsequently, the raw images are retrieved from the vector store and fed into a multimodal LLM to generate answers based on the retrieved images.

3.1.2. Image-Description Input

In the second pipeline, 'GPT-4o', a multi-modal LLM, is used to generate text summaries from images. These text summaries are then embedded as semantic descriptions of the raw images using the 'text-embedding-3-large' model and stored

as a text collection in the vector store. The text chunks are subsequently passed to an LLM for answer synthesis.

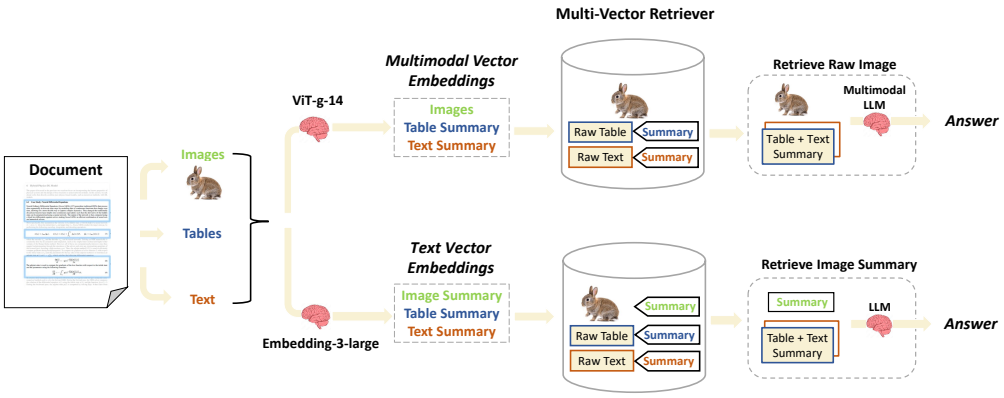
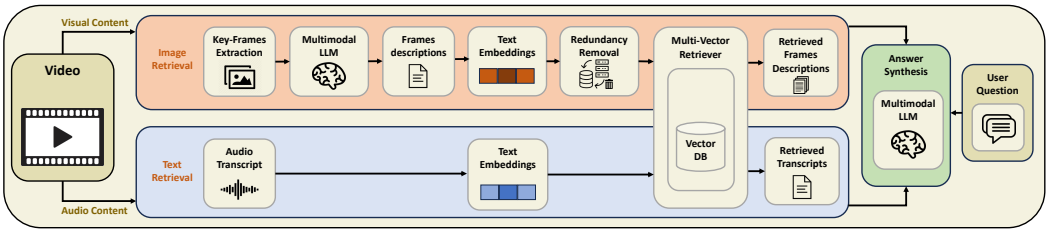
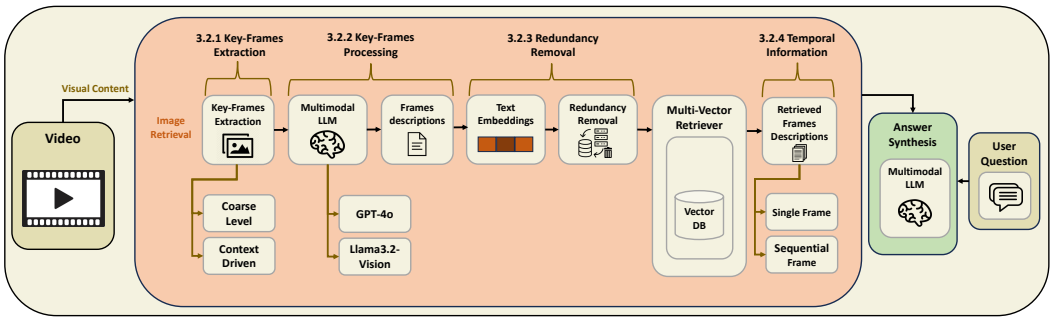


Figure 3.1.: **Image-RAG pipelines:** A multi-vector retriever framework that processes documents by extracting images, tables, and text for embedding. Two approaches are shown: multimodal embedding and text embedding. In the multimodal path, table and text summaries are generated, enabling retrieval of raw images and associated data for processing by a multimodal LLM to generate answers. The text embedding path instead generates and retrieves summaries (including of images), which are then used by a standard LLM for answer generation. This setup highlights trade-offs between raw data usage and summarized input for efficient retrieval and reasoning.



(a) Video-RAG Pipeline.



(b) Key-frames selection components in Video-RAG pipeline.

Figure 3.2.: **Video-RAG Pipeline:** (a) A video question-answering pipeline that integrates both visual and audio contents. The image retrieval branch extracts key-frames from the video, generates corresponding descriptions using a multimodal large language model (LLM), embeds these descriptions, and subsequently removes redundant embeddings. In parallel, the text retrieval branch transcribes the video's audio into text, which is then embedded for downstream processing. A multi-vector retriever pulls relevant frame descriptions and transcripts from a vector database based on a user query. Finally, a multimodal LLM synthesizes the retrieved information to generate a comprehensive answer. For an intuition regarding the audio content, which was not the primary focus of the study, Figure A.1 in the appendix presents an example. (b) A closer look at the main components of the pipeline, organized according to the visual processing phases of the video. Key-frames selection cycle starting from extracting the unique frames from the video input, storing the processed descriptions, synthesing answers about the input video. Showing different methodologies for each subcomponent of the pipeline.

3.2. Video-RAG Pipeline

Figure 3.2 illustrates the implementation of the Video-RAG pipeline, highlighting all key components used for key-frame extraction, processing, and storage, ultimately leading to the generation of the final answer based on video information. The following sections detail each component of the pipeline. Section 3.2.1 outlines key-frame extraction methodologies, while Section 3.2.2 describes the technologies used to generate semantic descriptions for the extracted frames. Redundancy removal technique is covered in Section 3.2.3, followed by an explanation of two RAG-based approaches for temporal information retrieval in Section 3.2.4.

3.2.1. Key-Frames Extraction

Coarse-Level Key-Frames Extraction Approach

In the coarse frame selection stage, a uniform sampling method is applied to extract a set of coarse frames by specifying a fixed Frames-Per-Second (FPS) value in the video frame extraction function. The term key-frames in this context refers to the coarse-level frames extracted at fixed intervals based on a predetermined FPS value. This approach prevents the selection of frames with excessively small temporal spacing and promotes sample diversity by experimenting with different FPS values tailored to the characteristics of domain-specific expert video datasets.

Various experiments were conducted to evaluate the functionality of the Video-RAG pipeline. In this context, the FPS value was determined for the coarse-level key-frame extraction approach applied to the hardware troubleshooting scenario described in Section 5.2. Several factors influenced this estimation, including the video's overall dynamics (which were moderate), the frequency and complexity of interactions particularly between the expert's hand movements and the wiring diagrams as well as the pace of movement between subcomponents during the recording. Accordingly, the FPS value was estimated based on these factors and basic motion principles, then experimentally evaluated to determine the most appropriate setting for this specific type of video content. An FPS value of 6 was selected based on the specific properties and characteristics of the recorded videos.

Context-Driven Key-Frames Extraction Approach

Figure 3.3 shows in detail how fine level key-frames are produced using CLIP-ViT approach. In the context-driven frame selection stage, the previously extracted coarse-level video frames are considered as a set F, obtained using a fixed Frames-Per-Second (FPS) value manually configured based on the characteristics of the input video. The set F is then input into the image encoder, where each frame in F is associated with a corresponding visual vector v. Next, we compute the similarity of these visual vectors v with the word embedding to obtain the similarity score. The word embeddings consist of a question–answer pair designed to focus on key-frames of interest in relation to these embeddings.

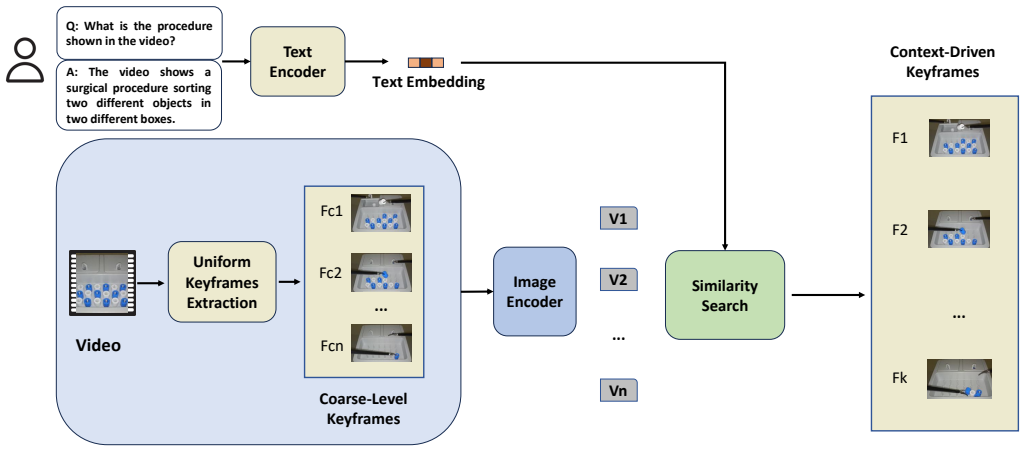


Figure 3.3.: The diagram illustrates the context-driven key-frames extraction pipeline using CLIP-ViT model, where coarse-level key-frames are extracted from a video and encoded alongside a user question for similarity search. This process identifies context-driven key-frames most relevant to the query for accurate video understanding.

This process is performed offline, executed once prior to runtime, and subsequently updated as the focus of interest in the input video changes. The similarity score is calculated as follows:

$$\text{cosine similarity} = \cos(\theta) = \text{score}(\mathbf{v}_i, \mathbf{w}) = \frac{\mathbf{v}_i \cdot \mathbf{w}}{\|\mathbf{v}_i\|\|\mathbf{w}\|} \tag{3.1}$$

where v denotes the i-th visual embedding and w represents the word embedding.

- $\mathbf{v}_i \cdot \mathbf{w} = \sum_{j=1}^{n} v_{ij} w_j$ is the dot product of the vectors $\mathbf{v}_i$ and $\mathbf{w}$.
- $\|\mathbf{v}_i\| = \sqrt{\sum_{j=1}^{n} v_{ij}^2}$ is the Euclidean norm (magnitude) of $\mathbf{v}_i$.
- $\|\mathbf{w}\| = \sqrt{\sum_{j=1}^{n} w_j^2}$ is the Euclidean norm (magnitude) of $\mathbf{w}$.
- The resulting score is the cosine of the angle θ between the two vectors, reflecting their directional similarity.

The similarity scores are then sorted, and the top k values with the highest scores are selected. Subsequently, the corresponding video frames are identified and used to form a fine-level key-frame set based on a specified similarity threshold. Finally, the selected frames are recombined into a video sequence following the original temporal order, ensuring that each dialogue text is paired with a unique set of corresponding key-frames.

3.2.2. Key-Frame Processing

One of the crucial components in the Video-RAG pipeline is the generation of key-frames descriptions. Two MM-LLMs were used to compare the frames descriptions generated by each model, employing both basic and optimized prompt definitions. The extracted key-frames are then processed using each of the two models to evaluate which model provides more detailed descriptions, ultimately ensuring better knowledge conservation from the video inputs.

In addition to setting an appropriate Frames-Per-Second (FPS) value, a prompt optimization experiment was conducted to efficiently process video frames and generate comprehensive descriptions that capture the majority of the visual content. This experiment utilized both GPT-4o and LLaMA3.2-Vision-90B, testing various prompts to evaluate the quality and completeness of the resulting descriptions across different scenarios. For example, the hardware troubleshooting case presented in Section 4.2 involved multiple influencing factors, including background noise around the hardware testbench, the complexity of expert hand movements and wiring diagrams, the resolution of the recording camera, and the presence of textual elements within the video. These factors collectively informed the design of LLM agent instructions, ensuring that the generated semantic descriptions preserved rich and relevant

GPT-4o generated semantic descritpion for example key-frame

Prompt

Describe the images as an alternative text.

Frame Description

A close-up image of a medical or laboratory device featuring a rectangular container with two closed compartments on the top. Below the compartments, there is an open section containing a grid of cylindrical objects, alternat ing in blue and white colors. Each cylinder has a small metallic pin in the center. Two metallic tools with hooked ends are positioned on either side of the container, suggesting manipulation or placement of the cylinders.

Figure 3.4.: GPT-4o generated semantic description using a baseline prompt applied to an example key-frame from the surgical procedure training video input. This example case will be introduced in detail in Section 4.2.

content from the input video. The following sections present several experiments designed to demonstrate the functionality of the Image-Description pipeline.

GPT-4o Semantic Description

Figure 3.4 shows an example of a baseline prompt and the resultant description for one of the extracted key-frames from surgical procedure training use case using the GPT-4o model. This example illustrates Image-Description pipeline and how variations in prompt formulation have been experimentally evaluated, resulting in different output descriptions. To ensure consistency across different video sets, another case Figure 3.6 was used to test the performance of the model in generating consistent key-frame semantic descriptions.

For more detailed frame descriptions, an optimized prompt was used, as shown in Figure 3.5, to highlight the difference in the resultant description. This approach provides more detailed information about the key-frame, which is expected to enable the retrieval of more accurate and comprehensive answers from the video inputs.

LLaMA3.2-Vision-90B Semantic Description

Given that the GPT-4o model incurs significant costs to process all key-frames in a short 2-minute video sample, an open-source alternative was explored to reduce the overall processing cost while generating the multimodal semantic descriptions to be stored in the database. Consequently, the LLaMA3.2-Vision-90B model was also employed with an optimized prompt definition, Figure 3.8, which produced descriptions quite similar to those generated by the GPT-4o model. More detailed results are numerically explained in Chapter 5. Here, examples are provided to illustrate the effectiveness of the prompt definition for both models.

As shown in Figure 3.6, the GPT-4o model can still generate highly detailed descriptions even with a simple prompt definition.

Figure 3.7 shows the description generated by the LLaMA3.2-Vision-90B model using the same prompt as the one used for the GPT-4o model. However, to achieve a comparable description to the one generated by GPT-4o, the prompt definition for the LLaMA3.2-Vision-90B model required refinement. This refinement was supposed to achieve a sufficient level of knowledge conservation across different videos within the overall pipeline.

To evaluate the consistency of the LLaMA3.2-Vision-90B model across different use cases, the refined prompt was tested on a surgical procedure scenario, as illustrated in Figure 3.9.

3.2.3. Redundancy Removal

After storing the embeddings into the vector database, similar embeddings from frames with little to no key differences can arise, particularly when using a coarse-level frame extraction approach with higher FPS values. To address this, a threshold value is set to remove vector embeddings that have a similarity score equal to or greater than the threshold. The similarity search is conducted using the cosine similarity function, as shown in eq. (3.1).

GPT-4o generated semantic descritpion for example key-frame

Prompt

Provide a detailed semantic description about the provided image; including all segments, their relative positions in the image, their colors, their relative sizes, their counts if more from same segment, their relative motion and interaction between them.

Frame Description

The image depicts a rectangular container with a white or light gray color. The container is divided into two main sections: the upper section and the lower section.

Upper Section
Position: Located at the top of the container.
Components: Two square lids with handles.
- **Lids**:
  - **Color**: White or light gray, matching the container.
  - **Handles**: Each lid has a small, curved handle in the center.
  - **Relative Position**: The lids are side by side, with a small gap between them.

Lower Section
Position: Located below the upper section, occupying the majority of the container's space.
Components: A grid of cylindrical objects.
- **Cylinders**:
  - **Count**: 16 cylinders in total.
  - **Arrangement**: Arranged in a 4x4 grid.
  - **Colors**: Alternating blue and white.
    - **Blue Cylinders**: 8 in total.
    - **White Cylinders**: 8 in total.
  - **Relative Position**: The cylinders are evenly spaced within the grid.
  - **Size**: All cylinders appear to be of the same size.
  - **Top View**: Each cylinder has a central hole or indentation.

Additional Elements
Tools: Two metallic tools with black handles.
- **Position**: One tool is on the left side and the other on the right side of the container.
- **Interaction**: The tools are positioned as if they are about to interact with the cylinders, possibly for picking or placing them.

Relative Motion and Interaction
Cylinders: Stationary within the grid.
Tools: Positioned to potentially interact with the cylinders, suggesting a possible motion of picking or placing the cylinders.

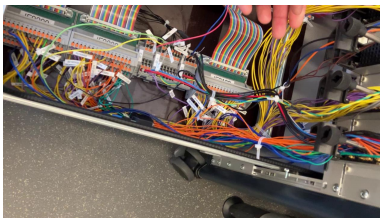
Summary
The image shows a container with two sections: an upper section with two lids and a lower section with a 4x4 grid of blue and white cylinders. Two metallic tools are positioned on either side of the container, indicating pote ntial interaction with the cylinders. The overall color scheme is white or light gray for the container and alternating blue and white for the cylinders.

Figure 3.5.: GPT-4o generated semantic description using an optimized prompt applied to an example key-frame from the surgical procedure training video input.

GPT-4o generated semantic descritpion for example key-frame

Prompt

Provide a detailed semantic description about the provided image; including all segments, their relative positions in the image, their colors, their relative sizes, their counts if more from same segment, their relative motion and interaction between them.



Frame Description

The image depicts a section of an electronic control unit (ECU) setup, likely part of a larger testing or development environment. The image can be divided into several key segments:

1. **ECU Modules**:
   - **Count**: There are four ECU modules labeled "ECU 1" to "ECU 4".
   - **Position**: These modules are located in the upper part of the image, arranged horizontally from left to right.
   - **Color**: The modules themselves are primarily gray with white labels and black connectors.
   - **Relative Size**: Each module is of similar size, occupying a small portion of the overall image.
   - **Interaction**: Each module has multiple wires connected to it, indicating they are part of a larger circuit or system.

2. **Wiring**:
   - **Count**: Numerous wires are connected to the ECU modules, with a variety of colors including yellow, orange, purple, blue, black, and more.
   - **Position**: The wires extend from the ECU modules downwards and towards the bottom part of the image.
   - **Color**: The wires are multi-colored, with each color likely representing different signals or functions.
   - **Relative Size**: The wires are thin compared to the ECU modules but occupy a significant portion of the image due to their length and spread.
   - **Interaction**: The wires are bundled together in some places with zip ties, indicating organized routing. They connect the ECU modules to other components in the lower part of the image.

3. **Ribbon Cables**:
   - **Count**: There are several ribbon cables visible, with one prominent multi-colored ribbon cable.
   - **Position**: The ribbon cables are located in the lower left part of the image, extending horizontally.
   - **Color**: The prominent ribbon cable has a striped pattern with multiple colors.
   - **Relative Size**: The ribbon cables are wider than the individual wires but shorter in length.
   - **Interaction**: The ribbon cables are connected to a circuit board or connector block, indicating they are part of the data transmission system.

4. **Connector Blocks and Circuit Boards**:
   - **Count**: Multiple connector blocks and at least one visible circuit board.
   - **Position**: These components are located in the lower part of the image, beneath the ECU modules and wiring.
   - **Color**: The connector blocks are primarily gray with some orange and green elements, while the circuit board is green with various components.
   - **Relative Size**: These components are larger than the wires but smaller than the overall ECU setup.
   - **Interaction**: The wires and ribbon cables are connected to these blocks and boards, indicating they are part of the electrical connections and data pathways.

5. **Drawer and Frame**:
   - **Count**: One drawer and part of the frame of the setup.
   - **Position**: The drawer is in the lower right part of the image, partially open, with the frame visible on the right edge.
   - **Color**: The drawer and frame are metallic gray.
   - **Relative Size**: The drawer is large, occupying a significant portion of the lower right area.
   - **Interaction**: The drawer holds the connector blocks and circuit boards, providing a structured environment for the components.

6. **Floor and Background**:
   - **Count**: One visible floor area and part of the background.
   - **Position**: The floor is in the bottom right corner, with the background visible on the right side.
   - **Color**: The floor is dark gray with a speckled pattern, and the background is a mix of gray and black.
   - **Relative Size**: The floor and background occupy a small portion of the image.
   - **Interaction**: The floor provides a base for the setup, and the background indicates the environment in which the setup is placed.

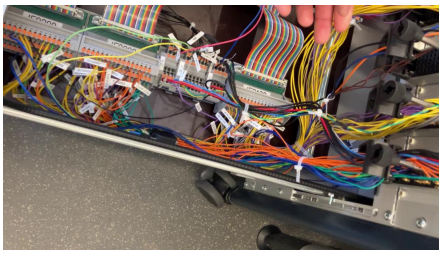
Overall, the image shows a well-organized electronic setup with multiple ECU modules, a variety of wires and cables, and several connector blocks and circuit boards, all housed within a drawer and frame structure. The components are interconnected, indicating a complex system likely used for testing or development purposes.

Figure 3.6.: GPT-4o generated semantic description using an optimized prompt applied to an example key-frame from the hardware (HW) troubleshooting video set.

LLaMA3.2-Vision-90B generated semantic descritpion for example key-frame

Prompt

Provide a detailed semantic description about the provided image; including all segments, their relative positions in the image, their colors, their relative sizes, their counts if more from same segment, their relative motion and interaction between them.



Frame Description

The image depicts a complex arrangement of wires and cables within an electronic device or system. The visible components include:

- **Wires and Cables:**
  * Multiple wires and cables are bundled together, with some appearing to be connected to various ports or connectors.
  * The colors of the wires vary, including red, blue, yellow, green, orange, and white.
  * Some wires have labels or markings on them, but they are not legible in the image.
- **Electronic Components:**
  * Several electronic components are visible, such as circuit boards, capacitors, and resistors.
  * These components are attached to a metal frame or chassis, which provides structural support for the device.
- **Connectors and Ports:**
  * Various connectors and ports are present, including USB ports, Ethernet ports, and power outlets.
  * Some of these connectors appear to be connected to wires or cables, while others seem to be unused.
- **Cable Management:**
  * The cables and wires are managed using cable ties, clips, and other organizational tools.
  * This suggests that the device is designed for efficient cable management and ease of maintenance.

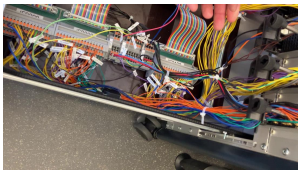
In summary, the image shows a complex electronic device with numerous wires, cables, and components. The device appears to be well-organized, with careful attention paid to cable management and component placement. However, without more context or information about the specific device, it is difficult to provide further details about its purpose or functionality.

Figure 3.7.: LLaMA3.2-Vision-90B generated semantic description using an optimized prompt applied to an example key-frame from the HW troubleshooting video set. This example case will be introduced in detail in Section 4.2.

LLaMA3.2-Vision-90B generated semantic descritpion for example key-frame with fine-detailed prompt

Prompt

Analyze the given image with the highest level of detail possible.
- Break the description into structured sections:
- **Overall Composition**: Describe the image's overall layout and theme.
- **Foreground Elements**: List all visible elements in the foreground, their shapes, sizes, and positions relative to each other.
- **Background Elements**: Describe the background objects, colors, and textures.
- **Colors & Lighting**: Mention dominant colors, shading, and how lighting affects the scene.
- **Object Count & Arrangement**: Provide accurate counts for repetitive objects and their spatial organization.
- **Interactions & Movement**: If elements interact (e.g., overlapping, shadows, reflections, motion), describe them explicitly.
- Use precise adjectives and spatial relationships (e.g., 'centered', 'adjacent to', 'partially overlapping'



Frame Description

Overall Composition The image presents a detailed view of an electronic control panel or server rack, showcasing its internal components and wiring. The theme revolves around the intricate organization and complexity of modern electronics.

Foreground Elements
Wires and Cables: A multitude of wires and cables in various colors (red, blue, yellow, green, black) are visible, connecting different parts of the system. These wires vary in thickness and are neatly organized using cable ties.
Position: They span across the entire width of the image, with some extending from the top to the bottom.
Size: The sizes range from thin (likely for data transmission) to thicker (possibly for power supply).
Electronic Components: Several electronic components such as circuit boards, capacitors, and resistors are scattered throughout the panel. These components are attached to the main board or placed on shelves within the rack.
Position: They are positioned both horizontally and vertically across different levels of the rack.
Size: The sizes vary significantly, from small surface-mount devices to larger components like fans and power supplies.
Fans and Ventilation Systems: At least two fans are visible, one at the top and another near the bottom. These fans contribute to cooling the system by circulating air through vents.
Position: One fan is located at the very top of the image, while the other is situated towards the lower section.
Size: Both fans appear to be standard sizes used in computer systems for airflow management.

Background Elements
Metal Frame and Mounting Brackets: The background consists of a metal frame that houses all the components. This frame includes mounting brackets for securing boards, drives, and other peripherals.
Color: The metal has a silver finish with some parts appearing slightly darker due to shadows or oxidation.
Texture: The surface appears smooth but may have slight ridges or patterns from manufacturing processes.

Colors & Lighting
Dominant Colors: The dominant colors are shades of gray (from the metal), black (wires, components), and silver (screws, connectors). Accents include red, blue, yellow, green, and orange from wires and LED indicators.
Lighting Effects: Soft, diffused lighting illuminates the scene without harsh shadows, suggesting an indoor environment with controlled lighting conditions. This helps in highlighting details across the image.

Object Count & Arrangement
Wires/Cables: Approximately 20-25 distinct wires or cable bundles are visible.
Electronic Components: There are around 10-15 identifiable electronic components (boards, capacitors, resistors).
Fans/Ventilation Systems: Two fans are clearly visible.

Interactions & Movement
* No direct interactions between elements are depicted beyond their functional connections. However, the presence of fans implies movement within the system for cooling purposes.
* Shadows cast by components and wires indicate depth but do not suggest any motion or dynamic interaction at the moment captured in the image.

Figure 3.8.: LLaMA3.2-Vision-90B generated semantic description using a more optimized prompt applied to an example key-frame from the HW troubleshooting video set. In this case, the prompt is more detailed, with information clustered to guide the generation of a description within a specific layout, thereby preserving richer information.

LLaMA3.2-Vision-90B generated semantic descritpion for example key-frame with fine-detailed prompt

Prompt

Analyze the given image with the highest level of detail possible.
- Break the description into structured sections:
- **Overall Composition**: Describe the image's overall layout and theme.
- **Foreground Elements**: List all visible elements in the foreground, their shapes, sizes, and positions relative to each other.
- **Background Elements**: Describe the background objects, colors, and textures.
- **Colors & Lighting**: Mention dominant colors, shading, and how lighting affects the scene.
- **Object Count & Arrangement**: Provide accurate counts for repetitive objects and their spatial organization.
- **Interactions & Movement**: If elements interact (e.g., overlapping, shadows, reflections, motion), describe them explicitly.
- Use precise adjectives and spatial relationships (e.g., 'centered', 'adjacent to', 'partially overlapping'

Frame Description

Overall Composition The image presents a close-up view of two white plastic trays, each containing 16 cylindrical objects in blue and white colors.
The trays are positioned side by side, with the left tray slightly closer to the camera than the right one.

Foreground Elements
Trays: Two identical white plastic trays are visible, each measuring approximately 10 cm in length, 5 cm in width, and 2 cm in height.
- The trays have a flat bottom and straight sides, with a small lip around the edges to prevent objects from falling out.
- Each tray contains 16 cylindrical objects arranged in four rows of four columns.

Cylindrical Objects: There are 32 cylindrical objects in total, divided evenly between blue and white colors.
- The objects are made of plastic or ceramic material and have a smooth surface finish.
- They measure approximately 1 cm in diameter and 2 cm in height.
- The blue objects are placed in the top-left and bottom-right corners of each tray, while the white objects occupy the remaining positions.

Background Elements
Wall: A plain beige wall serves as the background for the image.
- The wall is painted with a matte finish and has no visible patterns or textures.
- It provides a neutral backdrop that allows the viewer to focus on the trays and their contents.

Colors & Lighting
Dominant Colors: The dominant colors in the image are white (from the trays) and blue/white (from the cylindrical objects).
- The beige color of the wall adds warmth to the scene without overpowering the other elements.

Lighting: Soft, diffused lighting illuminates the scene from above, casting minimal shadows on the objects.
- The lighting helps to accentuate the textures and colors of the trays and cylindrical objects.

Object Count & Arrangement
Trays: There are two trays in total, each containing 16 cylindrical objects.
- The trays are arranged side by side, with a small gap between them.

Cylindrical Objects: There are 32 cylindrical objects in total, divided evenly between blue and white colors.
- Each tray contains eight blue objects and eight white objects, arranged in alternating rows.

Interactions & Movement
No Interactions or Movement: The image does not depict any interactions or movement between the trays or cylindrical objects.
- All elements appear stationary and static within the frame.

Figure 3.9.: LLaMA3.2-Vision-90B generated semantic description using a more optimized prompt applied to an example key-frame from the surgical procedure training video set. LLaMA3.2-Vision-90B generated semantic description with similar structure as in Figure 3.8 for the HW troubleshooting case.

3.2.4. Temporal Information

The ability to capture temporal information has been critical to the development of video understanding models. While there have been numerous attempts at modeling motion in videos, an explicit analysis of the effect of temporal information on video understanding is still lacking. In this work, we aim to investigate the following research question: How important is motion in a video for recognizing sequential actions? This question is explored within two specific contexts: (i) a hardware troubleshooting scenario, where a worker uses hand gestures to trace a specific wire associated with a malfunction, and (ii) a surgical procedure training scenario, where anomalous tool behavior occurs as surgical instruments struggle to grasp objects. Further details regarding both use cases will be provided in the following chapter. To this end, Two approaches have been evaluated: (i) using single frame descriptions as the knowledge base, (ii) using sequential frames descriptions as the knowledge base. This section of the research employed a coarse-level key-frame extraction approach for both technologies.

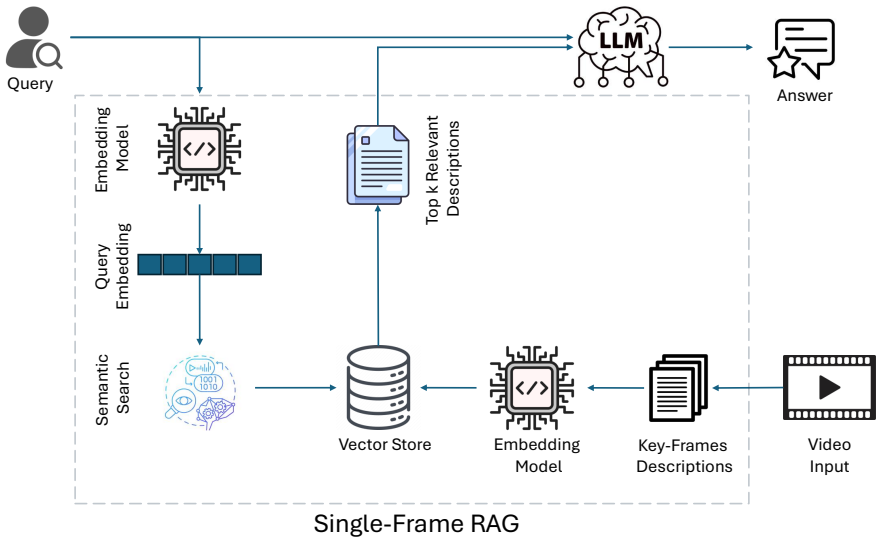


Figure 3.10.: Single-Frame Video-RAG Pipeline

Single-Frame Video-RAG

In the single frame approach, frames descriptions are ingested separately into the vector store, and the retrieved information depends on the descriptions of individual frames. Given a user query, the top k relevant frames are retrieved using a cosine similarity search between the query and the descriptions stored in the vector database as shown in Figure 3.10. The most relevant frame description is then combined with the query and passed to the LLM for answer synthesis.

Sequential-Frame Video-RAG

Figure 3.12 shows how sequential frames descriptions are generated. As usual, each individual frame description is ingested into the database, and then the top k retrieved descriptions are then selected based on a cosine similarity search between the user query and the descriptions stored in the database. A window w is taken around each of the top k retrieved descriptions to capture the $+w$ and $-w$ neighboring frames descriptions as illustrated in Figure 3.11. These neighboring frames descriptions are then fed as the retrieved context, along with the target frame and the user query. An additional refinement step involves retrieving the top k relevant frames for each of the neighboring frames identifiers, resulting in a total of k x $2w$ descriptions to be combined with the user query for answer synthesis. In this approach, the most relevant frame is provided alongside its neighboring frames, which may contain useful interactions between the required objects in the video. This method is supposed to enhance the model's ability to understand sequential interactions and temporal relationships in the video.

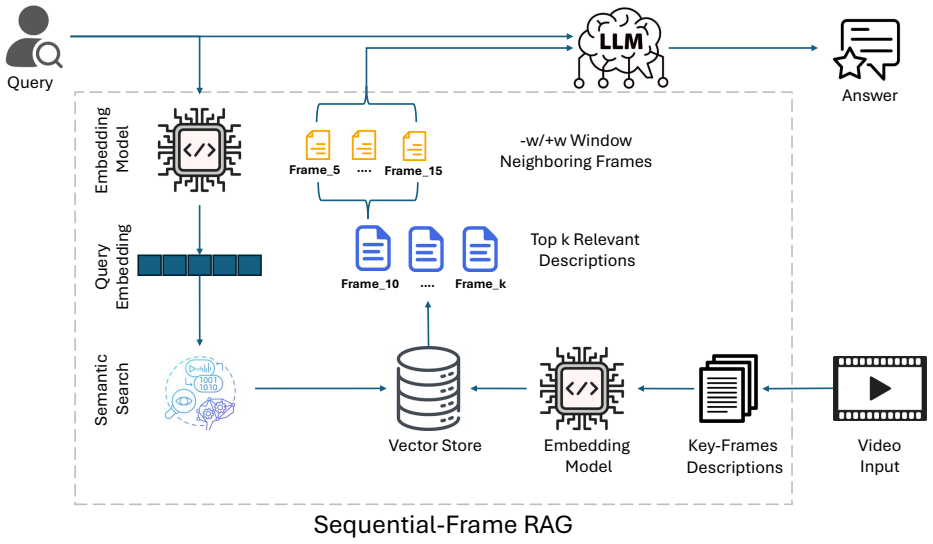


Figure 3.11.: Sequential-Frame Video-RAG Pipeline

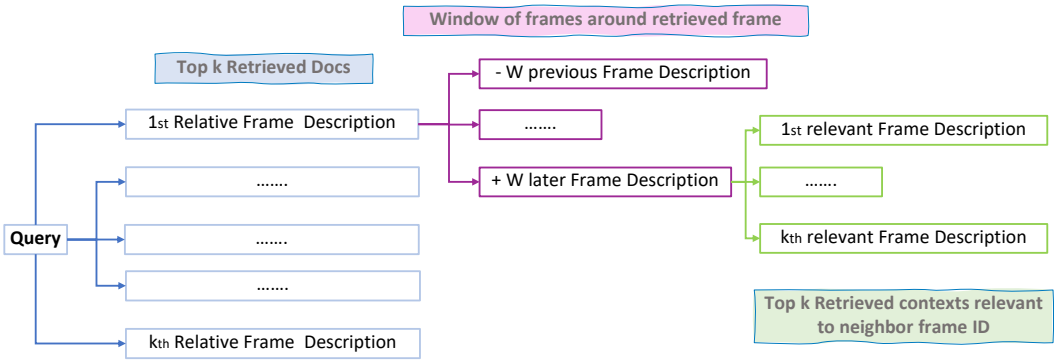


Figure 3.12.: Context Retrieval of Sequential-Frame in Video-RAG

4. Evaluation Setup

To evaluate the pipelines outlined in Chapter 3, the Visual Question–Answer (VQA) test set from the Hugging Face framework [34] was utilized as the academic benchmark. Additionally, domain-specific test sets were annotated for a range of real-world use-cases, curated by domain experts from various departments within the BOSCH Power Solutions – Diesel Components (PS-DC) business unit.

4.1. Academic Multimodal Test Set

LiveXiv [49] is a scalable, evolving live benchmark built upon scientific ArXiv papers. It continuously accesses domain-specific manuscripts at any given timestamp and proposes the automatic generation of VQA pairs. The dataset includes approximately 9,000 Tabular Question–Answer (TQA) pairs and 10,400 VQA pairs. A subset of around 50 VQA pairs, available through the Hugging Face community, has been reformulated, as illustrated in Figure 4.1, and utilized as the academic multimodal test set for evaluating the Image-RAG pipeline.

4.1.1. Raw-Image Pipeline

As shown in Figure 4.1, the academic test set has been reformulated into a collection of questions, corresponding ground truth answers, and a directory of raw images. Additionally, responses generated by the LLM are included as the actual answers for evaluation purposes. Each image, along with its corresponding question, was provided as multimodal input to the multimodal large language model (MM-LLM) to generate responses. Subsequently, the RAGAS framework was used to evaluate these responses, where a critic LLM agent assessed the generated answers against the ground truth in order to determine their multimodal faithfulness and relevance within the context of the evaluation pipeline.

4.1.2. Image-Description Pipeline

Figure 4.2 presents a set of questions, corresponding ground truth answers, and GPT-4o generated descriptions for each of the raw images previously downloaded from the Hugging Face community page [34]. This test set was used to evaluate the Image-Description Pipeline. Each question, along with the corresponding image description, was provided to a LLM to generate an answer. This generated answer was then evaluated against the ground truth using a critic LLM agent, based on the evaluation metrics defined by the RAGAS framework.

	user_input	retrieved_contexts	response	reference
0	Which method reaches around 1000 cumulative re...	['Images_2_crawl_Bytes50_only\\image_49_385a05...	The "manual" method reaches around 1000 cumula...	Linear
1	Which detector maintains a constant BER across...	['Images_2_crawl_Bytes50_only\\image_49_c6feee...	The detector that maintains a constant BER (Bi...	MPA
2	In panel (a), which state exits downward after...	['Images_2_crawl_Bytes50_only\\image_49_9a423b...	In panel (a), the state that exits downward af...	|Ψ(t)$_{cat}$)$_{t}$
3	Which layer in the first stack is highlighted ...	['Images_2_crawl_Bytes50_only\\image_49_528b42...	The second layer in the first stack is highlig...	Layer 2
4	Which model has the most compact and defined s...	['Images_2_crawl_Bytes50_only\\image_49_8f5fb8...	The model with the most compact and defined sh...	LFM-8
44	Which model's validation loss peaks above 20 a...	['Images_2_crawl_Bytes50_only\\image_49_e09556...	The model whose validation loss peaks above 20...	ResNet18
45	Which sub-image illustrates the final optimize...	['Images_2_crawl_Bytes50_only\\image_49_52e452...	The provided image does not illustrate any seg...	(f)
46	Which condition has the smallest range of inqu...	['Images_2_crawl_Bytes50_only\\image_49_52e452...	The condition with the smallest range of inqui...	OKB with default location
47	Which algorithm shows a runtime outlier above ...	['Images_2_crawl_Bytes50_only\\image_49_951996...	The algorithm that shows a runtime outlier abo...	RC-PAMO*
48	What is the color of the connectors visible on...	['Images_2_crawl_Bytes50_only\\image_49_dba854...	The image shows a set of wires with connectors...	Green

Figure 4.1.: VQA test set sample for Raw-Image MM-RAG pipeline Evaluation

	user_input	retrieved_contexts	response	reference
0	Which method reaches around 1000 cumulative re...	['A line graph titled "Cumulative Reward" with...	Based on the provided context from the retriev...	Linear
1	Which detector maintains a constant BER across...	['The image is a line graph depicting the Bit ...	Based on the provided context from the retriev...	MPA
2	In panel (a), which state exits downward after...	['**Figure Description:**\n\n- **Figure (a):**...	In panel (a) of the described figure, the stat...	|Ψ(t)$_{cat}$)$_{t}$
3	Which layer in the first stack is highlighted ...	['The image consists of six sub-images labeled...	Based on the provided context, the image that ...	Layer 2
4	Which model has the most compact and defined s...	['The image shows three grayscale 3D reconstru...	Based on the provided context, the model label...	LFM-8
44	Which model's validation loss peaks above 20 a...	['The image is a line graph titled "Validation...	Based on the provided context from the image t...	ResNet18
45	Which sub-image illustrates the final optimize...	['The image consists of six sub-images labeled...	Based on the provided context, the sub-image t...	(f)
46	Which condition has the smallest range of inqu...	['The image is a box plot chart comparing the ...	Based on the provided descriptions of the imag...	OKB with default location
47	Which algorithm shows a runtime outlier above ...	['The image consists of two box plots comparin...	Based on the provided context, the algorithm t...	RC-PAMO*
48	What is the color of the connectors visible on...	['A metallic mechanical setup is mounted on a ...	Based on the provided context, the color of th...	Green

Figure 4.2.: VQA test set sample for Image-Description MM-RAG pipeline Evaluation

4.2. Real Use Case Video Test Set

For various real-world use cases, both within the company and externally, a set of domain-expert question–answer (QA) pairs was crafted through an experimental collaboration with subject matter experts. Two primary use-case evaluations were

conducted: (i) A surgical procedure was recorded twice for training purposes, (ii) A series of 6 videos were recorded in a company laboratory, where an engineer demonstrates hardware troubleshooting techniques to workers.

Table 4.1.: Domain-Experts Test Sets Statistics

Property	Medical	Industrial
Task Category	Surgical Procedure Training	Hardware Troubleshooting
No. of Videos	2 videos	6 videos
Duration	90-150 seconds	60-150 seconds 60 seconds for software interface and setup Higher for bug fix videos
QA Pairs	15 questions	20 questions
No. of Frames	Around 96 for 90 sec	Around 128 for 120 sec
Frame Resolution	864x540 Base64-encoded	864x540 Base64-encoded
Semantic Descriptions Size	approximately 470 tokens	approximately 763 tokens

4.2.1. Surgical Procedure Training

Figure 4.3 illustrates a sample of the Question-Answer (QA) pairs conducted across various class levels. The test set is based on surgical training videos, with questions designed to assess comprehension of procedural steps, tool usage, and anomalies detection. This setup demonstrates the model's capability to comprehend and reason over domain-specific visual content in the context of medical training tasks.

Two sample videos were collected from real surgical procedure training sessions conducted by a single medical student. The first video was recorded at the early stage of the learning curve, where a higher frequency of procedural errors was observed. In contrast, the second video was captured near the end of the learning curve, representing a more refined performance with fewer errors, indicating learning saturation. As illustrated in Figure 3.9, a representative frame from the recording depicts two colorful cylindrical objects alongside two surgical tools attempting to sort them into two separate boxes. This scenario is designed to test the AI model's ability to track movements, recognize sequential steps within the video, and detect any anomalies occurring during the procedure. As stated in Table 4.1 The test set consists of two videos, One video has a duration of approximately 90 seconds, while the other extends to around 150 seconds, and includes around 15 QA pairs designed for the surgical procedure training scenario. Using the QA pairs, the RAGAS framework was employed to evaluate the performance of the Video-RAG pipeline in generat-

ing answers based on video content. The generated responses were compared with expert-provided ground truth answers to assess their faithfulness and relevance with respect to both the questions and the retrieved video content.

	Class	Question	Answer
0	Action-Level	towards which door is the tool on right side moving after picking up the white object?	it is moving towards the left door
1	Action-Level	towards which door is the tool on left side moving after picking up the blue object?	it is moving towards the door on the right side
2	Action-Level	which tool is used for opening the left door	the left tool is used to open the left door
3	Action-Level	which tool is opening the door on right side	the right tool is used to open the door on right side
4	Action-Level	how many attempts from both tools were needed to grasp the peg	two attempts were needed; one tool to pick the peg and the other to open the door
10	Object-Level	what is the color of the peg that the left tool is picking up	the left tool is picking a blue peg
11	Object-Level	what is the color of the peg that the tool on right side is picking up	the right tool is picking a white peg
12	Object-Level	how many pegs of each color is visible at the begining of the procedure	9 blue and 9 white pegs are seen
13	Object-Level	what is position of the left instrument at the end of the procedure	it is localted above the left door, in the top right
14	Reason-Level	how many pegs of white color are under the left door at the beggining	there no pegs
15	Reason-Level	how many pegs of blue color are under the door on the right side at the end	9 blue pegs in total

Figure 4.3.: VQA test set sample of surgical procedure training use case for Video-RAG evaluation

4.2.2. Test-Bench Hardware Troubleshooting

Figure 4.4 shows a sample of the QA pairs conducted across various troubleshooting scenarios in the context of hardware testing. The test set is based on hardware troubleshooting videos demonstrating issues on a testbench, with questions designed to evaluate comprehension of diagnostic procedures, tool usage, and the identification of hardware components. This setup demonstrates the model's capability to comprehend and reason over domain-specific visual content in the context of hardware troubleshooting tasks. The videos focus on identifying and resolving hardware bugs that may occur in daily workflows. Specifically, they document two faulty hardware connections and the process of tracing the issues using both software tools and physical inspection of the hardware. As shown in the snippet in Figure 3.8, the expert's hand is visible tracing the faulty connections, serving as a test of the AI model's ability to perform gesture tracking and follow the physical hand movement

to infer the color of the defective wire. The test set includes six videos, each ranging from 60 to 150 seconds in duration, with 20 QA pairs developed for the hardware troubleshooting scenario as shown in Table 4.1.

	Class	Question	Answer
0	Object-Level	What is the tool the used to unplug the wires?	the tool used to unplug the wires is a screwdriver.
1	Object-Level	What is the color of the screw driver, and is it manual or automatic	the color of the screwdriver is blue black and is manual
2	Object-Level	What is the color of the wire that had a problem and a person was holding and adjusting it ?	the wire that had a problem is the blue wire
3	Object-Level	What is the company name written in the hardware testbench that contains all the hardware ECUs and wiring?	the company name written in the hardware testbench that contains all the hardware ECUs and wiring is "dSPACE."
4	Object-Level	What is the name of the software used to identify the problem?	The software used to control and identify the problem is called ControlDesk
10	Expert-Level	What safety instructions / regulations must be considered when working at the test bench or doing changes to it's hardware configuration?	Shoes have to meet the ESD regulations, to prevent damage of the sensitive electronic. Before working at the test bench hardware turn of the power consumption and main switch of the test unit.
11	Expert-Level	What resources are needed for the repair action (time/equipment)?	SW (ControlDesk), Tools (Screw driver, Multimeter), the action will take approximately 1 hour.
12	Expert-Level	What is the problem with the test bench?	The shutoff path is not working as expected.
13	Expert-Level	How do you know the correct hypertac connection lockation at the test bench?	Labels are attached above the hypertac connectors which are located to the outside of the test bench.
14	Expert-Level	Could you use an alternative tool to unlock the clamps and inject wires?	Yes, it could be used every tool similar to a screw driver to press the unlock pin at the clamps.
15	Expert-Level	Is a special training required to carry out maintenance work on the test bench?	No if you follow the safety instrictions.

Figure 4.4.: VQA test set sample of test-bench hardware troubleshooting use case for Video-RAG evaluation

5. Evaluation Results

5.1. Academic Multimodal Test Set

The academic test sets described in the previous chapter were used to evaluate both Image-RAG pipelines. The RAGAS evaluation framework was employed to calculate the evaluation metrics for the text-based and multimodal vector stores utilized in both pipelines.

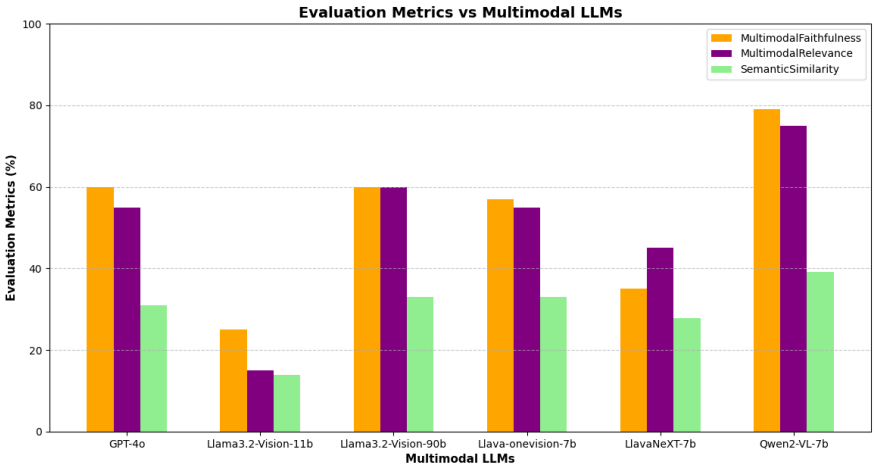


Figure 5.1.: VQA evaluation metrics over Raw-Image MM-RAG pipeline with different MM-LLMs using academic VQA test set in Figure 4.1.

5.1.1. Raw-Image Pipeline

Figure 5.1 illustrates the evaluation of the Raw-Image RAG pipeline using various MMLLMs, including GPT-4o, LLaMA3.2-Vision-90B, LLaVA-OneVision-7B, LlavaNeXT-7B, and Qwen2-VL-7B. Among these models, Qwen2-VL-7b outper-

formed the others in answering questions about the images, achieving a multimodal faithfulness of 79% and a multimodal answer relevancy of 75%. Both GPT-4o and LLaMA3.2-Vision-90B attained a multimodal faithfulness of approximately 60%, with LLaMA3.2-Vision-90B being particularly notable for its open-source nature, making it a viable alternative to GPT-4o.

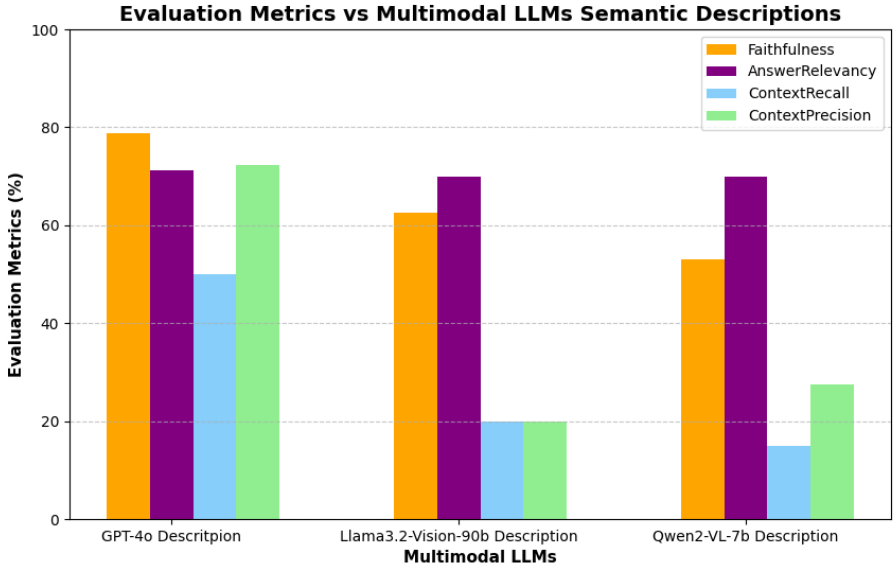


Figure 5.2.: VQA evaluation metrics over Image-Description MM-RAG pipeline with different MM-LLMs using academic VQA in Figure 4.2.

5.1.2. Image-Description Pipeline

In the Image-Description RAG pipeline, three different semantic descriptions were generated for each image input using models (GPT-4o, LLaMA3.2-Vision-90B, and Qwen2-VL-7B), the evaluation results from different generations of semantic description is shown in Figure 5.2. The GPT-4o model outperformed the other two models in terms of faithfulness score when answering questions about the images based on their generated descriptions. GPT-4o achieved a faithfulness score of approximately 78.8%, while the other models scored below 63%. However, the answer relevancy was similar across all models. Notably, GPT-4o exhibited a higher context recall score, recording around 50%, compared to the other models, which scored below

20%.

5.2. Real Use Case Video Test Set

To ensure the robustness of our Video-RAG pipeline, real-world cases were used to evaluate the main components of the pipeline, thereby validating the concepts employed in this research through domain-expert test sets. Two primary cases were used for evaluation, as detailed in the following subsections.

5.2.1. Surgical Procedure Training

A faithfulness score of 78.4% was achieved by Single-Frame-RAG approach, based on a set of QA pairs from the surgical procedure training test set, as presented in the previous chapter. In contrast, the Sequential-RAG approach yielded a significantly lower score of 41.1%, as shown in Figure 5.3, at an Frames-Per-Seconds (FPS) value of 1.

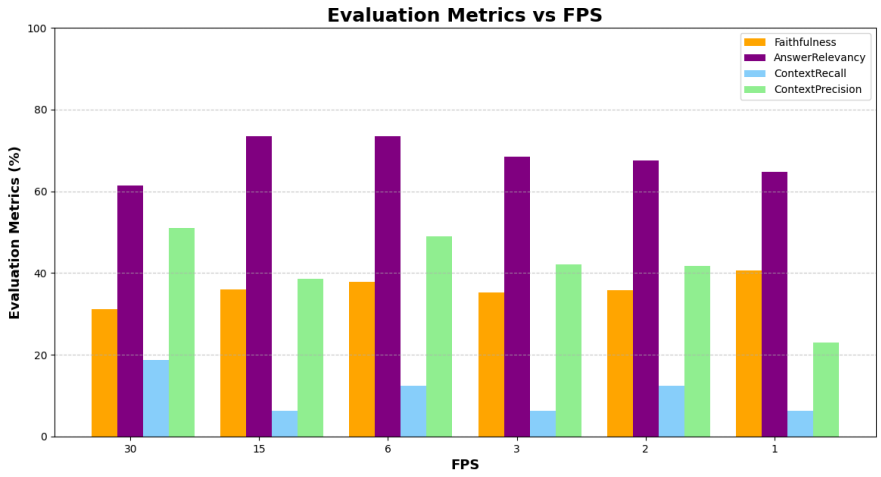


Figure 5.3.: **Sequential MM-RAG evaluation:** Different FPS values were set within the range of 1 to 30, resulting in varying metric values. GPT-4o was used with the Image-Description MM-RAG pipeline for the evaluation on the surgical procedure training test set shown previously in Figure 4.3.

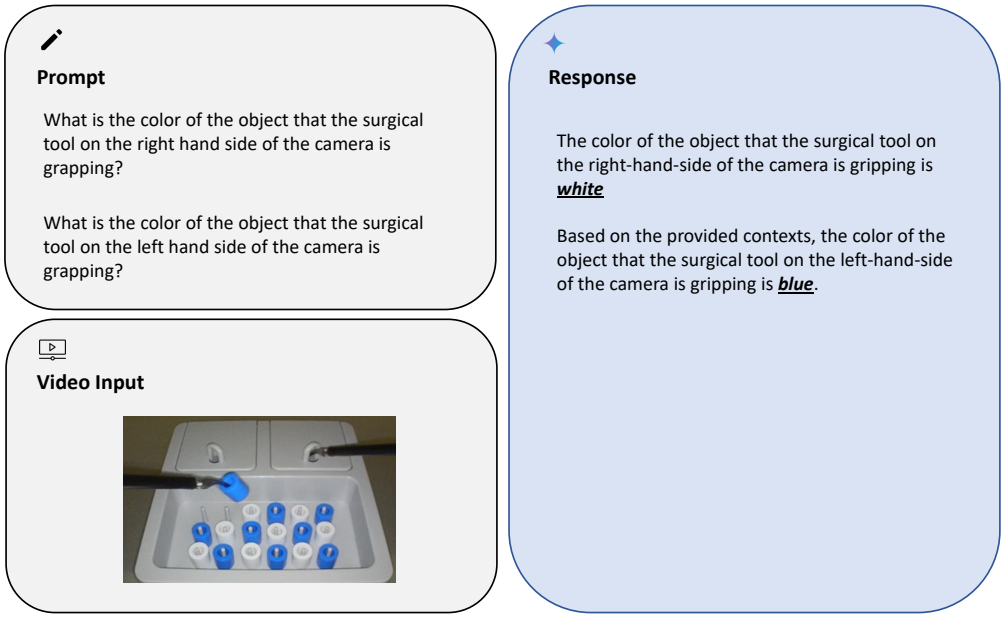


Figure 5.4.: An example prompt for 30 FPS extracted using coarse-level key-frame extraction approach (The figure illustrates a representative frame from the set of extracted key-frames of the video. It is important to note that the corresponding question is answered based on the aggregated semantic descriptions of all retrieved frames from the database).

Figure 5.4 shows the answers generated by the GPT-4o model for two input prompts based on a surgical procedure training video Section 4.2. An FPS value of 30 was used to extract coarse-level key-frames from the video input, resulting in correct answers to both queries, indicating that the right and left tools were holding different objects. The camera used for the video had a frame rate of 30 FPS, and this rate was maintained during the video capture. Overall, the 96-second video contained 2,879 frames, which required 8 hours of processing.

Figure 5.5 shows the answers generated by the GPT-4o model for two input prompts regarding a surgical procedure training video. An FPS value of 1 was used to extract coarse-level key-frames from the video input, which resulted in an-

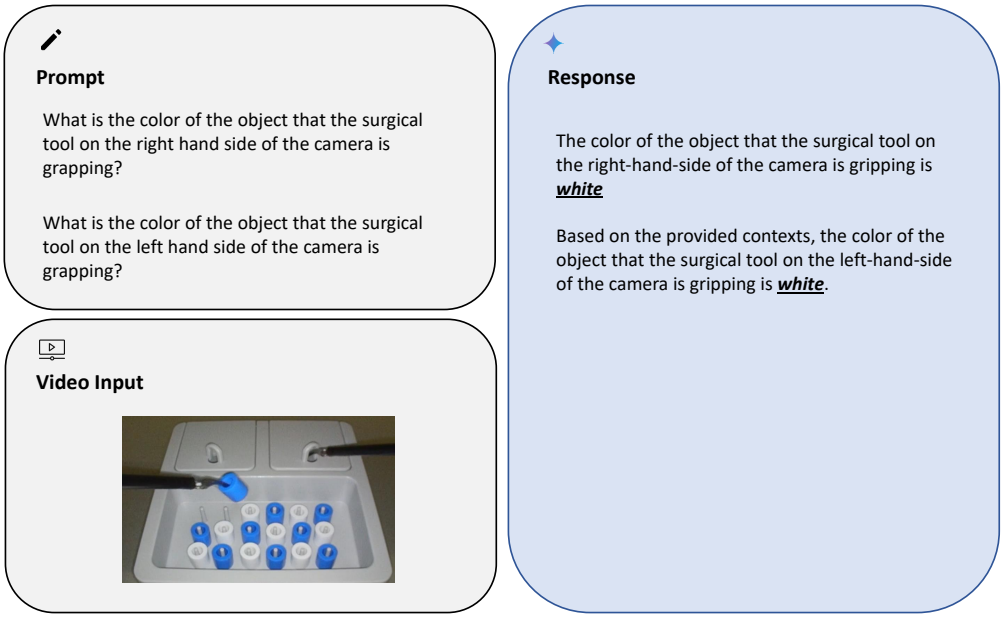


Figure 5.5.: An example prompt for 1 FPS extracted using coarse-level key-frame extraction approach (The figure illustrates a representative frame from the set of extracted key-frames of the video. It is important to note that the corresponding question is answered based on the aggregated semantic descriptions of all retrieved frames from the database).

swering one query incorrectly, as it failed to accurately reflect that the right and left tools were holding different objects. The example in the figure illustrates that extracting frames with an FPS of 1 led to suboptimal results. The camera used for the video had a frame rate of 30 FPS, but only 1 FPS was stored from the video. As a result, the 96-second video contained only 96 frames, requiring 16 minutes of processing.

Moreover, Figure 5.3 illustrates the evaluation results across various FPS settings. The results indicate that using an FPS value of 6 for key-frame extraction via the uniform sampling approach yields relatively better faithfulness and answer relevancy on the specific surgical procedure training recordings. Specifically, it attains a faith-

fulness score of approximately 38% and an answer relevancy score of around 74%. However, the context recall does not exhibit a consistent trend and is notably lower compared to using a 30 FPS setting. This evaluation was conducted on the surgical procedure training test set shown in Figure 4.3, using the coarse-level-frame extraction approach outlined in Section 3.2.1 and the Sequential-RAG pipeline in Section 3.2.4 for answer synthesis.

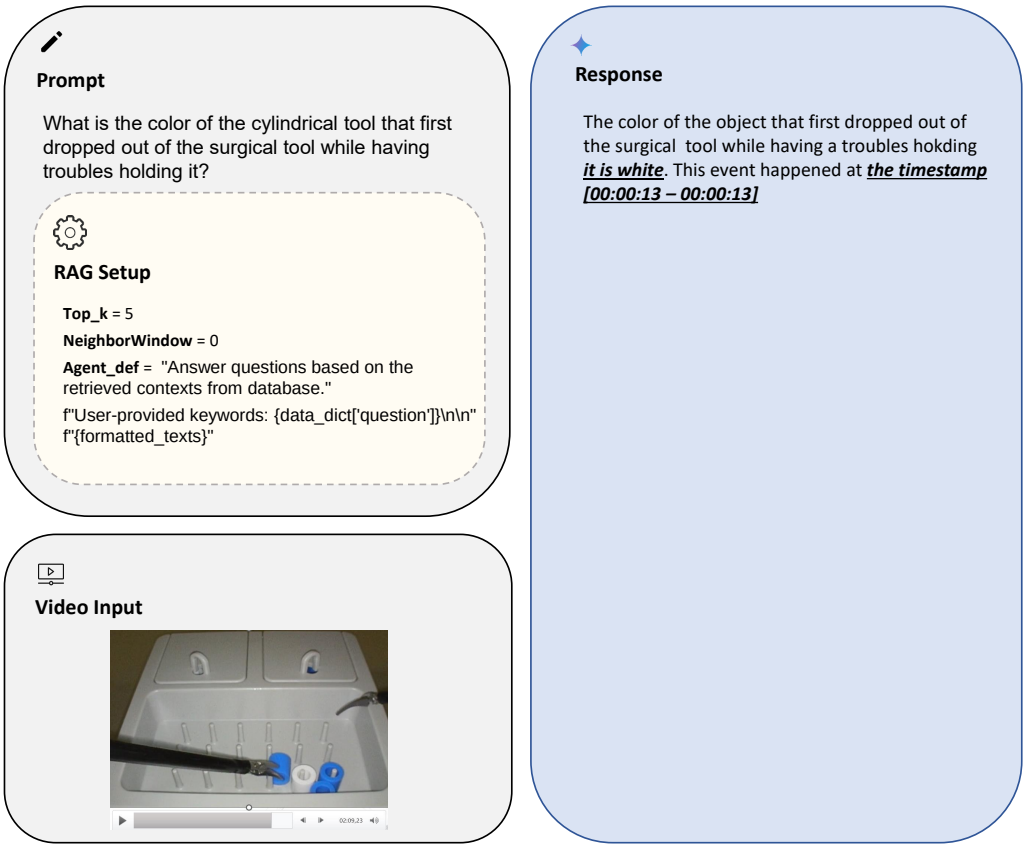


Figure 5.6.: Single-Frame MM-RAG pipeline sample result over surgical procedure training use case

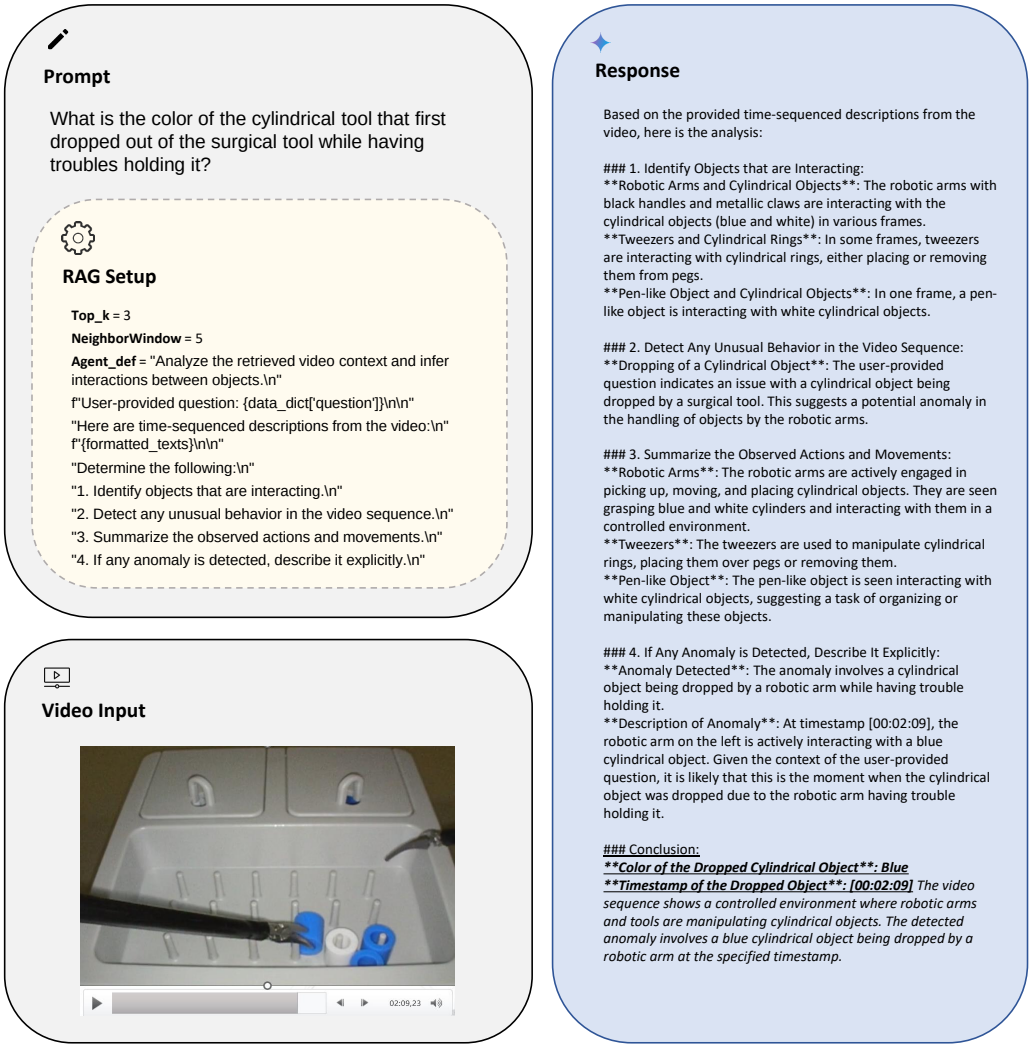


Figure 5.7.: Sequential MM-RAG pipeline sample result over surgical procedure training use case: A neighboring window size of 5 frames before and after the key-frame and the selection of the top 3 relevant frames were set, enabling the identification of the timestamps where the required behavior occurred.

As described earlier in Chapter 3, different approaches to generate frame descriptions lead to varying answers. In this context, Figure 5.7 and Figure 5.6 illustrate the differences between the Single-Frame-RAG and Sequential-RAG approaches in terms of gathering temporal information from the videos. The Sequential-RAG approach facilitates access to temporal information, as evidenced by its ability to accurately answer timestamp-related questions, an outcome not achieved by the Single-Frame-RAG method.

5.2.2. Test-Bench Hardware Troubleshooting

A faithfulness score of 58.6% was achieved using the Single-Frame-RAG technology, based on a set of test-bench hardware troubleshooting QA pairs presented in Figure 4.4 in the previous chapter. Compared to the surgical use case, this faithfulness score is approximately 20% lower.

5.3. Cost-Efficiency Analysis of Multimodal LLMs

As presented in Table 5.1, a comparative analysis between LLaMA3.2-Vision-90B and GPT-4o is provided, highlighting differences in both cost efficiency and performance. LLaMA3.2-Vision-90B can be used as a free, open-source model, incurring only minor costs related to electricity, hardware rental, and maintenance, providing answers with approximately 62.5% faithfulness when using two NVIDIA A6000 GPUs (48 GB each), totaling 55 GB of utilized GPU memory. We preferred to use the LLaMA3.2-Vision-90B model instead of GPT-4o to process the key-frames extracted from a video and generate the semantic descriptions of the frames. Despite GPT-4o achieving higher faithfulness, it costs around 60 euros to process a two-minutes video containing around 100 frames each with 864x540 resolution with a Base64-encoded representation of the image file. Therefore, LLaMA3.2-Vision-90B was utilized to process the frames and generate multimodal semantic descriptions of the key-frames, while GPT-4o was employed afterwards for simple answer synthesis.

Table 5.1.: Comparison between Multimodal LLMs for generation of the semantic image descriptions (Semantic description generation is performed as an offline process prior to runtime, enabling the preparation of information for ingestion into the vector store used by the RAG pipeline).

Value for Money	**GPT-4o**	**LLaMA3.2-Vision-90B**	**Qwen2-VL-7b**
Computational Resources	Azure OpenAI Server	Ollama Server 55 GB Utilized GPU Memory	Local Machine 16 GB Utilized GPU Memory
Inference Time	10 Secs per Frame	60 Secs per Frame	130 Secs per Frame
Licence Cost	Input: 2.4 EUR, Output: 9.6 EUR per 1M tokens	Open source	Open source
Hardware Cost	No cost	Around 67 cent/hour	Around 67 cent/hour
Electricity Cost	No cost	Around 30 cent/hour	Around 30 cent/hour
Maintainance Cost	No cost	Around 17 cent/hour	Around 17 cent/hour
Accuracy-Faithfulness	78.8%	62.5%	53%

6. Discussion

As introduced in previous chapters, the pipeline is structured around two core components: extracting representative key-frames from videos and generating semantic descriptions of those frames using multimodal large language models (MM-LLMs). These components were evaluated across two use cases to assess the faithfulness of the generated answers.

This chapter discusses the main findings from the development and evaluation of the proposed Video-RAG pipeline. Highlighting the technologies that proved most suitable for the tested application scenarios. Additionally, the discussion will address the limitations observed in each of the proposed methods, offering insight into their performance boundaries and areas for future enhancement.

6.1. Key-Frame Processing

As outlined in Section 3.1, two pipeline variants were evaluated using the academic Vision Question-Answering (VQA) test set from Hugging Face [49]. The evaluation results, illustrated in Figure 5.1 and Figure 5.2, indicate that processing image inputs via their semantic descriptions leads to improved accuracy and reduced computational complexity when compared to feeding raw images directly into the multimodal large language model (MM-LLM). This finding was validated using both GPT-4o and LLaMA3.2-Vision-90B models, for which prompt definitions were carefully optimized.

The Image-Description RAG pipeline proved to be more efficient than the Raw-Image RAG pipeline, as it requires significantly less memory and computational resources during inference. However, the primary factor contributing to this efficiency is the prompt design for the MM-LLM to generate the descriptions, which necessitates prompt optimization tailored to each specific use case. Nevertheless, this process may still result in less detailed descriptions, thereby reducing the over-

all faithfulness of the generated content. This increased efficiency is primarily attributed to the offline generation and storage of semantic descriptions for key-frames in the vector database, thereby eliminating the need to process raw images at runtime. As a result, retrieval is based on lightweight textual embeddings, which leads to a considerable improvement in runtime performance. For instance, processing a single frame using the GPT-4o model typically takes approximately 10 seconds, whereas answering a query using the pre-generated descriptions takes less than a second depending, of course, on the size of the database and the complexity of the query.

Moreover, the Image-Description pipeline outperforms the Raw-Image approach in answer quality, as the semantic descriptions are generated using carefully optimized prompts designed to extract rich, informative content from the LLM. In contrast, the Raw-Image pipeline relies solely on the user query without prompt optimization, which limits the depth and relevance of generated responses. In addition, raw image processing is performed using the Vision Transformer model (ViT-g-14), which has been observed to have limitations in accurately capturing all visual content within the images. This limitation may not apply when using GPT-4o, as it is assumed to utilize a different visual embedding mechanism.

Given these advantages in both performance and answer faithfulness, the Image-Description input method was selected as the foundational knowledge base for the Video-RAG pipeline developed in this research.

6.2. MM-LLMs Performance

As detailed in Chapter 3, both GPT-4o and LLaMA3.2-Vision-90B were employed to generate semantic descriptions for extracted key-frames, with the goal of evaluating the quality and completeness of the generated outputs. The results demonstrated that GPT-4o consistently produced more comprehensive and accurate descriptions, effectively capturing salient visual information even when using simple prompts. However, it is important to note that this assessment was conducted by a single individual, which may introduce subjectivity and uncertainty, as the results could vary with evaluations performed by different human assessors. In contrast, LLaMA3.2-Vision-90B, while less detailed, offered a practical alternative due to its open-source nature and the absence of processing costs beyond computational resources such as

electricity, hardware rental, and maintenance.

Despite GPT-4o's superior performance, the significantly higher cost of processing individual frames made it less suitable for large-scale offline generation. Consequently, a strategic decision was made to utilize LLaMA3.2-Vision-90B optimized with refined prompts, as illustrated in Figure 3.9 and Figure 3.8 for offline semantic description generation. Meanwhile, GPT-4o was reserved for runtime answer synthesis due to its strong reasoning capabilities and efficiency during interactive inference. Nevertheless, both models still exhibit limitations in the accuracy of the generated responses, which are influenced by the resolution of the input image and the complexity of segment relationships within the image. This limitation may potentially be addressed by future multimodal large language models (MM-LLMs), such as Qwen2.5-VL-72B, particularly when deployed on a powerful GPU machine.

Furthermore, to ensure robustness, identical requests were submitted multiple times, confirming consistency in the generated descriptions and maintaining the accuracy and reliability of the Video-RAG pipeline's responses.

6.3. Semantic Description

During the comparison of semantic descriptions generated by GPT-4o and LLaMA3.2-Vision-90B models, instances of hallucination were observed. As illustrated in Figure 3.5, both models exhibited inaccuracies in basic visual recognition tasks. Specifically, they failed to correctly count the number of objects or identify the layout of the grid in a surgical instrument setup. For example, the models erroneously described a 3x6 grid containing 18 cylindrical objects as a 4x4 grid with 16 objects.

The underlying cause of this discrepancy remains unclear and is currently under investigation. Future work should explore the integration of Convolutional Neural Network (CNN) and Vision Transformer (ViT) based attention mechanisms to enhance the models' ability to accurately perceive spatial structures and object counts in complex visual scenes.

Additionally, the resulting descriptions required prompt optimization experiments to ensure that the input frame details were fully captured and that the generated descriptions covered all segments of the image. This optimization process was tested using the examples shown in Figure 3.6 and Figure 3.5. However, these examples may not be complex enough to determine whether this approach would be effective

for more detail-intensive scenarios such as analyzing a classroom video where raw images might be necessary alongside descriptions to extract fine-grained information.

6.4. Key-Frames Extraction

The impact of different frame rates on key-frame selection was analyzed using the surgical procedure training videos. Figure 5.5 and Figure 5.4 illustrate the results obtained when applying low and high Frames-Per-Second (FPS) values, respectively. It was observed that higher FPS values, such as 30 FPS, led to more accurate answer generation. For instance, when asked a simple question—such as which surgical tool is holding which colored object—the model correctly identified that the left-hand tool was always grasping a blue object while the right-hand tool was always grasping a white one. However, this level of faithfulness deteriorated when lower FPS values were used, as evidenced in the figures. Furthermore, this configuration may not be suitable for different use cases where task characteristics such as dynamics, the number of interactions, and the recording speed must be considered when selecting an appropriate FPS value.

Overall, higher FPS values is assumed to enable the pipeline to retrieve fine-grained information, especially in high-dynamics videos. However, this also increases the processing time for extracting key-frame descriptions and leads to higher redundancy in low-dynamics videos.

To systematically evaluate this observation, an experiment was conducted to assess the influence of varying FPS values on answer faithfulness within the surgical use case. As shown in Figure 5.3, an FPS value of 6 was found to offer a trade-off between descriptive accuracy and processing efficiency for generating semantic key-frame descriptions. However, this evaluation was conducted on only two video samples, which limits the reliability and generalizability of the results. Incorporating a larger and more diverse set of videos is necessary to strengthen the validity of these findings.

6.5. Temporal Information

To evaluate the capability of the implemented Video-RAG pipeline in capturing temporal information from video inputs, two strategies were investigated, as discussed in

Section 3.2.4. The results indicate that the Single-Frame-RAG approach performs effectively for static videos that lack sequential actions or significant interactions between objects, where temporal information is minimal or absent. However, this method exhibits limitations in dynamic scenarios, prompting the investigation of a Sequential-RAG pipeline to address temporal reasoning challenges.

As shown in Figure 5.6 and Figure 5.7, the Sequential-RAG approach demonstrated an improved ability to answer questions related to specific timestamps and actions occurring throughout the video capabilities not effectively supported by the Single-Frame-RAG approach. Nevertheless, this evaluation was conducted on a relatively small test set comprising fewer than 50 question–answer pairs, and further validation on a larger and more complex video dataset is required to determine the pipeline's generalizability and robustness in temporal information extraction.

Although the Sequential-RAG approach enables access to temporal information, it reduces the overall accuracy of the RAG pipeline when answering non-temporal questions. This is primarily due to the increased retrieval of redundant information from neighboring frames around the top relevant frame corresponding to the user's question. Such redundancy can weaken the pipeline's performance by overwhelming the LLM with extraneous context, thereby diminishing metrics such as faithfulness and answer relevancy. This issue can be mitigated by carefully selecting an appropriate FPS value based on the specific characteristics of the use case, particularly the frequency and nature of interactions within the video.

6.6. Real Use Case Videos

As outlined in Section 5.2, the faithfulness score for the test-bench hardware troubleshooting videos was approximately 20% lower than that observed in the surgical procedure training use case. This discrepancy is primarily attributed to the inclusion of expert-level questions within the hardware QA set, which were designed to test foundational knowledge expected in typical test-bench environments. The inclusion of such questions was intended to yield a more comprehensive and robust evaluation.

Despite achieving a relatively higher faithfulness score of 78.4% in the surgical use case, the result remains inadequate for medical applications, where a minimum faithfulness of 95% is generally required to ensure safety and reliability. This highlights the need for further research, particularly through the collection of a broader

dataset encompassing diverse surgical procedures and an expanded set of QA pairs curated by domain experts. Such improvements are essential for building a more reliable and secure Video-RAG pipeline suitable for deployment in medical contexts.

In contrast, the faithfulness score for the hardware troubleshooting use case was deemed acceptable. In industrial settings, achieving extremely high accuracy is often less critical, with a threshold around 70% generally deemed sufficient. Instead, greater emphasis is placed on minimizing resource consumption and computation time. However, to ensure the robustness of the evaluation, additional test-bench videos covering a wider range of tasks and configurations are required. It is recommended that the test set include at least 100 diverse QA pairs to validate the generalizability and effectiveness of the pipeline in industrial scenarios. Furthermore, additional video recordings from diverse sites are required to enhance the reliability and generalizability of the findings.

Moreover, the faithfulness score for the surgical procedure training use case decreased from 78.4% in the Single-Frame-RAG pipeline to 41.1% in the Sequential-RAG pipeline, primarily due to the lack of temporal QA pairs. Additionally, this type of question requires highly precise answers, where even a one-second difference can be critical—an aspect not adequately handled by the implemented pipeline. The Sequential-RAG pipeline still requires a thorough evaluation using a properly constructed test set to validate its efficiency. In contrast, the evaluation of non-temporal information was based on 15 QA pairs, with only 2 questions specifically targeting distinct actions within the videos. This contributed to the higher faithfulness score observed with the Single-Frame-RAG pipeline.

7. Conclusion

In summary, the Video-RAG pipeline was primarily built around two key components: extracting non-redundant key-frames from the video inputs and generating efficient key-frame descriptions at no cost (exept for the electricity, hardware rental, and maintenance costs) while maintaining the accuracy and robustness of the pipeline. These two components still require further research to efficiently process longer-duration and more dynamic videos, where selecting key-frames and generating their descriptions is critical.

For key-frame extraction, a uniform sampling approach was identified as the most suitable based on its balance between efficiency and accuracy. Testing across different use cases revealed that a frame rate of 6 FPS was assumed to perform optimally in moderate dynamic scenarios, offering sufficient visual context without incurring unnecessary processing overhead.

For semantic description generation, LLaMA3.2-Vision-90B was selected for offline key-frame annotation due to its minor processing cost and acceptable performance (around 62% faithfulness score). For higher-fidelity use cases, such as those requiring preservation of fine-grained visual details, GPT-4o was employed, offering better answer faithfulness at a significant computational cost—up to 60 euros for processing a two-minute video at standard resolution.

To handle temporal reasoning, the Sequential-RAG pipeline was developed to address the limitations of the Single-Frame-RAG approach. While the Single-Frame method performed adequately for static videos, it lacked the capability to reason over sequences of actions. The Sequential-RAG model demonstrated improved ability to answer time-sensitive questions, though its evaluation was limited to a small test set, necessitating further validation on a larger, more complex video dataset.

In model selection, a trade-off between cost and performance led to the hybrid use of LLaMA3.2-Vision-90B for offline description generation and GPT-4o for runtime answer synthesis. As shown in Table 5.1, this combination balanced resource

constraints with the need for accurate semantic understanding.

To validate the ability of MM-LLMs to interpret contextual and visual information from video inputs and accurately answer user queries about content or interactions, two real-world use cases were employed as proof of concept, demonstrating the feasibility of deploying MM-LLMs in both medical and industrial contexts.

As discussed in the previous chapter, the implemented pipeline still has limitations that will be addressed in future research.

8. Future Work

As discussed in the previous chapter, this research still has certain limitations that should be addressed in future work. In the current MM-RAG pipeline, there is potential for improvement by integrating raw images alongside their semantic descriptions into the database before passing them to the MM-LLM. This approach may enhance answer synthesis. The ViT-g-14 model used in this study demonstrated limitations in detecting precise spatial details, particularly in low-resolution images. Future work can investigate more advanced vision transformers with improved spatial awareness.

Additionally, the current method of redundant frame removal conducted after embedding the key-frames into a vector database is computationally intensive in terms of both time and memory. A more efficient approach would involve eliminating redundancy prior to the embedding step by leveraging segment tracking across consecutive frames. Object tracking models such as YOLO could be utilized to identify and filter out static or repetitive content, thereby optimizing both memory usage and computational efficiency.

During testing of GPT-4o and LLaMA3.2-Vision-90B, instances of hallucinated outputs were observed. To address this, future experiments should explore architectures such as CNN-ViT attention mechanisms to assess their ability to generate more accurate semantic descriptions of visual input. Given the rapid development in the multimodal LLM field, more powerful and potentially open-source models are expected to emerge. One promising candidate is Qwen2.5-VL-72B, which is on the roadmap for future evaluation. However, due to its substantial hardware requirements (254 GB GPU memory), experimentation with this model is deferred.

As established in previous chapters, key-frame extraction remains a critical component of the Video-RAG pipeline. Further research should explore advanced methods for identifying the most descriptive frames, such as clustering-based approaches, motion analysis, and scene change detection techniques.

Lastly, to enable the Video-RAG pipeline to accurately access the temporal information within videos, a technique known as three-level description is employed. This method generates three layers of semantic descriptions over the key-frames—both individually and in combination. As a core element of the pipeline, this approach holds significant potential and should be a primary focus of future research.

A. Appendix

A.1. Software Dependencies

Below is the list of software dependencies used:

```
numpy==1.26.4
pandas==1.4.2
pillow==11.0.0
datasets==3.1.0
ipykernel==6.29.5
jupyterlab==4.3.2
langchain==0.3.9
langchain-chroma==0.1.4
langchain-community==0.3.9
langchain-core==0.3.21
langchain-experimental==0.3.3
langchain-openai==0.2.11
langchain-text-splitters==0.3.2
chromadb==0.5.23
matplotlib==3.9.2
open-clip-torch==2.29.0
openai==1.57.0
opencv-python==4.10.0.84
openpyxl==3.1.5
python-dotenv==1.0.1
schedule==1.2.2
scikit-learn==1.5.2
torch==2.5.1
torchvision==0.20.1
```

```
ragas==0.2.6
unstructured==0.16.11
pymupdf==1.25.1
pypdf==5.1.0
ollama==0.4.5
qwen-vl-utils==0.0.10
```

A.2. Hardware Resources Cost Calculations

The following calculations are based on assumptions derived from the company's previous hardware invoices.

Hardware Rental Cost

The total cost of GPU hardware over 5 years is:

$$\text{Total Cost} = 20\,000\,EUR \tag{A.1}$$

The annual cost is:

$$\text{Annual Cost} = \frac{20\,000\,EUR}{5} = 4\,000\,EUR \tag{A.2}$$

The monthly cost is then:

$$\text{Monthly Cost} = \frac{4\,000\,EUR}{12} \approx 333.33\,EUR \tag{A.3}$$

However, approximated here as:

$$\text{Monthly Cost (adjusted)} \approx 380\,EUR \tag{A.4}$$

Working time is calculated as:

$$\text{Working Time} = 24\,\text{hours/day} \times 30\,\text{days} = 720\,\text{hours} \tag{A.5}$$

Considering an 80% utilization rate:

$$\text{Utilized Hours} = 0.8 \times 720 = 576\,\text{hours} \tag{A.6}$$

Thus, the hardware rental cost per hour is:

$$\text{Hardware Cost per Hour} = \frac{380\,EUR}{576\,\text{hours}} \approx 0.67\,EUR/\text{hour} \tag{A.7}$$

Electricity Cost

The power requirements are:

- GPU power consumption: 350 W
- Full workstation power consumption: 1 kW

Electricity cost is assumed to be 0.30 EUR per kWh, thus:

$$\text{Electricity Cost per Hour} = 1\,\text{kW} \times 0.30\,EUR/\text{kWh} = 0.30\,EUR/\text{hour} \tag{A.8}$$

Maintenance Cost

The cost for two Nvidia A6000 GPUs is:

$$\text{Total GPU Cost} = 2 \times 7\,000\,EUR = 14\,000\,EUR \tag{A.9}$$

The maintenance labor cost is:

$$\text{Maintenance Labor Cost} = 100\,EUR/\text{hour} \tag{A.10}$$

Assuming maintenance distributed evenly across working hours, the maintenance cost per device per hour is:

$$\text{Maintenance Cost per Device per Hour} \approx 0.17\,EUR/\text{hour} \tag{A.11}$$

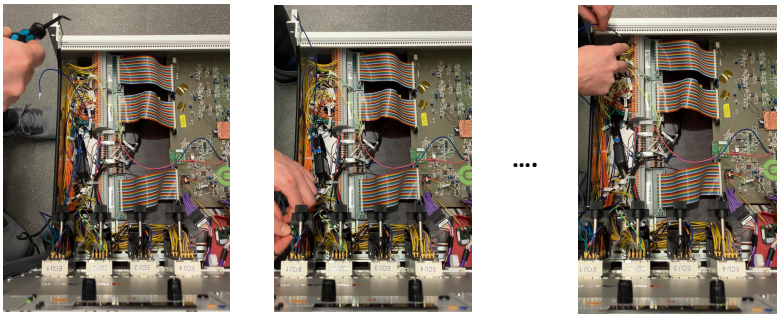
A.3. Audio Transcript Result

Below is a sample result from the Whisper model, which generated an audio transcript with timestamps for the audio content branch in the Video-RAG pipeline,

using hardware troubleshooting videos. It is important to note that this study was not the primary focus of the work.

Whisper generated audio transcript for example key-frames in hardware troubleshooting case

Video



Audio Transcript

[00:00 - 00:01] We now use this screwdriver to unlock the pin at IF9200 and connect it to the wire which is coming from ECU1D17.
[00:00 - 00:25] So we check again. This is the correct wire. So we go now to pin 16. Unlock it with this screwdriver.
[00:00 - 00:40] Therefore we press the orange pin here and inject the wire.
[00:00 - 00:52] When we pull the cable, the wire, it should be tight in the connection and not loosen.
[00:00 - 01:12] Now we recognize in the file that there is a second connection.
[00:00 - 01:18] From ECU1B15 to IF9215.
[00:00 - 01:27] So we search for the second cable at ECU1B15.
[00:00 - 01:42] Which is this one.
[00:00 - 01:48] We also see that this is not connected.
[00:00 - 01:51] And this needs to be connected to IF9215.
[00:00 - 02:00] Which is located here.
[00:00 - 02:02] There is this 15.
[00:00 - 02:08] We press the orange pin with the screwdriver.
[00:00 - 02:14] And inject the wire.
[00:00 - 02:20] Now it's down.
[00:00 - 02:22] You can see it is in plug with a ground cable.

Figure A.1.: The figure illustrates the generated audio transcript, highlighting information that was not explicitly discernible from the visual content of the videos.

List of Figures

2.1. The Transformer model serves as the foundational architecture for all large language models. It employs stacked self-attention mechanisms and point-wise fully connected layers in both the encoder and decoder, as illustrated in the left and right halves of the model architecture. Graph is referenced from [58]. 6

2.2. Embedding Vector Space: A common example illustrating the properties of semantic embeddings is that the vector difference between "man" and "woman" is approximately equivalent to that between "king" and "queen." Similarly, semantically related word pairs such as "dog" and "puppy" tend to have closely aligned vector representations, reflecting their conceptual similarity. 9

2.3. Vision Transformer Archeticture : split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the ViT is taken from [12]. 11

2.4. Qwen-VL model Archeticture: Qwen2-VL is capable of accurately identifying and comprehending the content within images, regardless of their clarity, resolution, or extreme aspect ratios. Graph is taken from [62] 15

2.5. Text-RAG Baseline Architecture [18] 17

2.6. Video-RAG architecture: Illustration of different video understanding approaches alongside Video-RAG. Video-RAG provides a resource-efficient, training-free pipeline that is easily compatible with any LVLM. By leveraging RAG, it retrieves auxiliary texts for input, leading to notable performance enhancement. The graph is referenced from [35]. 20

3.1. **Image-RAG pipelines:** A multi-vector retriever framework that processes documents by extracting images, tables, and text for embedding. Two approaches are shown: multimodal embedding and text embedding. In the multimodal path, table and text summaries are generated, enabling retrieval of raw images and associated data for processing by a multimodal LLM to generate answers. The text embedding path instead generates and retrieves summaries (including of images), which are then used by a standard LLM for answer generation. This setup highlights trade-offs between raw data usage and summarized input for efficient retrieval and reasoning. 26

3.2. **Video-RAG Pipeline:** (a) A video question-answering pipeline that integrates both visual and audio contents. The image retrieval branch extracts key-frames from the video, generates corresponding descriptions using a multimodal large language model (LLM), embeds these descriptions, and subsequently removes redundant embeddings. In parallel, the text retrieval branch transcribes the video's audio into text, which is then embedded for downstream processing. A multi-vector retriever pulls relevant frame descriptions and transcripts from a vector database based on a user query. Finally, a multimodal LLM synthesizes the retrieved information to generate a comprehensive answer. For an intuition regarding the audio content, which was not the primary focus of the study, Figure A.1 in the appendix presents an example. (b) A closer look at the main components of the pipeline, organized according to the visual processing phases of the video. Key-frames selection cycle starting from extracting the unique frames from the video input, storing the processed descriptions, synthesing answers about the input video. Showing different methodologies for each subcomponent of the pipeline. 27

3.3. The diagram illustrates the context-driven key-frames extraction pipeline using CLIP-ViT model, where coarse-level key-frames are extracted from a video and encoded alongside a user question for similarity search. This process identifies context-driven key-frames most relevant to the query for accurate video understanding. 29

3.4. GPT-4o generated semantic description using a baseline prompt applied to an example key-frame from the surgical procedure training video input. This example case will be introduced in detail in Section 4.2. 31

3.5. GPT-4o generated semantic description using an optimized prompt applied to an example key-frame from the surgical procedure training video input. 33

3.6. GPT-4o generated semantic description using an optimized prompt applied to an example key-frame from the hardware (HW) troubleshooting video set. 34

3.7. LLaMA3.2-Vision-90B generated semantic description using an optimized prompt applied to an example key-frame from the HW troubleshooting video set. This example case will be introduced in detail in Section 4.2. 35

3.8. LLaMA3.2-Vision-90B generated semantic description using a more optimized prompt applied to an example key-frame from the HW troubleshooting video set. In this case, the prompt is more detailed, with information clustered to guide the generation of a description within a specific layout, thereby preserving richer information. . . . 36

3.9. LLaMA3.2-Vision-90B generated semantic description using a more optimized prompt applied to an example key-frame from the surgical procedure training video set. LLaMA3.2-Vision-90B generated semantic description with similar structure as in Figure 3.8 for the HW troubleshooting case. 37

3.10. Single-Frame Video-RAG Pipeline 38

3.11. Sequential-Frame Video-RAG Pipeline 40

3.12. Context Retrieval of Sequential-Frame in Video-RAG 40

4.1. VQA test set sample for Raw-Image MM-RAG pipeline Evaluation . 42

4.2. VQA test set sample for Image-Description MM-RAG pipeline Evaluation 42

4.3. VQA test set sample of surgical procedure training use case for Video-RAG evaluation 44

4.4. VQA test set sample of test-bench hardware troubleshooting use case for Video-RAG evaluation 45

5.1. VQA evaluation metrics over Raw-Image MM-RAG pipeline with different MM-LLMs using academic VQA test set in Figure 4.1. 47

5.2. VQA evaluation metrics over Image-Description MM-RAG pipeline with different MM-LLMs using academic VQA in Figure 4.2. 48

5.3. **Sequential MM-RAG evaluation:** Different FPS values were set within the range of 1 to 30, resulting in varying metric values. GPT-4o was used with the Image-Description MM-RAG pipeline for the evaluation on the surgical procedure training test set shown previously in Figure 4.3. 49
5.4. An example prompt for 30 FPS extracted using coarse-level key-frame extraction approach (The figure illustrates a representative frame from the set of extracted key-frames of the video. It is important to note that the corresponding question is answered based on the aggregated semantic descriptions of all retrieved frames from the database). 50
5.5. An example prompt for 1 FPS extracted using coarse-level key-frame extraction approach (The figure illustrates a representative frame from the set of extracted key-frames of the video. It is important to note that the corresponding question is answered based on the aggregated semantic descriptions of all retrieved frames from the database). 51
5.6. Single-Frame MM-RAG pipeline sample result over surgical procedure training use case 52
5.7. Sequential MM-RAG pipeline sample result over surgical procedure training use case: A neighboring window size of 5 frames before and after the key-frame and the selection of the top 3 relevant frames were set, enabling the identification of the timestamps where the required behavior occurred. 53

List of Tables

4.1. Domain-Experts Test Sets Statistics 43

5.1. Comparison between Multimodal LLMs for generation of the semantic image descriptions (Semantic description generation is performed as an offline process prior to runtime, enabling the preparation of information for ingestion into the vector store used by the RAG pipeline). 55

Bibliography

[1] M. M. Abootorabi et al., "Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation," *arXiv preprint arXiv:2502.08826*, 2025.

[2] J. Achiam et al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] I. M. Alabdulmohsin, X. Zhai, A. Kolesnikov, and L. Beyer, "Getting vit in shape: Scaling laws for compute-optimal model design," *Advances in Neural Information Processing Systems*, vol. 36, pp. 16 406–16 425, 2023.

[4] X. Amatriain, "Prompt design and engineering: Introduction and advanced methods," *arXiv preprint arXiv:2401.14423*, 2024.

[5] A. Andreyev, "Quantization for openai's whisper models: A comparative analysis," *arXiv preprint arXiv:2503.09905*, 2025.

[6] R. Anil et al., "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[7] F. Bordes et al., "An introduction to vision-language modeling," *arXiv preprint arXiv:2405.17247*, 2024.

[8] Y. Chang et al., "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

[9] L.-C. Chen, M. S. Pardeshi, Y.-X. Liao, and K.-C. Pai, "Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model," *Computer Standards & Interfaces*, vol. 94, p. 103 995, 2025.

[10] C. Cui et al., "A survey on multimodal large language models for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 958–979.

[11] M. Dehghani et al., "Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution," *Advances in Neural Information Processing Systems*, vol. 36, pp. 2252–2274, 2023.

[12] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] M. Eibich, S. Nagpal, and A. Fred-Ojala, "Aragog: Advanced rag output grading," *arXiv preprint arXiv:2404.01037*, 2024.

[14] S. Es, J. James, L. E. Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2024, pp. 150–158.

[15] ExplodingGradients, *Ragas documentation*, Accessed: April 7, 2025, 2025. [Online]. Available: `https://docs.ragas.io/en/stable/`.

[16] L. Galke and A. Scherp, "Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide mlp," *arXiv preprint arXiv:2109.03777*, 2021.

[17] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 1762–1777.

[18] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, 2023.

[19] A. Gendia, "Cloud based ai-driven video analytics (cavs) in laparoscopic surgery: A step closer to a virtual portfolio," *Cureus*, vol. 14, no. 9, 2022.

[20] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[21] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[22] S. Hofstätter, J. Chen, K. Raman, and H. Zamani, "Fid-light: Efficient and effective retrieval-augmented text generation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1437–1447.

[23] S. Jeong, K. Kim, J. Baek, and S. J. Hwang, "Videorag: Retrieval-augmented generation over video corpus," *arXiv preprint arXiv:2501.05874*, 2025.

[24] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 696–15 707.

[25] M. A. Khaliq, P. Chang, M. Ma, B. Pflugfelder, and F. Miletić, "Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models," *arXiv preprint arXiv:2404.12065*, 2024.

[26] Langchain, *Query transformations*, `https://blog.langchain.dev/query-transformations/`, Accessed: 2024-03-23, 2023.

[27] B. Li et al., "Llava-onevision: Easy visual task transfer," *arXiv:2408.03326*, 2024.

[28] F. Li et al., "Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models," *arXiv preprint arXiv:2407.07895*, 2024.

[29] M. Li, X. Li, Y. Chen, W. Xuan, and W. Zhang, "Unraveling and mitigating retriever inconsistencies in retrieval-augmented large language models," *arXiv preprint arXiv:2405.20680*, 2024.

[30] Z. Li, L. Gu, W. Wang, R. Nakamura, and Y. Sato, "Surgical skill assessment via video semantic aggregation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2022, pp. 410–420.

[31] Z. Li, X. Wu, H. Du, F. Liu, H. Nghiem, and G. Shi, "A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges,"

[32] B. Lin et al., "Video-llava: Learning united visual representation by alignment before projection," *arXiv preprint arXiv:2311.10122*, 2023.

[33] Y. Liu et al., "Summary of chatgpt-related research and perspective towards the future of large language models," *Meta-radiology*, vol. 1, no. 2, p. 100 017, 2023.

[34] LiveXiv Contributors, *LiveXiv Dataset*, `https://huggingface.co/datasets/LiveXiv/LiveXiv`, Accessed: April 19, 2025, 2024.

[35] Y. Luo et al., "Video-rag: Visually-aligned retrieval-augmented long video comprehension," *arXiv preprint arXiv:2411.13093*, 2024.

[36] Z.-A. Ma, T. Lan, R.-C. Tu, Y. Hu, H. Huang, and X.-L. Mao, "Multi-modal retrieval augmented multi-modal generation: A benchmark, evaluate metrics and strong baselines," *arXiv preprint arXiv:2411.16365*, 2024.

[37] McKinsey Global Institute, *The economic potential of generative ai: The next productivity frontier*, Accessed: April 7, 2025, 2023. [Online]. Available: `https://www.mckinsey.de/~/media/mckinsey/locations/europe%20and%20middle%20east/deutschland/news/presse/2023/2023-06-14%20mgi%20genai%20report%2023/the-economic-potential-of-generative-ai-the-next-productivity-frontier-vf.pdf`.

[38] A. Nagy, Y. Spyridis, and V. Argyriou, "Cross-format retrieval-augmented generation in xr with llms for context-aware maintenance assistance," in *2025 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, IEEE, 2025, pp. 355–361.

[39] Ollama, *Llama 3.2 vision*, Accessed: April 7, 2025, 2024. [Online]. Available: `https://ollama.com/library/llama3.2-vision`.

[40] J. Pereira, R. Fidalgo, R. Lotufo, and R. Nogueira, "Visconde: Multi-document qa with gpt-3 and neural reranking," in *European Conference on Information Retrieval*, Springer, 2023, pp. 534–543.

[41] R. Qu, R. Tu, and F. Bao, "Is semantic chunking worth the computational cost?" *arXiv preprint arXiv:2410.13070*, 2024.

[42] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*, PMLR, 2023, pp. 28 492–28 518.

[43] A. Radford et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[44] RAGAS, *Available metrics - ragas documentation*, `https://docs.ragas.io/en/latest/concepts/metrics/available_metrics/`, Accessed: 2025-05-11.

[45] A. Ramesh et al., "Zero-shot text-to-image generation," in *International conference on machine learning*, PMLR, 2021, pp. 8821–8831.

[46] M. Riedler and S. Langer, "Beyond text: Optimizing rag with multimodal inputs for industrial applications," *arXiv preprint arXiv:2410.21943*, 2024.

[47] K. D. Rosa, "Video enriched retrieval augmented generation using aligned video captions," *arXiv preprint arXiv:2405.17706*, 2024.

[48] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," *arXiv preprint arXiv:1912.05911*, 2019.

[49] N. Shabtay et al., "Livexiv–a multi-modal live benchmark based on arxiv papers content," *arXiv preprint arXiv:2410.10783*, 2024.

[50] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132 306, 2020.

[51] M. F. Shojaei et al., "Ai-university: An llm-based platform for instructional alignment to scientific classrooms," *arXiv preprint arXiv:2504.08846*, 2025.

[52] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, "Alfworld: Aligning text and embodied environments for interactive learning," *arXiv preprint arXiv:2010.03768*, 2020.

[53] C. Si et al., "Prompting gpt-3 to be reliable," *arXiv preprint arXiv:2210.09150*, 2022.

[54] M. Suri, P. Mathur, F. Dernoncourt, K. Goswami, R. A. Rossi, and D. Manocha, "Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation," *arXiv preprint arXiv:2412.10704*, 2024.

[55] G. Team et al., "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[56] D. Tomkou et al., "Bridging industrial expertise and xr with llm-powered conversational agents," *arXiv preprint arXiv:2504.05527*, 2025.

[57] H. Touvron et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[58] A. Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[59] S. Vatsal and H. Dubey, "A survey of prompt engineering methods in large language models for different nlp tasks," *arXiv preprint arXiv:2407.12994*, 2024.

[60] J. Wang, H. Tang, T. Kantor, T. Soltani, V. Popov, and X. Wang, "Surgment: Segmentation-enabled semantic search and creation of visual question and feedback to support video-based surgery learning," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–18.

[61] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving text embeddings with large language models," *arXiv preprint arXiv:2401.00368*, 2023.

[62] P. Wang et al., "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.

[63] P. Xia et al., "Mmed-rag: Versatile multimodal rag system for medical vision language models," *arXiv preprint arXiv:2410.13085*, 2024.

[64] Z. Yang et al., "The dawn of lmms: Preliminary explorations with gpt-4v (ision)," *arXiv preprint arXiv:2309.17421*, vol. 9, no. 1, p. 1, 2023.

[65] Z. Yin, "Understand functionality and dimensionality of vector embeddings: The distributional hypothesis, the pairwise inner product loss and its bias-variance trade-off," *arXiv preprint arXiv:1803.00502*, 2018.

[66] Z. Yin and Y. Shen, "On the dimensionality of word embedding," *Advances in neural information processing systems*, vol. 31, 2018.

[67] K. Yuan, N. Navab, N. Padoy, et al., "Procedure-aware surgical video-language pretraining with hierarchical knowledge augmentation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 122 952–122 983, 2024.

[68] D. Zhang et al., "Mm-llms: Recent advances in multimodal large language models," *arXiv preprint arXiv:2401.13601*, 2024.

[69] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[70] Y. Zhang et al., "Video instruction tuning with synthetic data," *arXiv preprint arXiv:2410.02713*, 2024.

[71] R. Zhao et al., "Retrieving multimodal information for augmented generation: A survey," *arXiv preprint arXiv:2303.10868*, 2023.

[72] W. X. Zhao et al., "A survey of large language models," *arXiv:2303.18223*, vol. 1, no. 2, 2023.