A project report on

LEVERAGING LARGE LANGUAGE MODELS FOR AUTOMATED GRADING OF ANSWER SCRIPTS IN COMPUTER SCIENCE

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

ANEESH ASHWINIKUMAR SATHE (21BCE1625)

MANU MISHRA (21BCE1639)

VARUN SAXENA(21BCE1584)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2025

LEVERAGING LARGE LANGUAGE MODELS FOR AUTOMATED GRADING OF ANSWER SCRIPTS IN COMPUTER SCIENCE

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

ANEESH ASHWINIKUMAR SATHE (21BCE1625)

MANU MISHRA (21BCE1639)

VARUN SAXENA(21BCE1584)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2025

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled **"LEVERAGING LARGE LANGUAGE MODELS FOR AUTOMATED GRADING OF ANSWER SCRIPTS IN COMPUTER SCIENCE"** submitted by **ANEESH ASHWINIKUMAR SATHE (21BCE1625)**, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai is a record of Bonafide work carried out by me under the supervision of **Dr. Janaki Meena M.**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Leveraging Large Language Models for Automated Grading of Answer Scripts in Computer Science"** is prepared and submitted by **Aneesh Ashwinikumar Sathe (21BCE1625)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. Janaki Meena M

Date:

Signature of the Examiner

Name:

Date:

Signature of the Examiner

Name:

Date:

Approved by the Head of Department,
Bachelor of Technology in Computer Science and Engineering

Name: **Dr. Nithyanandam P**

Date:

(Seal of SCOPE)

ABSTRACT

Recent advancements in multimodal artificial intelligence have paved the way for innovative solutions to complex, domain-specific tasks. This project introduces a novel framework for automatically grading and generating feedback on answer scripts in computer science subjects, leveraging a hybrid system that integrates Vision Language Models (VLMs) and Large Language Models (LLMs). Specifically, the Qwen2.5-VL series is fine-tuned to specialize in interpreting handwritten inputs, including text, diagrams, and equations, while also converting visual content such as diagrams into natural language descriptions. Google's Gemma-3 model is fine-tuned to serve as a domain expert in Data Structures and Algorithms (DSA), equipped with a robust knowledge store to assess the validity, accuracy, and depth of student answers.

By combining these technologies, the proposed framework establishes a seamless connection between textual, visual, and structured data, enabling comprehensive processing, analysis, and grading of answer scripts related to DSA. The fine-tuned Qwen2.5-VL models contribute advanced visual understanding and multilingual support, critical for interpreting handwritten content and describing diagrams in natural language. Meanwhile, the fine-tuned Gemma-3 model enhances logical consistency, in-depth analysis, and reasoning-based problem-solving, ensuring accurate evaluation of student responses against domain-specific criteria.

This integrated approach not only enhances the model's ability to handle multimodal inputs but also bridges the gap between visual comprehension, structured knowledge representation, and high-level reasoning. The framework demonstrates significant potential to transform educational tools, coding assistants, and problem-solving platforms by offering a unified solution that merges visual understanding, domain expertise, structured data processing, and advanced reasoning. Future work will focus on optimizing the fine-tuning process, extending the model's capabilities to address more complex DSA problems, and improving the integration of visual and textual modalities for real-world applications.

Keywords: Multimodal AI, Vision Language Models, Large Language Models, Automated Grading, Data Structures and Algorithms, Fine-Tuning, Handwriting OCR

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Janaki Meena M, Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Machine Learning.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor, Dr. T. Thyagarajan Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr Nityanandam P, Head of the Department, B.Tech. Computer Science and Engineering and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: **Aneesh Ashwinikumar Sathe**

CONTENTS PAGE NO

LIST OF FIGURES

PAGE NO

LIST OF TABLES **PAGE NO**

LIST OF ACRONYMS

LLM	Large Language Model
VLM	Vision Language Model
RAG	Retrieval-augmented generation
LoRA	Low Rank Adaptation
OCR	Optical Character Recognition
M-RoPE	Multimodal Rotatory Positional Encoding
DSA	Data Structures and Algorithms
JSON	JavaScript Object Notation

Chapter 1

Introduction

1.1 INTRODUCTION

The rapid evolution of multimodal artificial intelligence has significantly transformed computational problem-solving, enabling models to process and integrate information across multiple modalities. This advancement has paved the way for more sophisticated AI-driven solutions, particularly in educational settings where automated assessment and feedback mechanisms are crucial. Among these innovations, Vision-Language Models (VLMs) have demonstrated remarkable capabilities in bridging the gap between textual and visual data, allowing AI systems to interpret and respond to complex inputs such as handwritten answers, diagrams, and structured text. Simultaneously, the fine-tuning of Large Language Models (LLMs) for specialized domains has emerged as an effective strategy to enhance performance in tasks requiring deep domain expertise. This report presents a novel framework that integrates VLMs with fine-tuned LLMs to create an automated grading system proficient in evaluating answer scripts related to Data Structures and Algorithms (DSA). By combining structured knowledge representation with advanced visual understanding, this approach enhances grading accuracy and efficiency in computational assessments.

A key challenge in automating the grading process lies in accurately interpreting handwritten responses, flowcharts, pseudocode, and mathematical expressions. Traditional Optical Character Recognition (OCR) systems often struggle with the diverse handwriting styles and intricate visual elements present in students' answer scripts. However, recent advancements in VLMs, particularly the Qwen-VL series, have introduced sophisticated techniques to address these challenges.

Qwen2.5-VL-3B-Instruct is a state-of-the-art vision-language model known for its adaptability in processing multimodal inputs. Its dynamic resolution handling allows it to process high-resolution images of handwritten content without significant loss of detail, ensuring more precise interpretation. Additionally, its multimodal rotary position embedding facilitates better spatial understanding of visual elements such as diagrams and equations, which is crucial for accurately assessing DSA problems. The model’s multilingual support further enhances its applicability across diverse student populations, making it a robust solution for automated grading in computer science education. The Qwen-VL models excel in extracting and understanding visual representations of DSA problems, enabling it to analyze algorithmic diagrams, recursion trees, and code snippets embedded within answer scripts. This capability ensures that the system can accurately interpret and assess not just textual responses but also graphical content that is integral to computer science education.

While Vision-Language Models (VLMs) provide a strong foundation for processing visual data, their capabilities must be complemented by fine-tuned Large Language Models (LLMs) that specialize in evaluating the correctness, efficiency, and clarity of students' responses. In this project, Google Gemma-3-4B-Instruct, a fine-tuned LLM, is employed to assess algorithmic solutions with high precision. Trained on extensive datasets comprising structured knowledge—including JSON-based repositories of algorithmic concepts, computational complexity analyses, and model solutions for diverse Data Structures and Algorithms (DSA) problems—Gemma 3 4B Instruct ensures context-aware evaluations by identifying logical errors, inefficiencies, and deviations from standard algorithmic implementations.

The integration of Gemma 3 4B Instruct enhances the grading system by moving beyond surface-level assessments to provide detailed, structured feedback on students' submissions. For instance, if a student submits a suboptimal or incorrect implementation of Dijkstra's algorithm, the system precisely identifies errors, recommends optimizations, and suggests alternative approaches. Additionally, it evaluates code readability, adherence to best practices, and computational efficiency, offering comprehensive feedback to foster better learning outcomes.

A key contribution of this model is its ability to seamlessly integrate multimodal data, combining textual, visual, and structured information for automated grading at scale. Qwen2.5-VL-3B-Instruct processes handwritten scripts and visual elements, extracting relevant data for Gemma 3 4B Instruct, which then performs a detailed evaluation. This synergy enables the system to assess open-ended problem-solving tasks beyond predefined multiple-choice formats, enhancing students' understanding of computational concepts. Furthermore, this multimodal approach reduces manual grading effort while maintaining accuracy and fairness, allowing educators to scale assessments efficiently.

By unifying VLMs with specialized LLMs like Google Gemma, this report contributes to the development of adaptive AI-driven solutions for technical problem-solving. The proposed framework enhances efficiency in automated grading, sets the foundation for advanced multimodal AI systems, and underscores the transformative potential of AI in education, assessment, and beyond. As AI continues to evolve, such integrations will play a pivotal role in scalable, interactive, and intelligent learning methodologies.

1.2 BACKGROUND OF THE PROBLEM

Traditional manual grading systems face significant challenges in terms of efficiency, consistency, scalability, and faculty workload. These issues are particularly pronounced in technical subjects like Computer Science, where assessments often involve complex problem-solving, mathematical notations, and multiple valid solution approaches. The proposed automated grading system aims to address these challenges by leveraging advanced technologies such as computer vision, natural language processing, and machine learning. Below is a detailed breakdown of the problems this system seeks to solve:

1. Slow Grading Speed

Problem : Manual grading of handwritten answer scripts is time-consuming, especially for large cohorts in standardized exams or university courses. Faculty members must carefully evaluate each response, verify intermediate steps, and assign partial credit where applicable.

Impact : Slow grading delays feedback to students, hindering their ability to learn from mistakes in a timely manner. It also creates bottlenecks in academic workflows, particularly during high-stakes evaluations.

Traditional grading methods rely heavily on human effort, which is inherently slower when dealing with complex, multi-step problems. For example, assessing algorithms, data structures, or mathematical proofs requires meticulous attention to detail, making the process even more labor-intensive. The proposed system accelerates grading by automating text recognition and logical evaluation, enabling near-instantaneous feedback and reducing turnaround times significantly.

2. Inconsistency in Grading

Problem : Human graders may inadvertently introduce inconsistencies due to subjective interpretations, fatigue, or varying levels of expertise. This can lead to discrepancies in scoring, especially for open-ended questions or alternative solution approaches.

Impact : Inconsistent grading undermines fairness and erodes trust in the evaluation process. Students may feel disadvantaged if their responses are graded differently from peers who provided similar answers.

Even with well-defined rubrics, human graders can struggle to maintain uniformity across hundreds or thousands of submissions. Factors such as cognitive biases, time constraints, and differing interpretations of partial credit policies exacerbate the issue. By employing a standardized, AI-driven approach, the proposed system ensures consistent application of grading criteria, regardless of the volume of submissions or complexity of the questions.

3. High Faculty Workload

Problem : Faculty members spend an excessive amount of time grading, detracting from other critical responsibilities such as teaching, research, and mentorship. This workload is particularly burdensome during peak evaluation periods.

Impact : Overburdened faculty may experience burnout, reduced job satisfaction, and limited capacity to engage in professional development activities. Additionally, institutions face challenges in scaling their teaching operations without proportionally increasing staff.

In many educational settings, grading constitutes a significant portion of faculty workloads, leaving less time for innovation in teaching methodologies or personalized student support. Automating repetitive tasks like transcription, logical validation, and score allocation alleviates this burden, allowing educators to focus on higher-value activities. The proposed system reduces manual intervention while maintaining the quality and rigor of assessments.

4. Limited Scalability

Problem : Manual grading is not scalable for large-scale evaluations, such as entrance exams, MOOCs (Massive Open Online Courses), or corporate training programs. The logistical challenges of handling thousands of submissions make it impractical to rely solely on human graders.

Impact : Institutions face difficulties in expanding their reach or offering flexible learning options due to limitations in assessment infrastructure. This restricts access to education and professional development opportunities.

As the demand for scalable education grows—driven by online learning platforms and global certification programs—the need for efficient, automated grading solutions becomes paramount. The proposed system addresses this challenge by providing a scalable framework capable of processing diverse input formats and delivering accurate results at scale. Its integration with cloud-based infrastructure further enhances its adaptability to varying workloads.

1.3 LITERATURE REVIEW

Huber et al. [1] examine the transformative potential of large language models (LLMs) in education while addressing the challenges associated with their integration. The authors underscore the dual nature of LLMs as both a valuable resource and a potential risk, requiring careful consideration. A key challenge highlighted is the stochastic nature of LLM-generated text, which can introduce biases and inaccuracies. To mitigate these risks, the paper emphasizes the need for domain expertise to critically evaluate and refine LLM outputs. The authors propose a two-phase approach to address these challenges. The first phase involves exploratory strategies such as prompt engineering, enabling educators to experiment with LLMs and manage their inherent unpredictability. Prompt engineering is presented as a playful yet systematic way to harness the strengths of LLMs while minimizing their limitations. This phase lays the groundwork for balancing the risks and opportunities associated with these advanced models. The second phase focuses on leveraging LLMs within game-based learning environments. Drawing on extensive research in game-based pedagogy, the authors argue that well-designed educational games can effectively foster domain-specific knowledge, enhance motivation, and sustain student engagement. LLMs are positioned as tools for generating dynamic, gamified content tailored to diverse learning contexts. By integrating LLM capabilities into game design, educators can create interactive experiences that encourage critical thinking and the development of specialized skills. The paper envisions a collaborative future where LLMs act as partners in education, augmenting human expertise rather than replacing it. The authors stress the importance of maintaining human judgment and fostering critical evaluation to address the inherent limitations of LLMs. This partnership between human and machine intelligence aims to preserve the role of educators while enriching the learning experience. By blending the generative potential of LLMs with carefully designed pedagogical strategies, the proposed framework highlights the transformative possibilities of this collaboration. Game-based learning is presented as a key avenue for integrating LLMs, enabling educators to capitalize on their strengths while mitigating risks. The authors emphasize that this approach not only enhances the quality of educational content but also promotes the development of critical human competencies, ensuring a balanced and effective use of technology in education.

Alhafni et al. [2] examine the transformative impact of large language models (LLMs) in education, focusing on their applications in writing support, reading assistance, spoken language learning, and intelligent tutoring systems (ITS). The study highlights the advancements made by cutting-edge LLMs such as GPT-3.5, GPT-4, and LLaMA, which have significantly influenced educational technologies. Their integration into platforms like Duolingo and Grammarly underscores their role in delivering enhanced personalization, writing assistance, and interactive learning experiences, marking a paradigm shift in educational application development. The authors discuss the pedagogical advantages offered by LLM-powered tools, particularly Grammatical Error Correction (GEC), which provides learners with instant feedback and tailored insights to improve writing skills. This personalized approach fosters better engagement and understanding for students. In the domain of reading assistance, LLMs facilitate tasks such as text simplification and readability assessment, empowering learners to comprehend complex material more effectively. Emerging techniques like zero-shot learning, prompt tuning, and fine-tuning are instrumental in advancing these capabilities, allowing LLMs to adapt to varied educational needs. In spoken language learning, LLMs enable interactive and conversational learning environments, creating opportunities for real-time practice and feedback. Intelligent tutoring systems powered by LLMs are designed to offer adaptive, personalized instruction, simulating one-on-one tutoring experiences for a wide range of subjects. These innovations contribute to the scalability and accessibility of high-quality education globally.

Yan et al. [3] present a comprehensive scoping review on the applications of large language models (LLMs) in education, categorizing their use across nine key areas, including profiling, detection, grading, teaching support, and recommendation. The study identifies 53 specific use cases, showcasing the impact of advanced LLMs such as GPT-3 and GPT-4 in transforming educational practices. These models significantly reduce manual efforts through capabilities like zero-shot learning, enabling efficient and innovative solutions to traditional challenges in education. The review emphasizes the increasing adoption of LLMs for diverse educational applications, including personalized learning experiences, automated grading systems, and tailored teaching support. By leveraging the advanced capabilities of LLMs, educators can provide more targeted interventions and improve learning outcomes. However, the authors also identify critical challenges hindering the effective integration of LLMs into education. Key issues include technological readiness, limited replicability of results, and ethical concerns related to transparency, data privacy, and fairness. To address these challenges, the study proposes several practical solutions. These include utilizing open-source LLM systems to improve accessibility and customization, adopting state-of-the-art models to enhance performance, and implementing human-centered design principles to ensure ethical and effective deployment. These strategies aim to mitigate risks while promoting the responsible use of LLMs in educational settings.

Hasanbeig et al. [4] present ALLURE, a protocol designed to audit and improve the accuracy of text evaluation by large language models (LLMs) through iterative in-context learning (ICL). The study focuses on GPT-4's evaluation of outputs from eight LLMs, identifying and addressing failure modes in two key areas: reinforcement learning (RL) planning tasks and news summarization. By systematically categorizing these failure modes and refining ICL prompts with failure-specific examples, ALLURE enhances GPT-4's evaluation accuracy. A central finding of the study is the critical role of example selection in improving ICL performance. The authors observe that the relevance of examples to specific failure modes has a more significant impact on accuracy than the quantity of examples. While introducing too many examples initially hinders performance, carefully adding optimal examples related to failure modes eventually leads to improved accuracy. This nuanced relationship between example relevance and performance emphasizes the need for a targeted approach when refining ICL prompts.

Lyu et al. [5] explore the impact of Large Language Models (LLMs) on introductory computer science education through a semester-long field study using CodeTutor, an LLM-powered virtual assistant. The study involved 50 students and compared the learning outcomes of students who used CodeTutor with those in a control group. The findings reveal that students who utilized CodeTutor achieved significant improvements in their final scores, with first-time LLM users benefiting the most from the tool. Despite these positive results, the study also notes a gradual decline in students' reliance on CodeTutor's suggestions over time, with a growing preference for human teaching assistants. Students provided favorable feedback regarding CodeTutor's ability to assist with programming syntax but expressed concerns about its limitations in fostering critical thinking and problem-solving skills. This feedback underscores the importance of balancing LLM assistance with human guidance to address more complex educational needs effectively. The study highlights that the quality of prompts plays a pivotal role in enhancing the relevance and accuracy of LLM-generated responses. To better align the capabilities of tools like CodeTutor with educational objectives, the authors advocate for integrating Generative AI literacy into computer science curricula. Such an initiative could help students develop the skills needed to use LLM-powered tools more effectively, thereby maximizing their potential benefits in education.

Ismail et al. [6] investigate the potential of Vision Language Models (VLMs) to enhance educational assessment within the framework of Indonesia's Independent Curriculum, focusing on automating the evaluation of students' essay responses. The study aims to improve the efficiency, objectivity, and

alignment of assessments with curriculum goals, addressing the limitations of traditional evaluation methods. Using a mixed sequential explanatory approach, the research demonstrates that integrating VLMs significantly enhances the accuracy, consistency, and time efficiency of assessments. The findings reveal that VLMs not only streamline the grading process but also provide timely feedback to students, enabling a more responsive and effective learning environment. Teachers involved in the study expressed positive views about the usefulness and reliability of VLM-based systems, particularly their potential to reduce workload and improve assessment quality. The study emphasizes the role of VLMs in supporting curriculum reform by addressing the challenges of traditional assessment methods, such as subjectivity and time constraints. By automating essay evaluations, VLMs enable educators to focus more on teaching and curriculum implementation, fostering a more balanced and effective educational ecosystem. The integration of VLMs aligns seamlessly with the objectives of the Independent Curriculum, which prioritizes competency-based learning and holistic student development.

Kim et al. [7] investigate the application of flipped learning and virtual rotations in radiation oncology education during the COVID-19 pandemic, focusing on the effectiveness of a hybrid learning program. The study involved 110 fourth-year medical students who participated in a week-long program combining online and in-person activities. Students first engaged with video lectures covering radiation oncology concepts, followed by interactive online discussions and site visits. The results revealed high levels of student motivation and satisfaction, demonstrating that the flipped learning approach effectively maintained educational quality despite the challenges posed by the pandemic. The study also highlights the potential of virtual reality (VR) in medical education, particularly in simulating complex processes like radiotherapy. VR technology enables students to visualize and practice procedures that would otherwise be difficult to replicate in a traditional classroom setting, offering an immersive learning experience. Additionally, Kim et al. discuss the growing role of artificial intelligence (AI) and deep learning in advancing medical training, with applications in areas such as cancer detection and treatment planning. These technologies offer promising tools for enhancing medical education by providing more accurate and tailored training opportunities. Kim et al. advocate for the continued integration of flipped learning strategies in post-pandemic medical education. They emphasize the importance of maintaining coherence between online and in-person components to ensure a seamless and effective learning experience. The authors suggest that future research should focus on evaluating the long-term impacts of flipped learning on medical education, its scalability, and the broader integration of emerging technologies like AI and VR. This research could provide valuable insights into how these innovative tools can be used to enhance medical training and improve student outcomes in the future.

Liu et al. [8] conduct a systematic study on the design of large multimodal models (LMMs) under the LLaVA framework, focusing on improving data efficiency and model performance. They investigate the effectiveness of a fully-connected vision-language connector, which proves to be unexpectedly powerful and efficient. By implementing simple modifications, such as an MLP cross-modal connector and using academic-task-oriented VQA data, the study establishes strong baselines that achieve state-of-the-art results across 11 benchmarks. The authors highlight LLaVA-1.5's data efficiency, noting that it is built on just 1.2M publicly available data samples and outperforms existing models, completing training in about one day on a single 8-A100 node. The research also addresses challenges in scaling LMMs, particularly with higher-resolution inputs, compositional tasks, and model hallucinations. By applying a grid division technique to scale LLaVA to high-resolution images, the model shows improved perception of fine details and a reduction in hallucinations. The study also demonstrates that LMMs can effectively handle compositional tasks, enhancing multimodal writing capabilities. Moreover, LLaVA maintains stable performance even with up to 75% downsampling of training data, suggesting the potential for sophisticated data compression strategies.

Zhang et al. [9] explore the ability of Vision-Language Models (VLMs) to reason about time and location in images, with a particular focus on whether these models can recognize and interpret socio-cultural cues. The study introduces the WikiTiLo dataset, a curated collection of images containing distinct visual features associated with different times and locations. The authors design a two-stage probing task—RECOGNITION and REASONING—to evaluate the performance of both discriminative and generative VLMs. This framework assesses the models' ability to recognize time- and location-relevant features and reason about these cues effectively. The study's results reveal a performance gap between discriminative and generative VLMs. Discriminative models, such as CLIP, are shown to excel in recognizing time- and location-specific features in images, performing the task in a context-agnostic manner. These models can identify visual cues associated with particular times and places, demonstrating their ability to discern relevant information from images directly. In contrast, generative VLMs, which rely on powerful language models, struggle to perform reasoning tasks effectively using these visual cues. Despite their advanced language modeling capabilities, generative VLMs face challenges in utilizing visual features for reasoning, which points to a gap in their ability to integrate and process multimodal information. The authors attribute this disparity in performance to two main factors. First, context-conditioned visual features, which provide information about time and location, are often not retained effectively in generative models. Second, the language models used in generative VLMs have limitations in reasoning with these visual features. This means that while discriminative models are proficient in recognizing visual cues, generative models encounter difficulties when tasked with complex reasoning that requires a deep understanding of socio-cultural contexts embedded in the visual data.

Lamm and Keuper [10] explore the potential of substituting traditional Optical Character Recognition (OCR)-based Visual Question Answering (VQA) pipelines with pre-trained Vision Language Models (VLMs) in real-world production environments, particularly within the retail industry. Their study utilizes the Retail-786k dataset, which contains images from product advertisements, to assess the effectiveness of VLMs in answering questions regarding product information, such as details about brands, prices, and other attributes. The authors evaluate several VLMs, including both commercial models like GPT-4V and GPT-4o, as well as open-source alternatives such as InternVL, LLaVA 1.5, LLaVA-NeXT, and CogAgent, in practical, retail-related scenarios. The results of the study indicate that VLMs perform relatively well on tasks like identifying product brands and prices. Notably, there is no significant difference in performance between the commercial models and open-source models when handling these basic tasks. However, the study highlights certain limitations in VLMs' performance, particularly in fine-grained classification tasks. For instance, the models struggle with identifying specific product names and discounts. This issue arises from the models' difficulty in processing abstract concepts, such as recognizing discounts, which are crucial in the retail context. The authors suggest that this challenge is largely due to the models' lack of specialized domain knowledge, which hampers their ability to handle such complex queries effectively. To address these limitations, Lamm and Keuper propose the integration of Retrieval-Augmented Generation (RAG) techniques. By incorporating RAG, VLMs could be supplemented with domain-specific knowledge, improving their ability to handle nuanced and specialized questions. This approach could help enhance the models' performance, particularly in tasks that require a deeper understanding of the retail domain, such as identifying discounts or interpreting specific product details.

Carbune et al. [11] investigate ways to enhance the reasoning capabilities of Vision-Language Models (VLMs) by transferring knowledge from Large Language Models (LLMs). VLMs are often limited in their ability to perform complex reasoning tasks, especially when it comes to understanding charts, plots, and diagrams. To address these limitations, the authors propose a novel technique that improves chart representations by pre-training on a chart-to-table translation task. They further fine-tune the model using

synthetic datasets that include reasoning traces generated by more advanced LLMs, which are designed to teach the VLMs how to reason through complex data visualizations. The authors evaluate their approach using the ChartQA benchmark, which tests models on their ability to answer questions about charts and diagrams. Their modified VLM, ChartPaLI-5B, outperforms the significantly larger PaLI-X model (10 times its size) on chart-based reasoning tasks. This result demonstrates that their method can achieve superior performance with far fewer parameters, showing the potential of fine-tuning and knowledge transfer in improving VLM reasoning. Additionally, the study highlights the effectiveness of using a multitask loss setup during fine-tuning, which helps the model perform better on a variety of tasks and improves its ability to handle reasoning and numerical operations more effectively.

Oh et al. [12] introduce a novel fine-tuning method called Fine-grained Selective Calibrated CLIP (FSC-CLIP), designed to enhance the compositional understanding of pre-trained vision and language models (VLMs) while preserving their performance on multi-modal tasks. Traditional fine-tuning techniques often degrade the model's ability to perform multi-modal tasks, especially when global hard negative (HN) losses are used, which can disrupt the model’s capacity to maintain accurate multi-modal representations. FSC-CLIP seeks to address this issue by incorporating two key innovations. The first innovation is the introduction of Local Hard Negative (LHN) Loss. This method improves compositional reasoning by using dense alignments between image patches and text tokens. By focusing on local relationships rather than global ones, LHN loss ensures a more accurate understanding of how different parts of an image correspond to specific parts of text, enhancing the model's ability to reason about complex compositions. The second key innovation is Selective Calibrated Regularization (SCR), which fine-tunes the hard negative supervision. SCR works by adjusting the calibration of the model, helping it distinguish between similar or confusing text inputs. This helps to reduce errors caused by similar or ambiguous text, preventing the model from becoming confused and improving its overall ability to process multi-modal data. Through extensive experiments, the authors show that FSC-CLIP outperforms existing methods, such as DAC-LLM, particularly in zero-shot recognition and image-to-text retrieval tasks. In addition to these performance improvements, FSC-CLIP also demonstrates comparable effectiveness in compositional reasoning tasks, making it a well-rounded approach for enhancing both the model’s understanding of complex compositions and its ability to perform general multimodal tasks.

Li et al. [13] introduce the Multimodal ArXiv dataset, comprising ArXivCap and ArXivQA, to enhance the scientific comprehension capabilities of Large Vision-Language Models (LVLMs). The research addresses a critical limitation in LVLMs: their struggle to accurately interpret abstract scientific figures such as geometric shapes and plots. This issue arises largely due to the lack of domain-specific training datasets tailored to scientific content. To address this challenge, the authors present ArXivCap, a figure-caption dataset that includes 6.4 million images and 3.9 million captions extracted from 572,000 scientific papers. This dataset is designed to facilitate fine-tuned evaluation of LVLMs on tasks involving scientific figures, enabling better comprehension and interpretation of visual content related to scientific research. Alongside ArXivCap, Li et al. introduce ArXivQA, a question-answering dataset generated using GPT-4V, which is aimed at enhancing LVLMs' ability to reason mathematically. This dataset plays a critical role in improving the models' performance on multimodal reasoning tasks by providing them with the tools to handle more complex mathematical and scientific reasoning, achieving a notable 10.4% improvement in accuracy on a multimodal reasoning benchmark. The authors evaluate LVLMs using ArXivCap by conducting four vision-to-text tasks, such as image captioning and interpretation of scientific figures. The results indicate that while in-domain training using datasets like ArXivCap significantly improves the performance of LVLMs, challenges remain. Specifically, LVLMs continue to struggle with misinterpreting scientific figures, recognizing certain elements incorrectly, and providing overly simplified captions that fail to capture the nuances of the images. These errors highlight the need

for further advancements in LVLMs, particularly when it comes to the comprehension of complex scientific imagery.

Li et al. [14] introduce Mini-Gemini, a framework designed to enhance the performance of multi-modal Vision Language Models (VLMs), addressing the existing performance gap between current VLMs and advanced models like GPT-4 and Gemini. The study aims to improve VLM capabilities by focusing on three main strategies: efficient high-resolution image enhancement, high-quality data integration, and VLM-guided generation. These strategies are designed to enhance the model's ability to process visual content more effectively and efficiently, while also maintaining computational efficiency. One of the key innovations of Mini-Gemini is the use of a dual-encoder system for visual token enhancement. This system allows the model to process higher-resolution images without increasing the number of visual tokens, which helps preserve computational efficiency while enriching the visual detail captured by the model. This approach enhances the model's ability to understand and interpret fine-grained visual information, such as subtle image features, without incurring significant computational overhead. Another important aspect of the Mini-Gemini framework is the incorporation of a high-quality dataset to further enhance the model's ability to understand and reason about images. By integrating meticulously curated data, the framework improves image comprehension and reasoning capabilities, enabling the model to perform better on tasks requiring complex visual interpretation. The dataset used is designed to expose the model to diverse visual contexts, ensuring that it can handle a variety of image types and reasoning tasks.

Zhou et al. [15] examine the use of Large Language Models (LLMs) in Vision-and-Language Navigation (VLN) tasks, aiming to bridge the gap between LLM-based agents and specialized VLN models. Their proposed approach, NavGPT-2, integrates visual content alignment with frozen LLMs and navigation policy networks to enhance navigational reasoning capabilities. This integration allows LLMs to be effectively utilized in VLN tasks, maintaining their general language processing abilities while enhancing their application in specific navigation tasks. The key innovation of NavGPT-2 is its ability to combine the strengths of LLMs in language understanding with VLN-specific navigational tasks. The authors demonstrate how LLMs, traditionally known for their language generation abilities, can be adapted to perform complex navigation tasks without losing their capacity for human-interpretable reasoning. The model also facilitates natural communication, allowing for more interactive navigation experiences where the agent can follow instructions and engage in multi-turn dialogues with users. This aspect of communication allows for a more human-like interaction, improving the user experience during navigation tasks. NavGPT-2 employs multi-image perception and visual instruction tuning to improve the model's ability to track long-term navigation history. This ability to retain and reference past navigation steps allows for more effective backtracking, a key feature for tasks that require understanding and responding to dynamic environments. The model's integration of visual and linguistic information enables it to navigate and respond to complex environments while considering previous actions, creating a more reliable navigation process. The authors emphasize that NavGPT-2 offers a more efficient and data-effective solution compared to existing VLN-specialized models. By combining the flexibility of LLMs with the precision of traditional VLN agents, the framework strikes a balance between high-level language processing and task-specific navigation performance. This efficiency allows for better utilization of available data and resources, making the model more scalable and adaptable to different environments. In addition to its navigational abilities, NavGPT-2 highlights the potential for LLMs to act as communicative agents, capable not only of following instructions but also engaging in dynamic, multi-round interactions with users. This opens up the possibility of more interactive and intelligent navigational AI systems, which could be applied in a range of real-world scenarios, such as robotic navigation and autonomous vehicles.

Liu et al. [16] delve into the concept of visual instruction-tuning, a technique that adapts the successful instruction-following approach of large language models (LLMs) to the multimodal domain, integrating both vision and language. In their work, they introduce LLaVA (Large Language and Vision Assistant), a multimodal model designed to combine a vision encoder with an LLM to enhance general-purpose visual and language understanding. This model aims to bridge the gap between language processing and visual comprehension, extending the capabilities of LLMs to handle multimodal inputs effectively. The study employs GPT-4 to generate multimodal instruction-following data, incorporating both images and corresponding language instructions. This data is then used to fine-tune the LLaVA model, enabling it to follow visual and linguistic instructions with improved precision. The results of the study highlight LLaVA's outstanding performance in various multimodal tasks, including Science Question Answering (QA), where it achieves state-of-the-art results, surpassing previous models in accuracy and efficiency. This demonstrates LLaVA's potential as a highly capable multimodal assistant that can process both visual and textual inputs in a way that mimics human-like understanding. Furthermore, LLaVA's performance is shown to exceed that of GPT-4 on a synthetic multimodal instruction-following dataset, which underscores the model's ability to generalize to new and unseen images and instructions. This indicates that LLaVA is not only effective on the data it was trained on but can also adapt to novel multimodal tasks, making it a versatile tool for real-world applications. The authors emphasize the importance of visual instruction tuning, suggesting that this method significantly enhances the model's ability to interpret and respond to complex visual and linguistic inputs simultaneously.

Wang et al. [17] present the Qwen2-VL series, an advanced iteration of the Qwen-VL models that redefines traditional methods for visual data processing. One of the key innovations in the Qwen2-VL series is the introduction of Naive Dynamic Resolution and Multimodal Rotary Position Embedding (M-RoPE), which significantly improve the model's ability to process visual data at varying resolutions and in multimodal settings. These mechanisms enable the model to adapt to dynamic visual inputs, providing enhanced flexibility and scalability compared to previous models that relied on fixed-resolution approaches. The study examines the scalability of large vision-language models (LVLMs) within the Qwen2-VL series, which ranges from 2B to 72B parameters. The research demonstrates that even the smaller versions of these models achieve competitive performance across several multimodal benchmarks, with the Qwen2-VL-72B model standing out as particularly impressive. It outperforms other generalist models, such as GPT-4 and Claude 3.5, in tasks involving visual perception, video understanding, and decision-making. This showcases the model's ability to effectively process and interpret complex multimodal data, making it a highly capable tool for a variety of tasks requiring visual understanding. In addition to image and video analysis, the study highlights Qwen2-VL's potential integration with autonomous devices, such as robots and mobile phones, for real-time operation based on visual inputs. This integration could pave the way for more interactive and responsive devices that can react to and understand their environments in real time. The authors also emphasize the model's ability to handle not only visual data but also multilingual contexts, enhancing its usefulness for global applications and diverse user interactions. The advancements brought by Qwen2-VL go beyond simple image processing to include dynamic video analysis and enhanced multimodal context understanding. This makes the model a versatile and robust tool for a wide range of real-world applications, from robotics to mobile technology and beyond. Looking ahead, the authors suggest that further optimization of the dynamic resolution mechanism could improve performance, particularly in tasks involving real-time video interaction. Additionally, expanding the model's capabilities to handle an even broader range of visual and multimodal tasks could unlock new possibilities for its deployment in dynamic and complex environments.

Beyer et al. [18] introduce PaliGemma, a flexible Vision-Language Model (VLM) that integrates the SigLIP-So400m vision encoder with the Gemma-2B language model, showcasing its strong capabilities

across various open-world tasks. The model excels on conventional benchmarks such as COCO captions, VQAv2, and InfographicVQA, in addition to specialized tasks like Remote-Sensing VQA and video captioning. PaliGemma's success stems from its innovative design, combining a smaller vision encoder with a relatively compact 3B parameter language model. Despite the smaller scale, it competes effectively with larger VLMs like PaLI-X and PaLM-E, offering a compelling argument for the efficiency of more compact models. The authors highlight the potential of smaller VLMs, demonstrating that they can achieve state-of-the-art results without the need for extensive instruction fine-tuning. This challenges the prevailing belief that larger models are necessary for competitive performance, suggesting that efficient model designs can deliver high results even with fewer parameters. By leveraging a base model like PaliGemma, researchers can explore further improvements through instruction tuning and fine-tuning for specialized tasks, offering a practical foundation for ongoing advancements in the field. The study also presents valuable insights into model optimization, with ablation studies identifying key components that contribute to the success of VLMs. These findings could guide future developments, enabling the creation of more efficient models that can handle a diverse range of vision and language tasks. In particular, the authors emphasize the importance of carefully balancing the vision and language components in VLMs to achieve optimal performance across various tasks, from image captioning to complex multimodal question answering. PaliGemma represents a significant step toward more accessible and efficient vision-language models, offering a new approach to achieving high performance without relying on the massive scale of existing models. This work contributes to the ongoing effort to create versatile, high-performing models that can be applied to a broad spectrum of real-world tasks, all while maintaining a focus on computational efficiency. The authors' conclusions suggest that smaller VLMs, when optimized correctly, can deliver impressive results across a range of domains, making them valuable tools for both academic research and practical applications.

Steiner et al. [19] introduce PaliGemma 2, an upgraded version of the original PaliGemma Vision-Language Model (VLM) that incorporates the Gemma 2 language model family. This new iteration features models ranging from 3B to 27B parameters, trained at multiple image resolutions (224px², 448px², and 896px²), aiming to enhance transfer learning across a wider range of tasks. The study demonstrates that PaliGemma 2 outperforms its predecessor in several standard tasks, including image captioning and Visual Question Answering (VQA), while achieving notable advancements in more complex tasks, such as text detection, table structure recognition, and radiology report generation. A key insight from the study is that increasing the model size and image resolution leads to enhanced transfer learning performance, particularly for tasks that demand significant computational power. The results indicate that PaliGemma 2's improvements are especially noticeable in tasks requiring intricate visual and linguistic comprehension, where larger models and higher resolutions prove to be crucial. The authors emphasize that these advancements allow the model to better adapt to a broader array of real-world applications, from complex visual understanding to detailed text generation. In addition to improvements in traditional tasks, PaliGemma 2 excels in previously underexplored domains, such as molecular structure recognition and optical music score recognition. These new achievements set novel benchmarks in these specialized fields, demonstrating the model's versatility and the potential of high-resolution VLMs in tackling unique and complex challenges. By extending the range of tasks PaliGemma 2 can handle, the authors underline the value of this model as a state-of-the-art solution in Vision-Language research.

Chi et al. [20] introduce LlamaGuard3Vision, an advanced multimodal safeguard system designed to ensure safe interactions between humans and AI, incorporating both text and image inputs. The research addresses limitations in previous versions of LlamaGuard, which were restricted to text-based interactions, by enhancing the system with image comprehension for better classification of multimodal prompts and responses. LlamaGuard3Vision is fine-tuned using Llama 3.2-Vision and evaluated through the

MLCommons taxonomy, demonstrating strong performance in identifying harmful content across various multimodal scenarios, which include both text and images. One of the key strengths of LlamaGuard3Vision is its robust ability to detect harmful content within multimodal inputs, significantly improving on earlier models. The study also assesses the system's resilience to adversarial attacks, showing that LlamaGuard3Vision excels in response classification when compared to prompt classification. This highlights its potential in providing more reliable safeguards for human-AI interactions that involve both visual and textual elements. However, the authors acknowledge several limitations in the current version, such as its reliance on English language processing, which restricts its utility in non-English contexts. Additionally, the system currently supports only single-image inputs, which may hinder its application in more complex scenarios involving multiple images. The potential vulnerability of the system to adversarial manipulation also remains a concern, suggesting that there is room for improvement in its robustness against such threats.

Bai et al. [21] present the Qwen-VL series, a collection of large-scale vision-language models (LVLMs) that aim to extend the capabilities of traditional large language models (LLMs) by integrating the ability to process both visual and textual inputs. Building upon the Qwen-LM language model, the Qwen-VL series incorporates visual receptors and a robust 3-stage training pipeline. This enables the models to perform well on various vision-oriented tasks such as image captioning, question answering, and visual grounding. A key strength of the Qwen-VL models is their ability to achieve fine-grained visual understanding, outperforming similar models on numerous benchmarks, demonstrating their ability to effectively process and interpret visual data. One of the distinctive features of the Qwen-VL models is their multilingual support. These models are designed to work not only with English but also with other languages, including Chinese. This multilingual functionality allows for greater versatility and accessibility across different linguistic contexts. Additionally, Qwen-VL models are capable of handling multi-image inputs, which allows them to engage in more complex interactions that involve multiple images, facilitating interleaved conversations that incorporate both visual and textual elements. This ability represents a significant step forward in enhancing the model's interactive capabilities, enabling richer, more dynamic dialogues that involve multiple visual cues.

Wang et al. [22] introduce CogVLM, an innovative open-source visual-language foundation model designed to bridge the gap between pretrained language models and image encoders. Unlike traditional alignment techniques that rely on shallow mappings of image features to the input space of language models, CogVLM employs a more sophisticated approach. The model incorporates a trainable visual expert module directly into the attention and feed-forward network layers. This integration enables a deeper fusion of vision and language features while preserving performance in natural language processing (NLP) tasks, marking a significant advancement in the field of multimodal learning. CogVLM-17B achieves state-of-the-art results across 17 diverse cross-modal benchmarks, encompassing multiple domains. These benchmarks include widely used datasets for image captioning (e.g., NoCaps, Flickr30k) and visual question answering (e.g., OKVQA, TextVQA, OCRVQA, ScienceQA). Additionally, it performs exceptionally well on benchmarks for large vision-language models (e.g., MMBench, SEED-Bench, LLaVABench, MMVet, POPE, MMMU, MathVista) and visual grounding datasets (e.g., RefCOCO, RefCOCO+, RefCOCOg, Visual7W). The breadth of CogVLM’s performance highlights its capability to tackle complex multimodal tasks effectively. The authors emphasize that CogVLM represents a paradigm shift by moving beyond the limitations of shallow alignment approaches. Its ability to achieve deeper and more integrated vision-language representations sets it apart from previous models. This enhanced fusion of modalities allows CogVLM to better understand and generate contextually rich responses across a variety of tasks, positioning it as a powerful tool in multimodal AI research.

Hong et al. [23] present the CogVLM2 family, a new generation of visual-language models (VLMs) designed to push the boundaries of image and video understanding through advanced vision-language fusion, support for higher resolutions, and expanded modality coverage. The family includes three models: CogVLM2 for image analysis, CogVLM2-Video for video understanding, and GLM-4V. These models collectively demonstrate significant advancements in handling diverse multimodal tasks. CogVLM2 incorporates an enhanced visual expert architecture, coupled with refined pre-training and post-training methodologies, enabling it to process high-resolution inputs up to 1344 × 1344 pixels. This high-resolution capability makes the model particularly effective for tasks requiring fine-grained visual understanding. For video analysis, CogVLM2-Video extends this functionality by integrating multi-frame inputs and temporal grounding, supported by automated temporal data construction. This approach improves the model's ability to analyze sequences of frames, enabling accurate and contextually aware video understanding. The CogVLM2 family achieves state-of-the-art results across a wide range of benchmarks. These include MMBench and MM-Vet, which test the models' performance on general vision-language tasks, as well as specialized benchmarks like TextVQA, MVBench, and VCGBench, which evaluate specific capabilities such as video question answering and multimodal reasoning. This exceptional performance underscores the models' versatility and their potential to address complex challenges in image and video understanding.

Tan et al. [24] introduce the Patient-level Multi-organ Pathology Report Generation (PMPRG) model, a specialized solution designed to address the challenges of generating pathology reports for whole slide images (WSIs). While vision-language models (VLMs) have demonstrated success in natural language and image understanding tasks, their application to WSIs has been constrained by the enormous size of multi-scale WSIs and the high costs of annotation. Additionally, prior approaches to automated pathology report generation often lack robust validation of their clinical utility. The PMPRG model addresses these limitations through its innovative design. It incorporates a multi-scale regional vision transformer (MR-ViT) to effectively extract features from the multi-scale structure of WSIs. This allows the model to focus on key regional features within the slides, ensuring accurate and relevant data extraction. To guide the training process, the model leverages real pathology reports, enabling it to learn how to generate detailed, clinically meaningful reports. By attending to critical regional features, the PMPRG model produces comprehensive pathology reports that align closely with clinical needs. The proposed model was evaluated on a diverse dataset that includes WSIs from multiple organs, such as the colon and kidney. It achieved a METEOR score of 0.68, underscoring its ability to generate accurate and clinically relevant pathology reports while significantly reducing manual effort. This high level of performance highlights the model's potential to streamline pathology workflows and improve efficiency in clinical practice.

Qu et al. [25] present the Pathology-Knowledge Enhanced Multi-instance Prompt Learning (PK-MIP) framework, a novel approach designed to overcome the limitations of current multi-instance learning algorithms in pathology image analysis. This framework is especially relevant for scenarios where training data is scarce, such as in clinical settings where the availability of Whole Slide Images (WSIs) is often limited due to privacy concerns or the rarity of certain diseases. Few-shot weakly supervised WSI classification is therefore a critical task in these environments. While existing approaches that leverage pre-trained models like CLIP have shown promise, they are primarily focused on patch-level analysis or language-based prompts, which can be restrictive when dealing with complex pathology patterns.

Nguyen et al. [26] introduce TQx, a groundbreaking approach designed to enhance the quantitative and explainable analysis of histopathology images using text-based methodologies. While prior vision-language models have made notable progress in computational pathology, most of them focus on aligning image-text pairs through contrastive pre-training techniques. These models have shown success

in tasks like pathology image classification via zero-shot or transfer learning. However, the authors propose that instead of solely relying on image-to-text alignment, vision-language models can also be utilized to quantify histopathology images through simple image-to-text retrieval, which would both analyze and explain the images in a more accessible way.

Liu et al. [27] introduce mTREE, an innovative framework developed to tackle the challenges of combining multi-scale image representations with textual data for Whole Slide Image (WSI) analysis. Although multi-modal learning has demonstrated potential in integrating visual and textual information, it faces significant obstacles when applied to large, high-resolution histopathology images like gigapixel WSIs. Traditional methods typically rely on manual labeling of image regions or multi-step processes to compile local image representations (such as patch-level) into global features (like slide-level), but these approaches struggle to seamlessly fuse multi-scale image features with textual information in a unified, end-to-end learning framework.

Fadeeva et al. [28] explore the use of Vision-Language Models (VLMs) for online handwriting recognition, specifically tackling the challenge of converting digital ink (handwriting captured on touchscreens with styluses) into readable text. Although VLMs have demonstrated exceptional performance in various image-based tasks, they have faced difficulties in the domain of handwriting recognition, especially when using traditional Optical Character Recognition (OCR) methods. OCR, which treats handwriting purely as an image, fails to account for the sequential and time-ordered nature of handwritten strokes, which is essential for accurately interpreting online handwriting. To address this issue, the authors propose an innovative representation of online handwriting that combines both the sequential strokes (treated as time-ordered text) and visual features of the handwriting image. This hybrid tokenized representation enables the model to better capture the inherent structure and dynamics of online handwriting, thus enhancing the recognition process. The new representation significantly improves the performance of VLMs compared to traditional OCR-based approaches. Through comprehensive experiments, the authors demonstrate that their method achieves recognition results that are either on par with or outperform state-of-the-art online handwriting recognition systems across various public datasets.

Pereira Júnior et al. [29] explore the use of pre-trained Vision-Language Models (VLMs) in recognizing handwritten mathematical expressions, specifically those written by children. This task is particularly important in the context of Intelligent Tutoring Systems (ITS), which aim to improve educational access, especially in underserved areas. However, recognizing children's handwriting, particularly their mathematical notations, is a challenging task due to factors such as poor handwriting quality, incomplete characters, and erased or unclear marks. The study evaluates the performance of several pre-trained VLMs—GPT-4V, LLaVA 1.5, and CogVLM—on a dataset consisting of 251 handwritten images containing mathematical expressions. The authors found that while these models showed potential, their performance remained suboptimal unless fine-tuned or utilized in a zero-shot learning setting. Among the models assessed, CogVLM achieved the best results. However, all models struggled with recognizing children's handwriting, particularly when the handwriting included erased sections or hard-to-read characters. The paper also noted that GPT-4V, despite being effective at recognizing mathematical equations, faced limitations due to its safety system, which hindered its ability to process handwritten content effectively.

Chauhan et al. [30] investigate the potential of Vision-Language Models (VLMs) in handwriting verification, an essential task in document forensics. Traditionally, handwriting verification has relied on deep learning methods; however, these approaches are often met with skepticism from forensic document examiners due to concerns about their lack of interpretability and their dependence on large, curated

datasets and handcrafted features. To address these challenges, the authors explore how VLMs, such as OpenAI's GPT-4o and Google's PaliGemma, can enhance handwriting verification by leveraging their Visual Question Answering (VQA) abilities and zero-shot Chain-of-Thought (CoT) reasoning. The authors conduct experiments using the CEDAR handwriting dataset to assess the effectiveness of VLMs in handwriting verification. They demonstrate that VLMs improve the interpretability of model decisions, reduce the need for extensive labeled data, and better adapt to the diverse and varied nature of handwriting styles. These models are particularly beneficial in providing human-understandable explanations of their decisions, a key aspect in forensic analysis. However, despite these advantages, the study shows that traditional deep learning models, particularly the CNN-based ResNet-18 architecture, achieve superior accuracy when compared to VLMs. On the CEDAR dataset, the ResNet-18 model attained an accuracy of 84%, while GPT-4o using a zero-shot CoT prompt engineering approach reached 70% accuracy, and the fine-tuned PaliGemma achieved 71% accuracy. These results emphasize that while VLMs show promise in improving the transparency of machine learning models and reducing the need for large, labeled datasets, they do not yet match the performance of specialized deep learning models like ResNet-18 in terms of accuracy. The findings highlight the strengths of VLMs in enhancing the interpretability of decisions, a significant advantage for forensic applications. However, the study also suggests that further research and development are needed to improve the performance of VLMs in handwriting verification tasks to make them competitive with traditional deep learning models.

Boteanu et al. [31] present the Read-Write-Learn framework as a solution to the challenges faced in handwriting recognition, particularly in contexts where annotated data is limited. Traditional handwriting recognition models rely on large supervised datasets, which include both the written content and the author's identity to capture individual writing styles. However, obtaining such annotated data can be difficult, especially for historical texts or languages that lack extensive documentation. To address this issue, the authors propose a self-learning framework that enhances the training process by integrating both a language model and a handwriting generator. The framework is divided into three distinct steps: reading, writing, and learning. During the reading step, the language model is employed to identify portions of text that are likely to be accurately recognized by the handwriting model. In the writing step, a handwriting generator is used to simulate the same writing style, thereby creating additional training data that mirrors the author's handwriting. In the final learning step, the recognition model is fine-tuned using this newly generated data, allowing it to progressively improve its ability to recognize the text in the specific style. The Read-Write-Learn framework is particularly valuable for enhancing handwriting recognition in scenarios with limited annotated data. Through experiments on historical handwritten documents, the authors demonstrate its effectiveness, showing that the framework leads to substantial improvements in recognition accuracy. By leveraging the combined capabilities of language models, handwriting generation, and incremental learning, the framework enables the recognition model to adapt to new writing styles over time. This allows the model to improve its performance without needing vast amounts of annotated training data.

Aguilar et al. [32] explore the challenges of Handwritten Text Recognition (HTR) applied to historical documents, particularly medieval and early modern manuscripts from the 10th to 16th centuries. While HTR has been effective for modern documents, it faces considerable obstacles when applied to historical texts due to their unique writing styles and the scarcity of resources, such as annotated training data. To address these difficulties, the authors leverage the TrOCR architecture—a transformer-based model designed for optical character recognition (OCR)—in combination with domain-specific large language models (LLMs) to improve the decoding of historical handwritten content. The study highlights the limitations of current HTR methods for historical documents, primarily the lack of large, annotated datasets needed for effective training. To overcome this challenge, the authors generate synthetic data

using Generative Adversarial Networks (GANs), which mimics the textual and graphical features of historical manuscripts. This synthetic dataset includes 420,000 graphical lines that closely resemble the handwriting in these documents. Additionally, the authors compile an annotated training dataset containing over 2 million tokens and 210,000 graphical text lines sourced from 52 different manuscripts in four ancient languages: Latin, French, Spanish, and High German. The study demonstrates substantial improvements in performance, with the proposed approach achieving up to a 30% relative enhancement in Character Error Rate (CER), Word Error Rate (WER), and BERT-score compared to traditional CRNN-based methods. These improvements suggest that the integration of VLMs with domain-specific LLMs and the use of GAN-generated synthetic data can significantly boost the accuracy of HTR for historical manuscripts.

Baral et al. [33] present DrawEduMath, a novel dataset designed to assess the performance of Vision Language Models (VLMs) in solving mathematical problems within educational settings. This dataset contains 2,030 images of handwritten student responses to K-12 math problems, each accompanied by comprehensive teacher annotations. These annotations provide free-form descriptions of the images and 11,661 question-answer (QA) pairs, offering valuable pedagogical insights into students' problem-solving approaches, the organization of their drawings, diagrams, and written work. The dataset serves as a benchmark for evaluating VLMs' ability to understand and reason about images of student math work. In the realm of K-12 education, VLMs are tasked with interpreting not only noisy and varied visual content but also domain-specific language and mathematical concepts. This paper evaluates how well existing VLMs perform when analyzing the annotations in DrawEduMath, which include real-world student math responses along with the pedagogical insights provided through teacher-written QAs. The findings reveal that, despite advancements in VLM technology, there is still considerable room for improvement in mathematical reasoning tasks. In particular, VLMs face difficulties when dealing with the diverse and noisy nature of handwritten math work, as well as the specialized language and concepts inherent in the annotations. The authors also explore the possibility of using synthetic QA pairs, generated from teachers' descriptions using language models (LMs), as an alternative source of training data. Although synthetic QAs are not flawless, they demonstrate a similar ranking in model evaluations to teacher-written QAs, indicating their potential usefulness for training and testing VLMs in educational contexts. This suggests that synthetic QAs could be a viable option for supplementing real-world annotations in cases where access to large amounts of annotated data is limited.

Chapter 2

System Architecture

2.1 PROPOSED SYSTEM

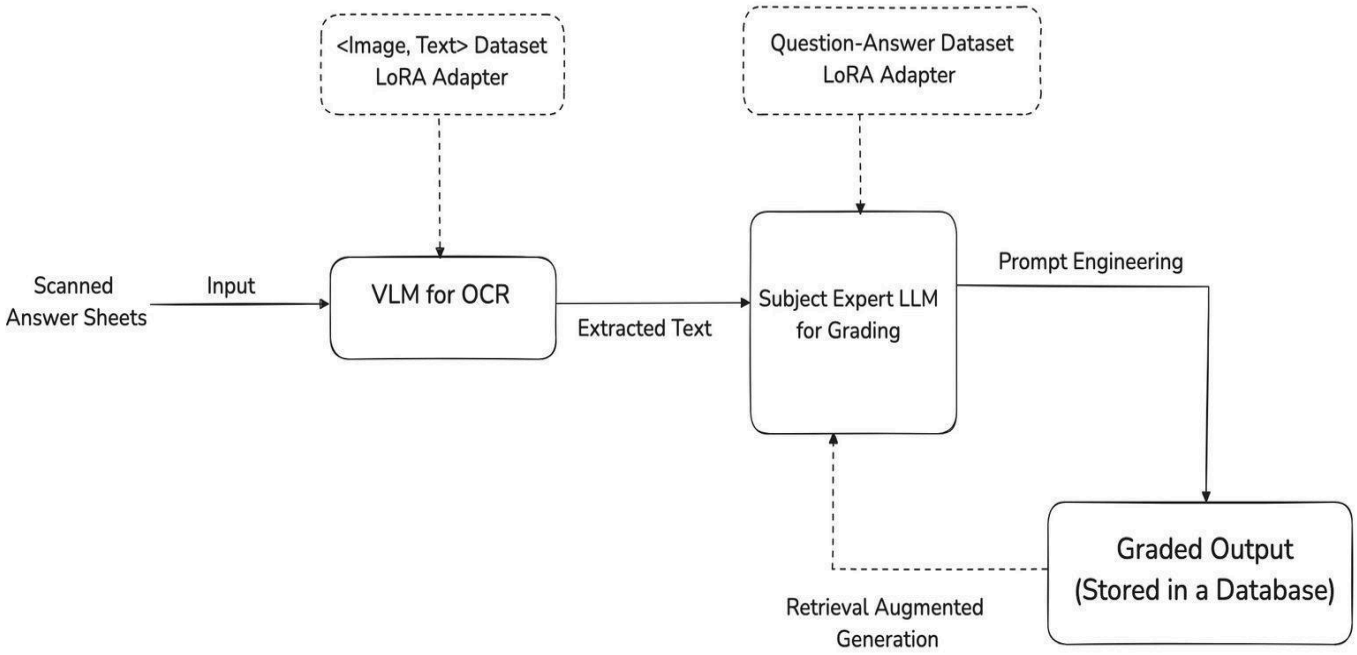


Fig 1. Proposed System Design

The proposed automated grading system for Computer Science subjects integrates advanced technologies such as computer vision, natural language processing, and machine learning to ensure accurate, scalable, and efficient assessments. The system is designed to handle diverse input formats, including scanned handwritten answer sheets, while maintaining consistency and contextual understanding throughout the grading process.

At its core, the system utilizes **Qwen2.5-VL-3B-Instruct**, a state-of-the-art Vision-Language Model (VLM) optimized for handwritten content recognition. Fine-tuned using an **<Image, Text>** dataset with Low-Rank Adaptation (LoRA), this model achieves high accuracy in recognizing handwritten text, preserving mathematical notation, and maintaining contextual integrity during transcription. The VLM converts manuscript answers into structured, machine-readable text, ensuring that critical details are

retained for subsequent evaluation.

For reasoning-based assessments, the system employs **Google Gemma-3-4B-Instruct**, a large language model specifically trained on a Data Structures and Algorithms question-answer dataset. This model excels in validating logical steps, verifying solution correctness, and awarding partial credit where applicable. Its ability to recognize alternative solution approaches, subject-specific terminology, and conceptual accuracy ensures a fair and adaptive grading process. By supporting context-aware evaluation, the model can assess logical reasoning, stepwise problem-solving, and programming methodologies, enabling a holistic assessment of student responses. This approach ensures that students receive credit for correct logic, intermediate steps, and near-correct answers, even if their final response is incorrect.

To enhance evaluation consistency, the system incorporates dynamic prompt engineering, utilizing adaptive prompt templates aligned with institutional rubrics and grading policies. These templates dynamically adjust to variations in problem structure, expected solutions, and grading criteria across disciplines, ensuring transparent and consistent grading.

An iterative feedback loop refines prompt structures based on past evaluations and identified inconsistencies, further improving long-term accuracy. The combination of Qwen2.5-VL-3B-Instruct and Google Gemma-3-4B-Instruct enhances the robustness of automated grading, improving contextual understanding and logical coherence while balancing automation efficiency with human oversight.

Finally, the system integrates a structured database to store and manage graded outputs efficiently. This database supports scalable storage, organized access to past grading results, and efficient retrieval based on predefined queries. Key functionalities include efficient search and retrieval of graded responses with feedback, tracking student performance trends over time, and facilitating adaptive learning insights by identifying common mistakes and misconceptions.

In the future, integrating Retrieval-Augmented Generation (RAG) can enhance the system's capabilities by enabling more advanced semantic retrieval and contextual analysis. By leveraging RAG, the system can generate more insightful feedback, improve grading consistency, and facilitate personalized learning by dynamically referencing and synthesizing past evaluations. Additionally, RAG enables querying for similar responses, ensuring consistency and reducing discrepancies in grading. Together, these components provide a robust and adaptable framework for technical evaluations and continuous academic assessments.

2.2 METHODOLOGY

2.2.1 LoRA (Low-Rank Adaptation)

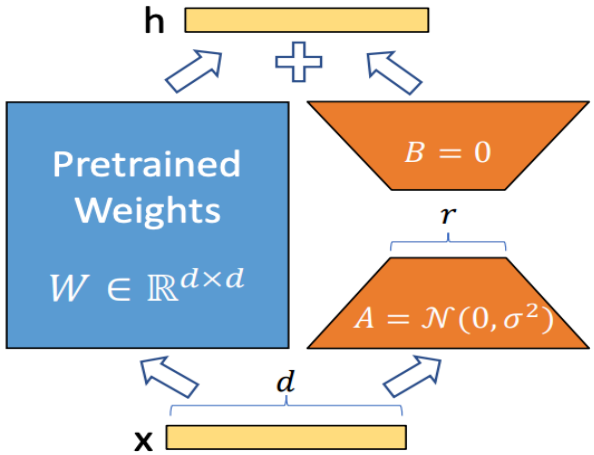


Fig. 2: Illustration of Low-Rank Adaptation (LoRA)

Fine-tuning vision-language models (VLMs) for specialized tasks, such as the automated grading of handwritten answers, necessitates a careful balance between achieving task-specific adaptation and maintaining computational efficiency. Low-Rank Adaptation (LoRA) provides an effective solution by introducing low-rank matrices to modify standard transformer-based updates during fine-tuning. This approach drastically reduces the number of trainable parameters, thereby lowering computational costs while retaining the robust generalization capabilities of the pre-trained model. By focusing on task-relevant updates without overwriting the foundational knowledge embedded in the original model, LoRA ensures efficient adaptation tailored to the unique demands of handwritten content interpretation and evaluation.

A typical fine-tuning approach updates all parameters θ of a pre-trained model to minimize a task-specific loss function L, given input x (handwritten answers + grading rubric) and target grade y:

$$\theta^* = \arg\min_{\theta} L(y, f(x; \theta))$$

where f(x;θ) represents the model's prediction. Fine-tuning all parameters is computationally expensive, particularly for large-scale models, leading to high memory and compute requirements.

In this approach, every weight matrix in the model is updated, leading to a high demand for memory and computation. For large-scale models with billions of parameters, such an approach becomes infeasible for real-world applications, especially when deploying models for large-scale automated grading in MOOCs, online assessments, and standardized testing.

Low Rank Adaptation (LoRA), instead of updating the entire weight matrix of a layer, introduces two smaller matrices and addresses this challenge by decomposing weight updates in the transformer layers into two trainable low-rank matrices.

1. A "low-rank" matrix **A** (of size *d* times *r*), where *d* represents the original dimensionality of the weights.

2. A second matrix **B** (of size (*r* times *d*)), where *r* is the rank.

The product **AB** approximates the update to the original weight matrix, allowing the model to capture task-specific patterns without altering the full set of pre-trained weights. This approach ensures that the model retains its generalization capabilities while adapting efficiently to new tasks, striking a balance between computational efficiency and performance.

$$\Delta W = AB$$

where $A \in \mathbb{R}^{d \times r}, \quad B \in \mathbb{R}^{r \times d}$, with rank r significantly smaller than d. Instead of updating the full weight matrix W, LoRA modifies it as:

$$W' = W + \Delta W = W + AB$$

Here, instead of updating the entire matrix W, LoRA only updates the smaller matrices A and B. This dramatically reduces the number of trainable parameters while maintaining the pre-trained model's knowledge and efficiency.

Advantages of LoRA in Fine-Tuning Vision-Language Models

1. Computational Efficiency

LoRA (Low-Rank Adaptation) achieves significant reductions in computational cost and memory usage by restricting updates to low-rank matrices, as opposed to modifying the entire parameter set during standard fine-tuning. This efficiency makes LoRA particularly advantageous for fine-tuning large-scale Vision-Language Models (VLMs) like Qwen2.5-VL-3B-Instruct on resource-constrained hardware, such as consumer-grade GPUs, without compromising performance.

2. Parameter Efficiency

By freezing the majority of pre-trained model parameters and only updating a small subset through low-rank matrices, LoRA preserves the model's general-purpose knowledge while enabling task-specific adaptation. This contrasts with full fine-tuning, where all parameters are updated, often leading to overfitting or loss of generalization. LoRA's approach ensures that the model retains its foundational capabilities while excelling in domain-specific tasks, such as grading handwritten responses.

3. Task-Specific Adaptation Without Overfitting

LoRA's selective fine-tuning minimizes the risk of overfitting, which is critical for educational assessment tasks. The model can effectively: Evaluate handwriting clarity and structural correctness. Align with institutional grading rubrics and marking schemes. Differentiate between minor errors and conceptual mistakes. By preserving the integrity of the pre-trained VLM parameters, LoRA ensures robust generalization across diverse handwriting styles, subject domains, and assessment formats, maintaining fairness and consistency in grading.

4. Scalability for Large-Scale Grading

LoRA's ability to fine-tune large models using only a fraction of the original parameter set makes it highly scalable for automated grading systems. This scalability is essential for:

Massive Open Online Courses (MOOCs): Handling thousands of handwritten submissions efficiently.

Institutional Assessments: Ensuring fair and consistent grading across large cohorts.

Adaptive Learning Systems: Dynamically evolving grading criteria based on student performance trends and feedback loops.

The use of LoRA (Low-Rank Adaptation) for fine-tuning Vision-Language Models (VLMs) like Qwen2.5-VL-3B-Instruct offers a highly efficient, scalable, and cost-effective solution for automating the grading of handwritten responses. By incorporating low-rank matrices into transformer weight updates, LoRA allows models to adapt to task-specific requirements while significantly reducing computational overhead. This makes it particularly advantageous for deployment in resource-constrained environments.

Unlike traditional full fine-tuning, which involves updating all parameters of a pre-trained model, LoRA selectively modifies only a small subset of low-rank matrices. This approach preserves the model's generalization capabilities, ensuring that it retains its foundational knowledge while adapting to domain-specific tasks such as evaluating handwritten responses against rubric-defined grading criteria. Additionally, LoRA minimizes the risk of overfitting, enabling more accurate and consistent assessments.

One of the most critical aspects of Qwen2.5-VL-3B-Instruct's adaptation for automated grading is its ability to align multimodal inputs effectively. The grading process demands precise correlation between visual features extracted from handwritten responses and text-based rubric guidelines, which define grading standards. LoRA facilitates this alignment by optimizing the model's attention mechanisms, enhancing its ability to recognize and evaluate handwritten mathematical notation, diagrams, and structured responses.

By drastically reducing the number of trainable parameters, LoRA facilitates fine-tuning on limited hardware, such as consumer-grade GPUs. This efficiency supports scalable deployment across diverse educational settings, including institutional assessments, online learning platforms, and standardized testing environments. As a result, LoRA not only enhances the practicality of automated grading systems but also ensures their adaptability and reliability in real-world applications.

2.2.2 Prompt Engineering:

Prompt engineering is a critical technique for optimizing the performance of LLMs and VLMs in automated grading systems. Leveraging Prompt Engineering ensures grading consistency, interpretability, and adaptability by carefully structuring inputs, constraints, and contextual guidance for the model. This methodology enables AI-based grading systems to provide accurate, context-aware, and rubric-aligned assessments without requiring external retrieval mechanisms. These components enable the model to understand complex grading criteria, recognize alternative solutions, and ensure fairness across different handwriting styles, response formats, and problem-solving approaches.

The contribution to the prompt-engineered grading process can be structured into three key phases:

The **First Phase** involves the design and implementation of context-rich prompts that guide the model in assessing responses based on specific grading criteria. The goal is to ensure that the model comprehends grading rubrics, domain-specific expectations, and assessment nuances without requiring external retrieval mechanisms.

Key Strategies in the First Phase:

1. **Rubric-Driven Prompt Templates**

Prompts are structured with explicit evaluation rubrics detailing scoring guidelines (e.g., "Award 3 points for a correct algorithm with proper time complexity analysis, 2 points for an incomplete but logically valid approach, and 0 points for incorrect reasoning"). These templates ensure that grading remains aligned with predefined educational standards and avoids inconsistencies.

2. **Contextualizing Expected Answers**

Prompts include sample solution structures to help the model recognize well-formatted, logically correct, and pedagogically sound responses. Instead of retrieving past solutions, prompts embed common reasoning patterns to guide assessment logic.

The **Second Phase** ensures that the system remains flexible and adaptable to different response formats, handwriting variations, and problem-solving styles. Unlike RAG-based methods that retrieve reference materials, adaptive prompting techniques enable the model to dynamically adjust to new responses without requiring additional training data.

Key Strategies in the Second Phase:

1. **Handwriting & Diagram Interpretation Prompts**
   For Vision-Language Models (VLMs), prompts include structured guidance on interpreting handwritten responses, mathematical notations, and diagrams. Example: "If the response includes a graph, evaluate whether the axes are labeled correctly and whether the plotted data matches the described solution." This technique ensures consistent grading of handwritten code, logic diagrams, and equations.

2. **Diversity-Aware Prompt Adjustments**

Prompt variations account for different phrasing styles, alternative solution approaches, and regional terminologies. The model is guided to recognize multiple correct answers and evaluate solutions holistically.

3. **Dynamic Template Expansion**

The system generates adaptive prompt variations based on question complexity, expected answer length, and required reasoning depth. Example: For multiple-choice answers, the model is instructed to verify correctness without penalizing minor wording variations. For long-form problem-solving questions, the prompt directs the model to evaluate step-by-step logical correctness and provide structured feedback.

4. **Bias Reduction via Structured Prompts**

Carefully designed neutral prompts prevent the model from favoring specific phrasing styles, cultural assumptions, or gendered language. Example: Instead of asking, "Does the student's solution follow the

expected pattern?" a structured prompt may use: "Evaluate the correctness of the student's response based on logic, clarity, and efficiency criteria."

The **Third Phase** ensures that the system remains reliable, interpretable, and scalable by incorporating confidence-based scoring mechanisms, human review triggers, and iterative prompt refinements.

Key Strategies in the Third Phase:

1. **Confidence-Scored Evaluations**

The model assigns a confidence score to each grading decision, helping identify low-confidence cases that require human review. Example: If the confidence score is below 70%, the system flags the response for manual assessment.

2. **Self-Consistency Prompting**

The system uses self-check prompts to ensure consistency: Compare your current grading decision with similar previous cases. If inconsistencies exist, adjust accordingly. This method prevents grading drift and ensures long-term reliability.

3. **Multi-Step Feedback Refinement**

Instead of providing an immediate final grade, the model undergoes multi-step validation: Initial Grading Pass: Assigns a preliminary score based on predefined rubrics. Error Analysis: Reviews common mistakes and applies error-based adjustments. Final Score Validation: Ensures the score aligns with overall grading trends before final submission.

4. **Human-AI Collaboration for Continuous Learning**

The system integrates human-in-the-loop refinement, where teachers review flagged responses and provide feedback-driven prompt updates to enhance future grading accuracy. Example: If a model frequently misgrades recursion-based answers, educators refine the prompt structure to emphasize recursive logic correctness criteria.

2.3 LANGUAGE MODELS

2.3.1 Qwen2.5-VL-3B-Instruct

Qwen2.5-VL-3B-Instruct, developed by the Qwen team at Alibaba Cloud, is a cutting-edge vision-language model that integrates advanced visual and textual data processing capabilities. As a smaller-scale variant with 3 billion parameters, it offers a balance between performance and computational efficiency, making it suitable for edge AI applications.

Qwen2.5-VL-3B-Instruct introduces a Dynamic Resolution Mechanism that adjusts image resolution based on input characteristics, improving OCR accuracy in applications like medical imaging and remote sensing. Its Enhanced Visual Localization uses bounding boxes and point-based detection for precise object identification, generating structured JSON outputs for seamless integration with tracking and inspection systems. The model's Advanced Document Parsing accurately extracts structured data from invoices, tables, and charts while preserving layout integrity, benefiting industries like finance and healthcare. It also supports Long Video Comprehension, enabling contextual segmentation and event detection in extended video sequences, useful for surveillance and media analysis. Beyond

vision-language tasks, the model acts as an interactive AI agent, assisting in software automation, GUI interaction, and intelligent workflows, expanding its applications in business and consumer AI systems.

Architecture of Qwen2.5-VL-3B-Instruct:

Qwen2.5-VL-3B-Instruct is a vision-language model (VLM) with a 3-billion-parameter transformer architecture, designed for efficient OCR, image understanding, and video processing. It incorporates several key innovations to enhance multimodal learning and structured data extraction:

1. **Naive Dynamic Resolution Mechanism**
   This mechanism dynamically adjusts image resolution based on input characteristics, optimizing processing efficiency without compromising accuracy. By tailoring resolution to content complexity, the model ensures effective text and object recognition across varied visual environments, such as scanned documents, low-quality images, and high-resolution medical scans.

2. **Multimodal Rotary Position Embedding (M-RoPE)**
   M-RoPE enhances the model's ability to align text and vision by encoding spatial relationships between objects and text within an image. This improves contextual understanding, allowing the model to accurately interpret structured documents, recognize handwriting, and maintain the spatial integrity of multi-column layouts, forms, and tables.

3. **Unified Image and Video Processing**
   Unlike traditional OCR models that specialize in either static images or videos, Qwen2.5-VL-3B-Instruct employs a frame-wise attention mechanism to process both formats seamlessly. This enables high-accuracy text extraction from dynamic content, making it suitable for applications such as real-time subtitle generation, video surveillance analysis, and multimedia indexing.

4. **Enhanced Visual Perception**
   The model utilizes self-attention layers and cross-modal encoders to improve recognition in cluttered, low-light, or noisy environments. By refining object detection and text localization, it enhances OCR performance in challenging scenarios, such as extracting text from overlapping elements in receipts, invoices, and scanned historical documents.

5. **Hierarchical Document Parsing Module**
   Qwen2.5-VL-3B-Instruct integrates a layout-aware transformer that preserves document structure while extracting text from complex layouts. This ensures accurate representation of tables, charts, and structured forms, making it particularly useful for financial, legal, and medical document processing, where maintaining relationships between elements is critical.

6. **Interactive AI Capabilities**
   The model incorporates reinforcement learning-based decision-making, enabling it to interact with

graphical user interfaces (GUIs) and execute tasks based on extracted text. This extends its applications beyond OCR, supporting automated document processing, intelligent workflow automation, and AI-driven customer service by responding dynamically to user queries and structured data requests.

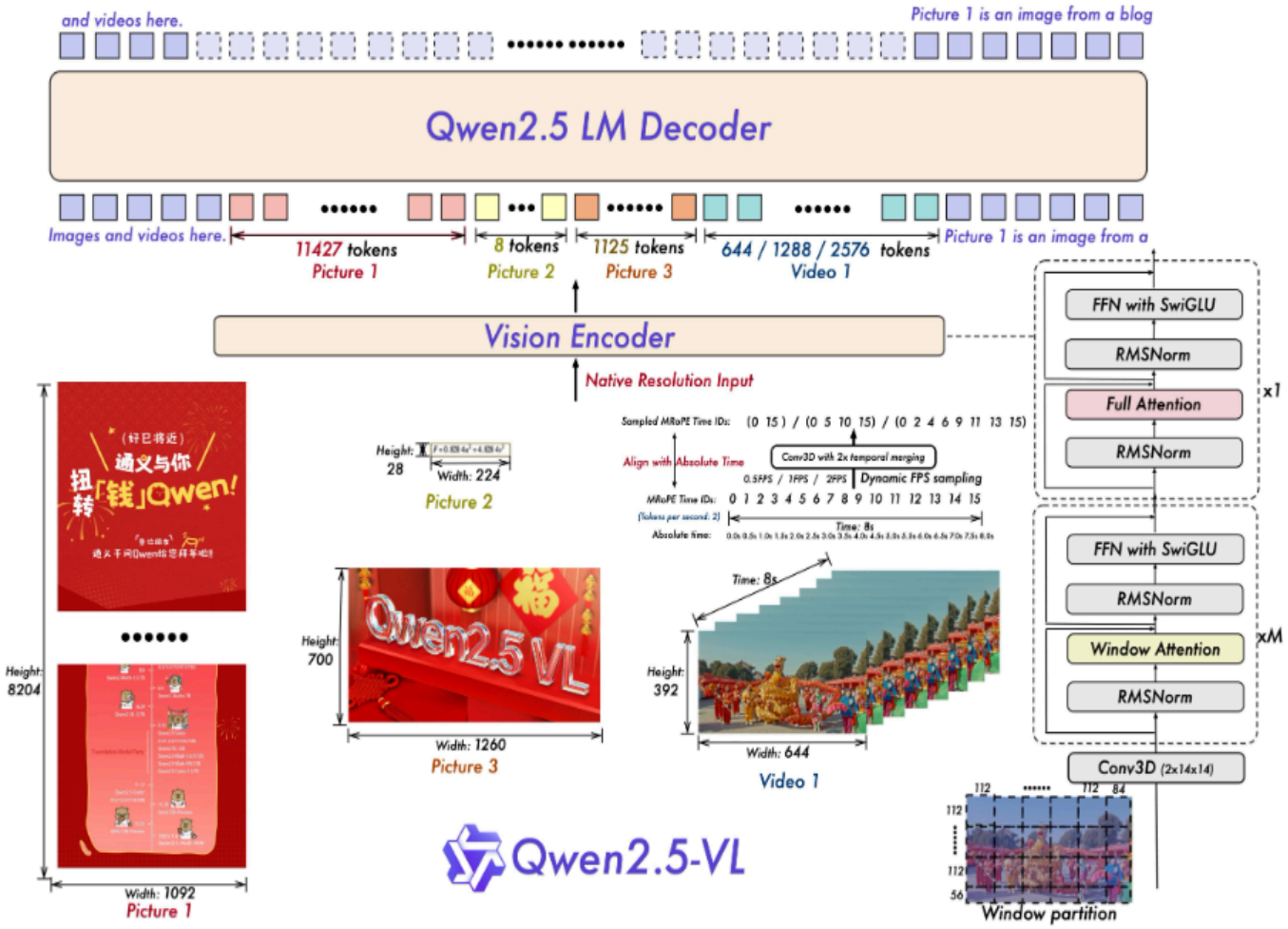


Fig. 3: Architecture of Qwen2.5-VL

2.3.2 Google Gemma-3-4B-Instruct

The Google Gemma-3-4B-Instruct Model is a state-of-the-art, lightweight model developed by Google, designed to deliver high performance in specialized tasks while maintaining computational efficiency. It is part of Google's broader Gemma family of models, which focuses on creating smaller, more efficient AI systems tailored for specific use cases. With 4 billion parameters, this model strikes a balance between size and capability, making it ideal for applications where resource constraints are a concern but high-quality outputs are still required.

The Google Gemma-3-4B-Instruct model is built on a transformer-based architecture, optimized for instruction-following tasks. Unlike larger models that rely on brute-force scaling, Google

Gemma-3-4B-Instruct emphasizes efficiency through several key design choices:

Parameter Efficiency: Despite its relatively small size, the model achieves impressive performance by leveraging advanced techniques such as Low-Rank Adaptation (LoRA) and sparse attention mechanisms. These methods reduce the computational load while maintaining the model's ability to understand and generate complex responses.

Task-Specific Fine-Tuning: Google Gemma-3-4B-Instruct is pre-trained on a diverse dataset but fine-tuned specifically for instruction-following tasks. This allows the model to excel in applications where precise, context-aware responses are critical, such as customer support, technical assistance, and educational tutoring.

Multi-Modal Capabilities: While primarily a text-based model, Google Gemma-3-4B-Instruct can integrate with external systems to handle multi-modal inputs, such as images or structured data, making it versatile for a wide range of applications.

Supervised Fine-Tuning: The model is initially fine-tuned on a curated dataset of instruction-response pairs, covering a broad spectrum of domains. This phase ensures that the model learns to follow instructions accurately and generates contextually relevant outputs.

Reinforcement Learning from Human Feedback (RLHF): After fine-tuning, the model undergoes RLHF, where it is trained using feedback from human evaluators. This process helps the model refine its responses, improving aspects such as clarity, coherence, and alignment with user intent.

Continuous Learning: Google Gemma-3-4B-Instruct is designed to support continuous learning, allowing it to adapt to new tasks and domains over time. This is achieved through periodic updates and fine-tuning based on user interactions and feedback.

It is ideal for scenarios where larger models may be impractical due to resource constraints. Its key applications span multiple domains, including customer support, education, technical assistance, content generation, and healthcare. The model enhances AI-driven chatbots, intelligent tutoring systems, technical support platforms, and document summarization tools, while also assisting healthcare professionals with medical information retrieval and decision support.

A major advantage of this model is its ease of deployment. It runs efficiently on standard hardware, making it accessible to a broad range of users, from individual developers to large enterprises. Deployment options include cloud-based hosting on Google Cloud for scalability, on-device deployment for offline usage and privacy-sensitive environments, and seamless integration into existing workflows via APIs.

Architecture of Google's Google Gemma-3-4B-Instruct:

Google Gemma-3-4B-Instruct is a 4-billion-parameter variant within the Gemma 3 series, designed to deliver robust performance while maintaining computational efficiency. Its architecture encompasses several key components:

Transformer Decoder Architecture: Google Gemma-3-4B-Instruct utilizes a decoder-only Transformer architecture, focusing on autoregressive text generation. This design enables the model to predict

subsequent tokens based on preceding context, facilitating tasks such as text completion and summarization.

Parameter Configuration: The 4B model features an embedding dimension (d_model) of 3,072, with 28 Transformer layers. Each layer incorporates a feedforward network with a hidden dimension of 49,152, enhancing the model's capacity to learn intricate patterns. The multi-head attention mechanism consists of 16 attention heads, each with a head size of 256.

Extended Context Window: To manage extensive textual data,Google Gemma-3-4B-Instruct supports a context window of up to 128,000 tokens. This extended context is achieved through interleaved local-global attention mechanisms, optimizing memory usage and enabling the model to process longer sequences efficiently.

Multimodal Capabilities: Beyond text processing, Google Gemma-3-4B-Instruct integrates multimodal functionalities, allowing it to interpret both text and visual data. This is facilitated by the SigLIP image encoder, which converts images into token representations compatible with the language model, enabling tasks that involve both textual and visual information.

Multilingual Support: The model is trained on a diverse dataset encompassing over 140 languages, enhancing its ability to understand and generate text across various linguistic contexts. This extensive language support is bolstered by a tokenizer with 262,000 entries.

Quantization and Efficiency: Google Gemma-3-4B-Instruct offers various quantization levels, including 4-bit precision, reducing memory usage and computational requirements while maintaining performance. This allows deployment on devices with limited resources.

2.4 DATASET

2.4.1 Handwritten Answer Script Dataset

To enhance automated grading capabilities in Data Structures and Algorithms (DSA), a meticulously designed handwritten dataset was developed. This dataset comprises 55 handwritten answer scripts, each varying in handwriting styles, textual clarity, and diagrammatic representations. The inclusion of diverse writing styles ensures the robustness of the grading model by enabling it to generalize across different handwriting patterns, including cursive, print, and mixed styles. The dataset also includes visual elements such as trees, graphs, and linked lists, which are crucial in DSA problem-solving.

A significant challenge in automated grading is the accurate extraction of textual and graphical elements from handwritten responses. To address this, the dataset is processed using Optical Character Recognition (OCR) and Vision Transformers. These technologies enhance the model’s ability to recognize characters, interpret structural relationships within diagrams, and map them to corresponding algorithmic concepts. By integrating multi-modal processing, the dataset allows the grading model to understand both textual and visual components, providing a holistic assessment of student responses.

Furthermore, the dataset covers a broad range of DSA problems, including recursion, dynamic programming, graph traversal, and tree balancing techniques. The answers include multiple solution approaches, allowing the grading model to differentiate between correct, partially correct, and incorrect solutions. This diversity in responses ensures that the grading model can recognize variations in problem-solving approaches, making it adaptable to real-world academic assessments.

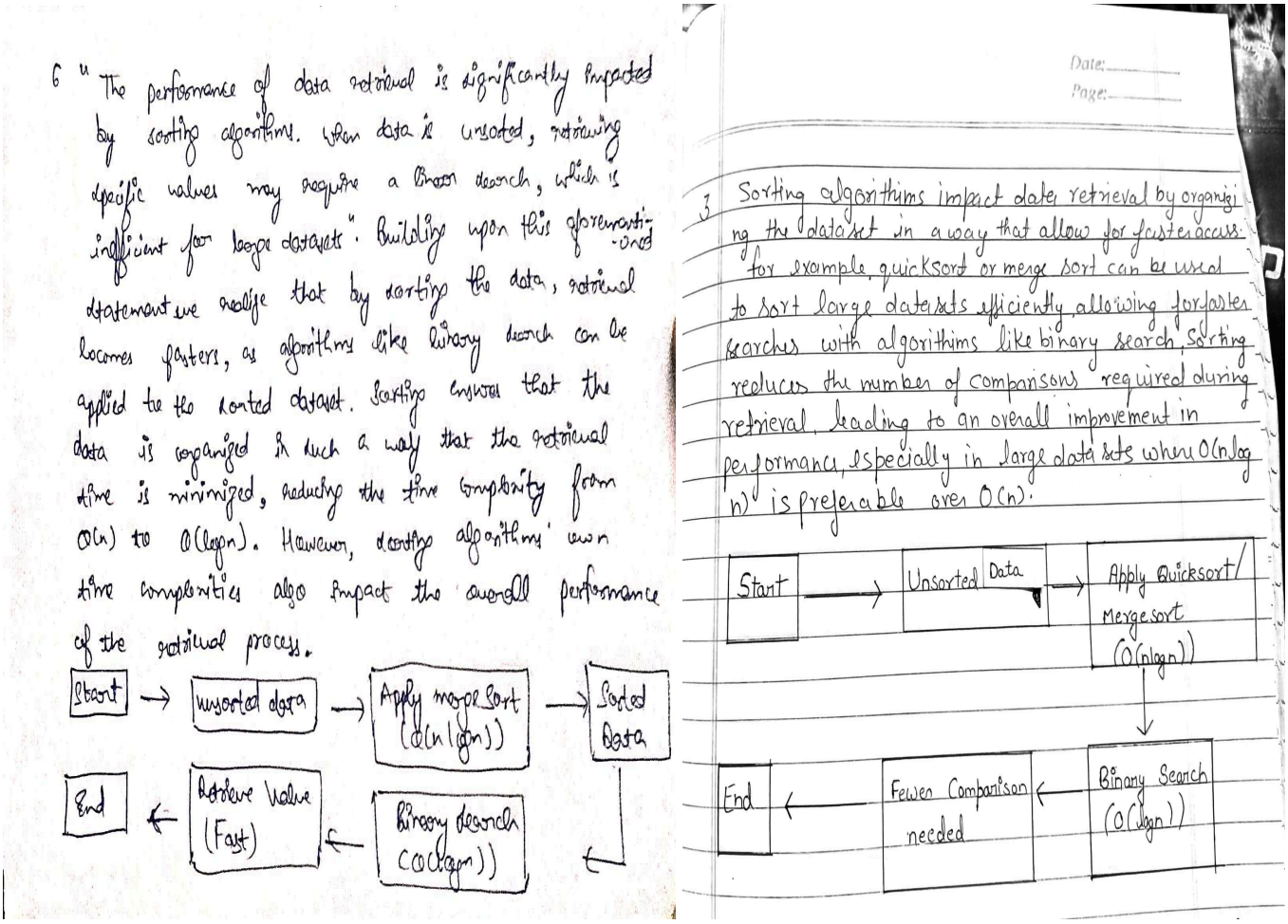


Fig. 4: Sample Answer Script

2.4.2 DSA Question-Answer Dataset

In addition to processing handwritten scripts, the system utilizes a structured **QA Dataset** in JSON format to standardize the grading process. Each entry in the QA Dataset corresponds to an individual answer script and includes detailed annotations with key components such as the problem statement, the student's response, and the correct solution approach. This structured format ensures that evaluations are consistent and aligned with well-defined grading criteria, enabling the model to apply these standards uniformly across diverse submissions.

The **QA Dataset** comprises 700 question-answer pairs from the domain of Data Structures and Algorithms, covering a wide range of topics, including linked lists, trees, graphs etc. has been generated using DeepSeek R1 model. This diverse mix of problems is used to fine-tune the **Gemma-3-4B-Instruct**

model, equipping it with a robust knowledge base for assessing student responses. By training on this dataset, the model develops a deep understanding of both fundamental concepts and complex problem-solving methodologies, ensuring accurate and context-aware evaluations.

This dataset plays a pivotal role in enhancing the model's explainability by enabling it to provide meaningful feedback rather than merely assigning scores. For instance, the model learns to reference specific elements of the structured assessments—such as logical errors, incomplete steps, or alternative valid solutions—to offer detailed insights into student performance.


```
{
  "questions": [
    {
      "question": "What is the time complexity of the Fast Fourier Transform (FFT) algorithm?",
      "answer": "The Fast Fourier Transform (FFT) algorithm has a time complexity of O(n log n), where n is the number of points in the input. This efficiency makes it widely used in signal processing, polynomial multiplication, and other applications requiring frequency domain analysis."
    },
    {
      "question": "How does the Knuth-Morris-Pratt (KMP) algorithm improve string matching?",
      "answer": "The Knuth-Morris-Pratt (KMP) algorithm improves string matching by preprocessing the pattern to create a partial match table (prefix function). This table helps avoid redundant comparisons, reducing the time complexity to O(n + m), where n is the text length and m is the pattern length."
    },
    {
      "question": "What is the purpose of the A* search algorithm?",
      "answer": "The A* search algorithm is used for finding the shortest path between two nodes in a graph. It combines the advantages of Dijkstra's algorithm (guaranteed shortest path) and greedy best-first search (efficiency) by using a heuristic function to estimate the cost to the goal."
    },
    {
      "question": "Explain the concept of dynamic programming in algorithm design.",
      "answer": "Dynamic programming is a method for solving complex problems by breaking them into simpler subproblems. It stores the results of subproblems to avoid redundant computations, optimizing both time and space. Examples include the Fibonacci sequence and the knapsack problem."
    },
```


Fig. 5: QA Dataset

Chapter 3

System Implementation

3.1 IMPLEMENTATION

3.1.1 LoRA Rank (r)

The rank (*r*) defines the dimensionality of the low-rank matrices introduced into the model's layers during fine-tuning. These matrices serve as an efficient approximation of the full parameter updates that would traditionally require modifying all model parameters—an approach that is computationally expensive. By leveraging low-rank decomposition, LoRA significantly reduces the number of trainable parameters while still enabling meaningful and effective updates.

The rank *r* is a critical hyperparameter in Low-Rank Adaptation (LoRA) that determines the dimensionality of the low-rank matrices used to approximate weight updates during fine-tuning. Its significance lies in striking a balance between model performance, computational efficiency, and generalization:

Control Over Model Capacity :

The rank r directly influences the expressiveness of the low-rank matrices. A higher rank allows the model to capture more complex patterns and nuances in the data, enabling richer task-specific adaptations. However, this comes at the cost of increased computational overhead and memory usage. Conversely, a lower rank reduces the number of trainable parameters, making the process more efficient but potentially limiting the model's ability to learn intricate details.

Trade-Off Between Efficiency and Accuracy :

By choosing an appropriate rank, LoRA achieves a trade-off between computational efficiency and fine-tuning accuracy. A smaller rank minimizes resource consumption, making it suitable for resource-constrained environments like consumer-grade GPUs. However, if the rank is too small, the approximation may become overly simplistic, leading to suboptimal performance. Selecting an optimal rank ensures that the model can adapt effectively without excessive computational demands.

Prevention of Overfitting :

A lower rank inherently constrains the model's capacity to overfit to the fine-tuning data. This is particularly valuable in scenarios where the fine-tuning dataset is small or noisy, such as in educational assessments with limited annotated examples. By restricting updates to a low-dimensional subspace, LoRA ensures that the model retains its pre-trained generalization capabilities while adapting to new tasks.

3.1.2 LoRA Alpha (α)

The alpha (α) parameter serves as a scaling factor applied to the low-rank updates before they are merged back into the original model weights. It controls the magnitude of the parameter changes introduced by LoRA during fine-tuning, ensuring that the updates are appropriately weighted relative to the pre-trained weights.

During fine-tuning, the low-rank update **AB** is scaled by ***α/r*** before being added to the original weights:

$$W_{new} = W_{original} + (\alpha/r)AB$$

where,

α amplifies or dampens the impact of the updates.

dividing by r ensures that the scaling remains proportional to the rank.

A common practice is to set $\alpha = r$ ($r = 8$, $\alpha = 8$), as this maintains a balanced scaling of updates. This configuration ensures that the updates are neither too aggressive nor too weak, promoting stable and effective fine-tuning. For our grading system , keeping α proportional to r ensures that the model adapts effectively to the nuances of DSA grading without overwriting its pre-trained reasoning capabilities.

Alpha allows precise control over how much influence the low-rank updates have on the original model. A higher α increases the impact of the updates, enabling more significant adaptation to the target task. Conversely, a lower α ensures that the updates remain subtle, preserving the model's generalization capabilities.

In the context of automated grading, α plays a key role in balancing domain-specific adaptation with the retention of pre-trained knowledge. For example, in DSA grading, where logical reasoning and step-wise correctness are crucial, α ensures that the model learns to evaluate these aspects without losing its foundational understanding of language and structure.

3.1.3 VLM Fine-tuning

The fine-tuning process leverages LoRA, a parameter-efficient technique designed to reduce computational costs while maintaining high performance. We configure LoRA rank (r) to 16 and LoRA alpha (α) to 16 , reflecting the complexity and demands of performing OCR on handwritten answer scripts. A higher rank (e.g., $r = 16$) is chosen to allow the model to capture more intricate patterns and nuances in the data, such as variations in handwriting styles, sizes, and orientations. Handwriting recognition is inherently challenging due to its diversity and unpredictability, and a higher rank enables the model to learn more expressive representations of these visual features. This ensures that the model can effectively extract text from images while retaining the fine-grained details necessary for accurate OCR.

Similarly, $\alpha = 16$, set proportionally to the rank, amplifies the impact of the updates during fine-tuning. This helps the model adapt more robustly to the unique characteristics of the handwriting dataset, capturing subtle differences that lower-rank configurations might miss. This is critical for learning

underlying patterns in handwriting, which often involve nuanced variations that require careful attention.

The dataset used for fine-tuning consists of 50 pairs of <Image, Text> samples , including scanned handwritten answer scripts and their respective text content, along with descriptions of flowcharts (if any) in natural language. During training, the model performs OCR to extract text from these images, ensuring it can handle both textual and visual elements. The training configuration specifies 25 epochs to allow sufficient iterations for the model to learn these tasks effectively. Handwriting datasets are typically diverse and noisy, requiring extended training to ensure the model generalizes well across different handwriting styles and maintains high OCR accuracy.

Memory usage during training is optimized through techniques like gradient checkpointing and mixed precision training (bfloat16) . These methods reduce memory overhead while maintaining computational efficiency. Monitoring the training dynamics reveals valuable insights into the model's learning process. Initially, the training loss starts high (around 0.83) but decreases sharply within the first two hundred steps, indicating rapid adaptation to the dataset. By the end of training, the loss stabilizes near 0.35, suggesting convergence and effective learning.

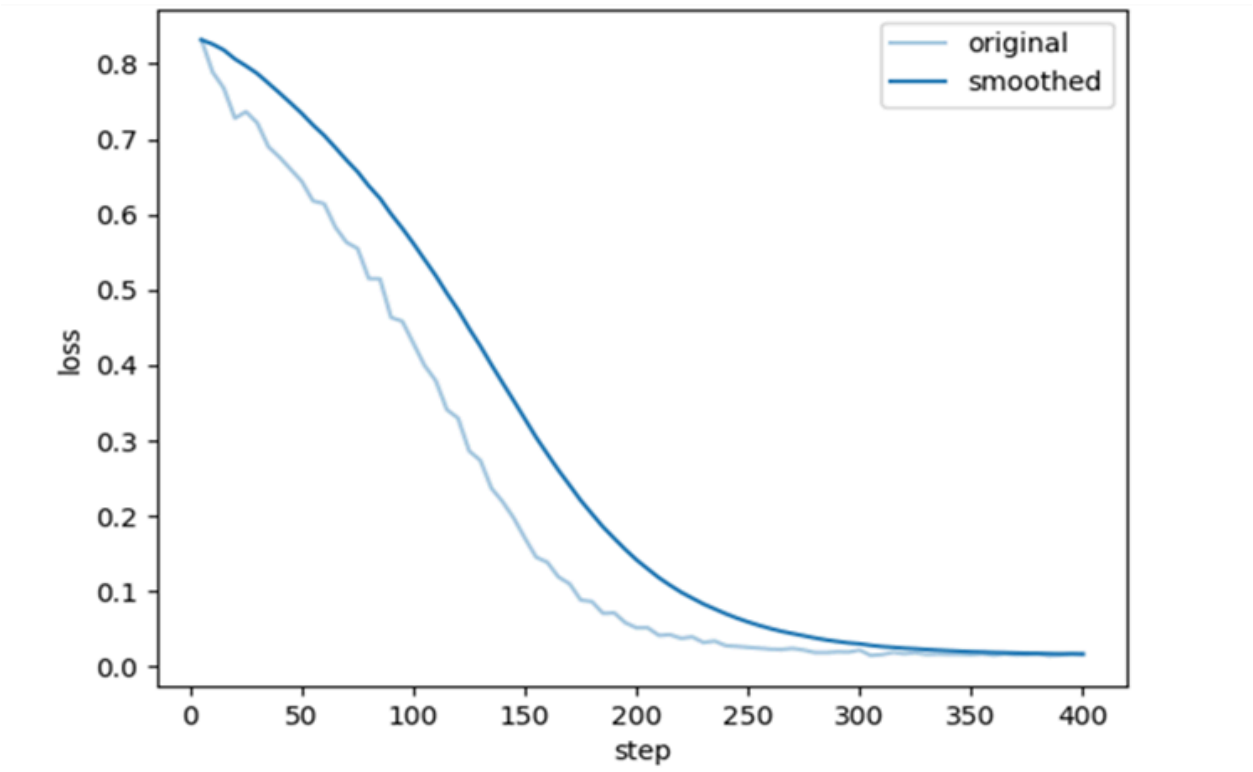


Fig. 6: VLM Fine-tuning Loss Curve

3.1.4 LLM Fine-tuning

We present a fine-tuning approach for domain-adapting the Google Gemma-3-4B-Instruct model using parameter-efficient fine-tuning techniques (PEFT) on a specialized DSA dataset. Our approach includes LoRA configurations to achieve memory-efficient fine-tuning while preserving model capabilities.

The fine-tuning process involved structured adaptation of Gemma-3-4B-Instruct on a curated dataset of 700 DSA question-answer pairs, employing LoRA (r = 8, α = 8) for efficient training. The choice of LoRA hyperparameters for fine-tuning Google Gemma-3-4B-Instruct is intentional and aligns with the model's specific purpose: grading answer scripts rather than solving or generating DSA question-answer pairs.

A low rank (r = 8) ensures that the model undergoes parameter-efficient updates, minimizing computational costs while still adapting to the nuances of grading tasks, such as recognizing logical steps, validating correctness, and assigning partial credit. Since the goal is not to generate complex solutions or solve problems but rather to evaluate existing ones, a high rank is unnecessary and could lead to overfitting on the grading dataset. This conservative configuration strikes a balance between task-specific adaptation and retention of the model's foundational reasoning capabilities, making it well-suited for grading purposes without over-complicating the learning process.

The model was optimized using SFTTrainer with a batch size of 2 (effective size=8 via gradient accumulation), learning rate of 2e-4, and 50 epochs, while monitoring key metrics. Post-training evaluation included gradient norm analysis to assess optimization stability, training loss curves to track convergence, and learning rate scheduling to verify proper adaptation dynamics.

The implementation involves checking the environment (local Jupyter Notebook) and installing the necessary dependencies and specific libraries like bitsandbytes, accelerate, xformers, and unsloth are installed to enable efficient fine-tuning with quantization and LoRA support. The latest Hugging Face Transformers library (version 4.49.0 with Gemma-3 support) is also installed to ensure compatibility with the Gemma-3-4B-Instruct model. The FastModel class from Unsloth is then used to load the pre-trained model (google/gemma-3-4b-it), which reduces memory usage while maintaining accuracy during LoRA fine-tuning.

During training, GPU memory usage is monitored. The setup uses an NVIDIA 2xH100 PCIe GPU cluster with a total memory of 2x79.097 GB . Initially, 4.225 GB of memory is reserved, and after training, the peak reserved memory reaches 4.631 GB , with only 0.406 GB allocated specifically for LoRA fine-tuning. This represents 5.855% of the GPU's total memory, with LoRA accounting for just 0.513% , showcasing the efficiency of parameter-efficient fine-tuning. Training takes 2999.53 seconds (approximately 50 minutes) over 2,100 steps , with 16 total batch size (4 per device × 4 gradient accumulation steps × 1 GPU). The model trains only 14,901,248 parameters out of 4 billion , or 0.37% , demonstrating LoRA's ability to minimize trainable parameters.

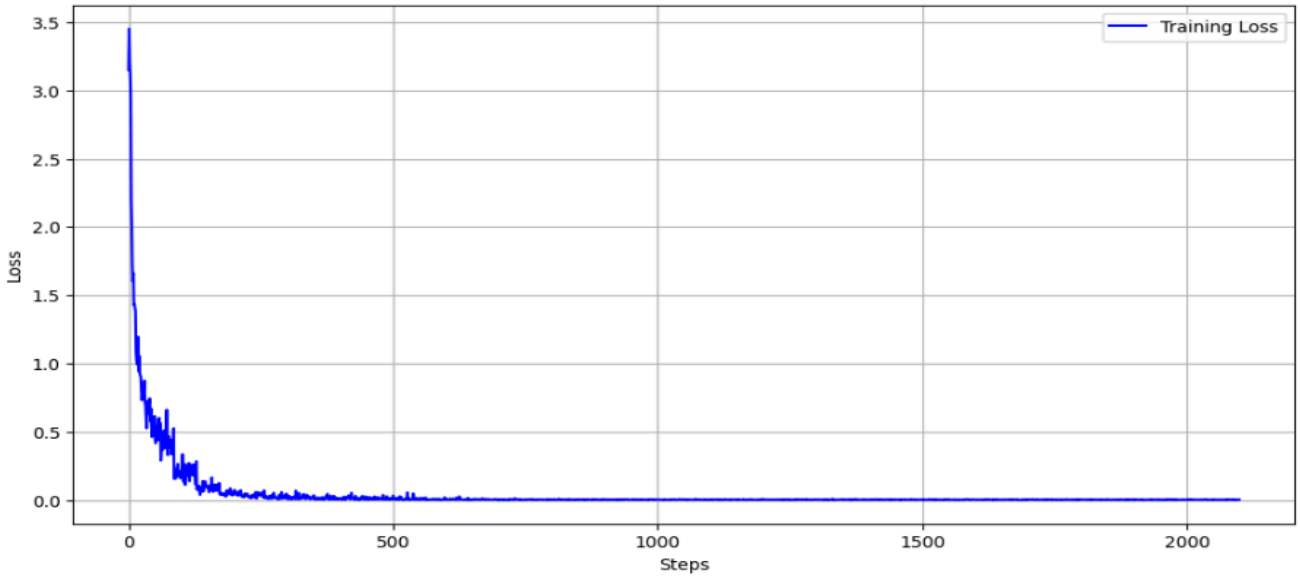


Fig. 7: Training Loss Curve (Google Gemma-3-4B-Instruct)

The **Training Loss** represents the error computed on the training dataset at each logging step, serving as a direct measure of how well the model is learning to fit the data by quantifying the discrepancy between its predictions and the actual target values. A decreasing trend in the training loss indicates that the model is progressively minimizing its prediction errors, suggesting effective learning and adaptation to the training data. This behavior is clearly reflected in the Training Loss Curve graph, which shows the model's error evolving over the course of training. Initially, at step 0, the training loss starts at a relatively high value (approximately 3.5) and experiences a sharp drop within the first few hundred steps, indicating that the model is quickly adapting to the training data by making significant adjustments to its parameters to reduce large prediction errors. As training progresses, the rate of decrease slows down significantly between approximately step 500 and step 1500 , where the loss gradually stabilizes around a much lower value, nearing 0 . This gradual stabilization reflects the model's fine-tuning process, where smaller, more precise updates refine its performance on the training data. By the end of training (after step 1500), the training loss plateaus near 0 , signaling that the model has converged to a solution where further updates no longer yield meaningful improvements.

Evaluation Loss, when computed on a separate validation dataset, assesses the model's generalization capability on unseen data. In our case, since eval_dataset = None. A persistent gap between training and evaluation loss typically indicates overfitting, whereas convergence of both losses suggests effective generalization.

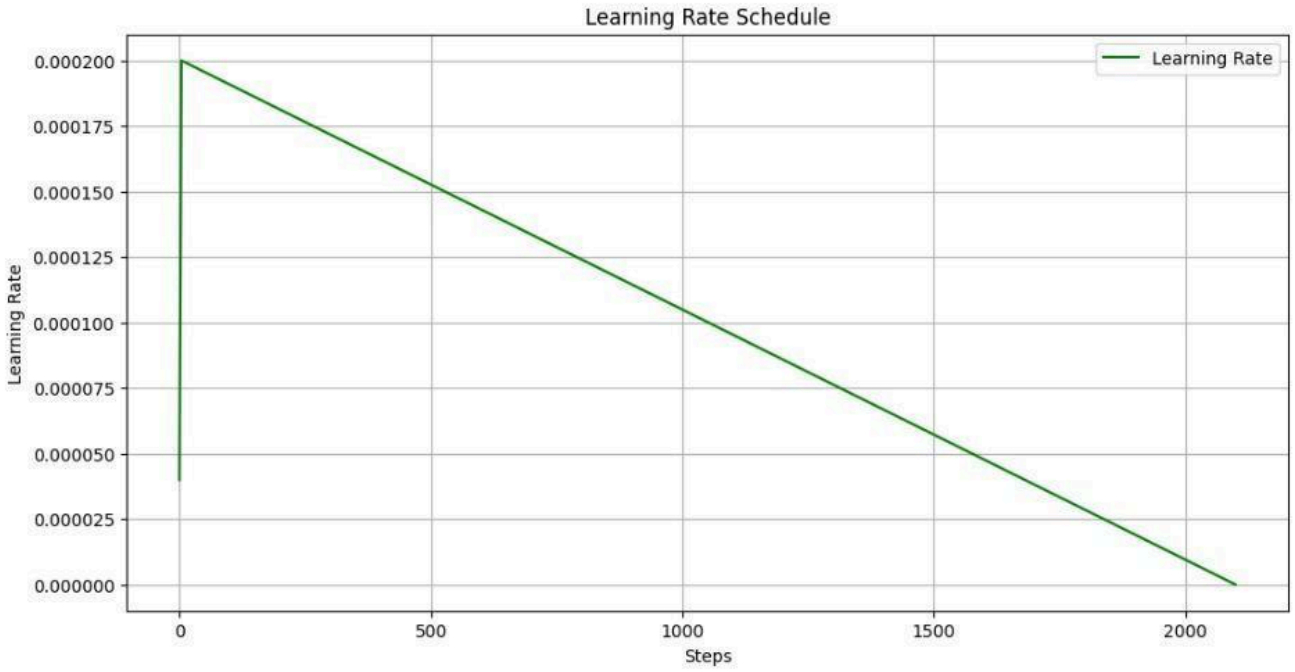


Fig. 8: Learning Rate Schedule

The Learning Rate (Fig. 8) governs the magnitude of parameter updates during optimization and plays a pivotal role in determining the speed and stability of convergence. Here, a linear scheduler is employed to gradually reduce the learning rate over the course of training, starting at an initial value of 2e-4 and decaying linearly to a lower value by the end. This approach ensures efficient training by allowing larger updates early in the process when the model is far from the optimal solution, followed by smaller updates later to fine-tune the parameters precisely. The controlled reduction of the learning rate mitigates the risk of overshooting optimal weights, stabilizing the training process while maintaining convergence efficiency. Early in training, the higher learning rate facilitates rapid adaptation to the overall structure of the training data, enabling significant progress in reducing the training loss.

3.1.5 Input to the Proposed System

The following scanned answer script is submitted as input to the proposed automated grading system. The response addresses the question: "What is the role of algorithms in optimizing computational tasks?" The system evaluates the answer based on a predefined rubric incorporated into the LLM's prompt. The grading criteria are structured as follows:

1. Clarity of Thought (10 points):
Assess how effectively the student communicates their understanding of the problem and solution. The response should demonstrate logical reasoning and a clear articulation of ideas.

2. Accuracy (10 points):
Evaluate the correctness of the concepts, algorithms, and techniques mentioned. This includes verifying the validity of the information provided and ensuring alignment with established principles.

3. Depth of Explanation (10 points):
Measure the depth of the student's understanding by analyzing whether they address edge cases, computational complexities, and trade-offs associated with the algorithms discussed.

4. Structure and Organization (10 points):
Examine the logical flow of the response, ensuring it progresses coherently from the problem statement to the solution. The answer should be well-organized and easy to follow.

5. Language and Grammar (10 points):
Assess the clarity, grammatical correctness, and overall readability of the response. The language should enhance, not hinder, the communication of ideas.

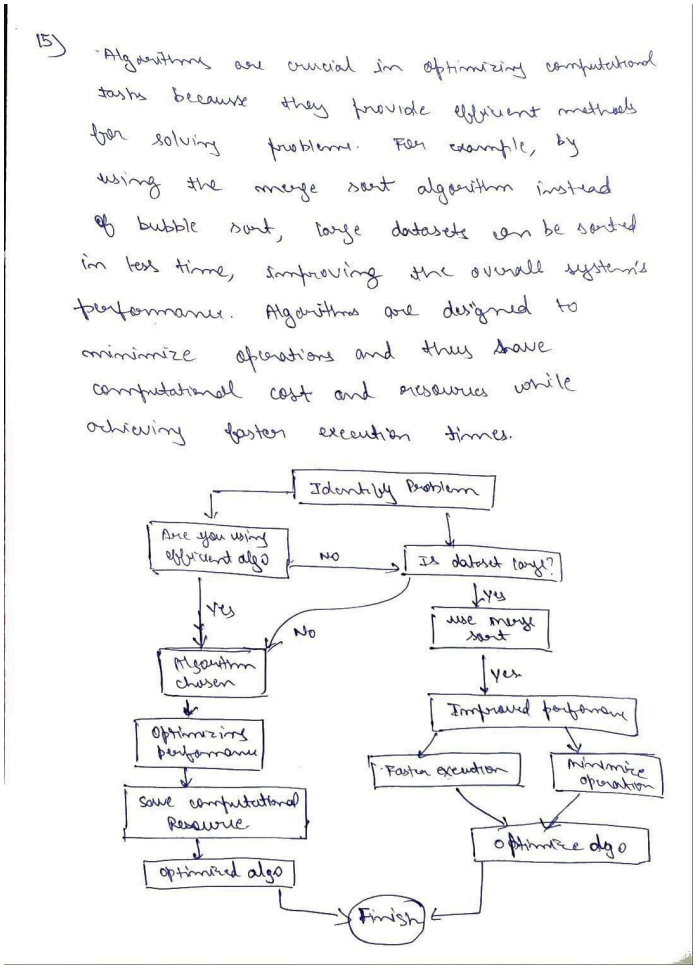


Fig. 9: Sample Input

3.1.6 Output from the Proposed System

The following is the output generated by the proposed system, formatted in markdown. The grading has been performed in accordance with the rubrics outlined alongside the input. The response is parsed, and a JSON object is extracted for structured storage in the database, ensuring efficient record-keeping and retrieval.


```json
{
 "answers": [
  {
   "student_id": "studentA15",
   "marks": 40,
   "rubric_evaluation": {
    "clarity_of_thought": 8,
    "accuracy": 9,
    "depth_of_explanation": 7,
    "structure_and_organization": 8,
    "language_and_grammar": 9
   },
   "explanation": {
        "clarity_of_thought": "The student provides a clear introductory statement about the role of algorithms. However, the explanation could benefit from a more precise definition of 'optimization' in the context of computation. The flowchart explanation is straightforward and easy to follow.",
        "accuracy": "The student accurately describes the core function of algorithms in improving computational efficiency and provides a relevant example (merge sort vs. bubble sort). The flowchart accurately represents a process for algorithmic optimization.",
      "depth_of_explanation": "The explanation touches upon key aspects like minimizing operations and resource saving, but could delve deeper into different optimization techniques (e.g., dynamic programming, caching, profiling). It also lacks discussion about algorithmic complexity (Big O notation) and its impact on performance.",
       "structure_and_organization": "The answer is well-structured, starting with a general explanation, followed by a flowchart description, and finally a detailed explanation of the flowchart. The progression is logical and easy to follow.",
      "language_and_grammar": "The language is clear, concise, and grammatically correct. The writing style is professional and easy to understand."
   }
  }
 ]
}
```

3.2 RESULTS

3.2.1 Comparison of Base Qwen and Fine-Tuned Qwen

We evaluate the performance of Qwen in OCR tasks, both as a base model and as a fine-tuned model specifically optimized for recognizing handwriting from DSA-based answer scripts. The fine-tuned model was trained on a dataset comprising 50 <image, text> pairs of scanned answer scripts and their corresponding transcribed text content.

The results for each configuration are presented in the table below:

Image Name	FT-Qwen	Base-Qwen
A12	Algorithms are crucial for optimizing computational tasks because they provide efficient methods for solving problems. For example, by using the merge sort algorithm instead of bubble sort, large datasets can be sorted in less time, improving the overall system's performance. Algorithms are designed to minimize operations and thus save computational cost and resources while achieving faster execution times.\n\n### Flowchart Description\n\n1. Identify Problem: Start by identifying the problem that needs to be solved.\n2. Are you using efficient algo?\n - If yes, proceed to the next step.\n - If no, determine if the dataset is large.\n3. Is the dataset large?\n - If yes, use merge sort.\n - If no, choose an appropriate algorithm.\n4. Algorithm chosen:\n - Optimize performance.\n - Save computational resource.\n - Optimize algo.\n5. Optimized algo:\n - Fast execution.\n - Minimize operation.\n6. Improved performance:\n - Optimize algo.\n7. Finish\n\n### Flowchart Explanation\n\nThe flowchart outlines a systematic approach to optimize algorithms for computational tasks. It begins by identifying the problem at hand. If an efficient algorithm is already being used, the process moves forward. However, if not, the next step is to determine whether the dataset is large. If the dataset is large, the merge sort algorithm is recommended; otherwise, an appropriate algorithm is chosen based on the specific requirements.\n\nOnce an algorithm is selected, the process optimizes its performance by saving computational resources and further refining the algorithm. This leads to improved performance, which is then optimized again to ensure efficiency. The final step marks the completion of the optimization process.\n\n This flowchart provides a clear and structured method for selecting and	Certainly! Here is the OCR for the provided image. \n\nThe text and flowchart provided pertain to algorithms and their optimization for computational tasks. Algorithms are fundamental tools for enhancing the efficiency of computational tasks, as they provide efficient methods for solving problems. For example, by employing optimized algorithms such as merge sort instead of bubble sort, significant improvements in performance can be achieved, especially when handling large datasets. The primary goal of algorithm design is to reduce the number of operations, conserve computational resources, and ensure faster execution. The flowchart presents a step-by-step process for selecting and optimizing algorithms to improve computational performance.

	optimizing algorithms to enhance computational performance.	
A13	'13) Algorithms serve to optimize computational tasks by breaking down the problem into smaller, manageable components, allowing for more efficient solutions.\nOptimization often involves using algorithms that minimize both time and space complexities.\nFor example, algorithms like A* or Dijkstra\'s algorithm for pathfinding are optimized to calculate the shortest path efficiently without unnecessary calculations.\n\n### Flowchart Description\n\nThe flowchart begins with "Start analysis" and proceeds through several stages:\n\n1. Break Problem: The first step is to break down the problem into smaller, manageable components. If the problem is already manageable, the process moves forward to the next step. If not, the problem is broken down further until it becomes manageable.\n\n2. Minimize Complexity: Once the problem is broken down, the next step is to minimize the complexity of the solution. This involves choosing an appropriate algorithm based on the problem size and requirements.\n\n3. Choose Algorithm: Based on the minimized complexity, the algorithm chosen will be either A* or Dijkstra\'s algorithm. These algorithms are specifically designed to find the shortest path efficiently.\n\n4. Calculate Shortest Path: After choosing the algorithm, the next step is to calculate the shortest path. This involves performing the necessary calculations to determine the optimal route.\n\n5. Avoid Unnecessary Calculations: Throughout the calculation process, efforts are made to avoid unnecessary calculations to ensure efficiency.\n\n6. Optimized Solution: Finally, the optimized solution is chosen, which is the result of the efficient computation of the shortest path.\n\nThe flowchart illustrates the systematic approach to solving complex problems by breaking them down, optimizing the solution, and ensuring efficiency throughout the process.'	The given image talks about algorithms and their optimization. Here is the complete extracted text. \n\nAlgorithms serve to optimize computational tasks by breaking down the problem into smaller, manageable components, allowing for more efficient solutions.\nOptimization often involves using algorithms that minimize both time and space complexities.\nFor example, algorithms like A* or Dijkstra\'s algorithm for pathfinding is optimized to calculate the shortest path efficiently without unnecessary calculations. The flowchart starts with analyzing the problem, then breaks it into smaller parts if needed. Next, it minimizes complexity by selecting the most efficient approach. Based on this, either A* or Dijkstra’s algorithm is chosen to find the shortest path. The calculations focus on avoiding unnecessary steps to improve efficiency, leading to an optimized solution.
A14	'The role of algorithms in optimizing computational tasks is to identify the most efficient way to solve a problem, often by reducing redundant operations. For example, a DFS or BFS algorithm can optimize graph traversal tasks. These algorithms minimize the number of costs and nodes visited and the steps required to reach a goal.\n\n The flowchart in the image outlines a process for optimizing computational	Certainly! 'The role of algorithms in optimizing computational tasks is to identify the most efficient way to solve a problem, often by reducing redundant operations. For example, a DFS or BFS algorithm can optimize graph traversal tasks. These algorithms minimize the number of costs and nodes visited and the steps required to

	tasks using algorithms. It begins with the step of identifying an efficient solution. The next decision point asks if the task involves graph traversal. If the answer is "Yes," the process moves to use either Depth-First Search (DFS) or Breadth-First Search (BFS) algorithms. These algorithms minimize the number of nodes visited and the steps required to reach the goal, thus reducing redundant operations. This leads to completing the optimization process. If the task is not graph traversal, the flowchart suggests considering other algorithms.'	reach a goal.\n\n The flowchart optimizes tasks by first identifying an efficient solution. If graph traversal is required, it selects DFS or BFS to minimize steps and avoid redundancy. Otherwise, it suggests exploring other algorithms for optimization.

Table 1: Comparison of Base Qwen and Fine-Tuned Qwen

The table above demonstrates that the fine-tuned version of Qwen significantly outperforms the base model in several key areas. Specifically, the fine-tuned Qwen not only extracts text from scanned scripts with higher accuracy but also provides precise descriptions of flowcharts and diagrams present in the answer scripts. In contrast, the base version of Qwen tends to summarize content and alter the original descriptions, which can lead to a loss of critical details and contextual accuracy. This highlights the effectiveness of fine-tuning in enhancing the model's ability to handle complex visual and textual elements in grading tasks.

3.2.2 Comparison of Base Gemma and Fine-Tuned Gemma

We assess the performance of Gemma-3-4B in addressing niche Data Structures and Algorithms (DSA) questions, evaluating both its base version and a fine-tuned variant that incorporates domain-specific knowledge of DSA. The models are tasked with providing technically precise and detailed responses to the following questions, showcasing their ability to handle complex subject matter effectively.

Q1. Why are binary heaps generally preferred over balanced binary search trees (BSTs) for implementing priority queues, even though BSTs also support O(logN) insert and delete operations?

Q2. When is it more efficient to use an adjacency list over an adjacency matrix for representing graphs, and how does this decision impact space and time complexity for different graph operations?

Q3. Why can’t we reverse a singly linked list in less than O(N) time, and how does pointer manipulation make it possible without using extra space?

Both sets of responses are then evaluated and judged by GPT-4o (out of 10 marks) , which serves as an impartial benchmark for assessing the quality, accuracy, and depth of the answers generated by each version of the model. This evaluation ensures a reliable comparison of the base and fine-tuned models' capabilities in tackling specialized DSA problems.

Base Gemma	FT Response	4o-Score
Binary heaps are generally preferred over balanced binary search trees for implementing priority queues due to their superior performance characteristics in terms of both insertion and deletion, specifically when retrieving the minimum or maximum element. While balanced BSTs offer O(log N) insertion and deletion, heaps provide O(log N) for both operations, but crucially, they guarantee that the minimum (or maximum) element is always at the root, allowing for constant-time retrieval – a significant advantage for priority queue functionality. Furthermore, heaps typically have simpler implementations and lower overhead than balanced BSTs, making them more efficient in practice, especially when frequent minimum/maximum element access is required, as is typical for priority queue applications; BSTs, on the other hand, require traversal to find the minimum/maximum, adding to their overall time complexity for these operations.	Binary heaps are generally preferred over balanced binary search trees (BSTs) for implementing priority queues because they offer better cache efficiency, simpler implementation, and faster constant factors in operations. Since binary heaps are implemented as arrays, they exhibit better spatial locality and avoid the pointer chasing overhead of BSTs, which rely on dynamically allocated nodes. While both data structures support O(logN) insert and delete operations, binary heaps achieve this with simple "heapify" operations (sift-up and sift-down), whereas balanced BSTs require complex rotations to maintain structural balance. Additionally, heaps guarantee O(1) time complexity for extracting the minimum or maximum element, whereas balanced BSTs require O(logN) for the same operation. These advantages make binary heaps the optimal choice for priority queues in most scenarios.	7 | 9
An adjacency list is more efficient than an adjacency matrix when dealing with sparse graphs, where the number of edges (E) is much smaller than the number of possible edges (V^2). The adjacency list representation requires only O(V+E) space, whereas an adjacency matrix requires O(V^2) space, making it impractical for large, sparse graphs. In terms of time complexity, adjacency lists allow efficient traversal of a node's neighbors in O(V), whereas adjacency matrices require O(V) time even for checking a single node's neighbors. However, adjacency matrices provide O(1) edge existence checks, whereas adjacency lists require O(V) search time. Adjacency lists are preferable for algorithms like DFS and Dijkstra's on sparse graphs, while adjacency matrices are better suited for dense graphs and algorithms requiring frequent edge lookups, such as the Floyd-Warshall algorithm.	Adjacency lists are generally more efficient than adjacency matrices for representing sparse graphs – graphs with relatively few edges compared to the number of vertices – due to their space efficiency. An adjacency matrix requires O(V^2) space, regardless of the actual number of edges, while an adjacency list uses O(V + E) space, where V is the number of vertices and E is the number of edges. This difference in space translates to faster insertion and deletion operations for edges, which are O(1) for adjacency lists versus O(V) for adjacency matrices. Furthermore, operations like finding neighbors (breadth-first search, depth-first search) are typically faster with adjacency lists, having a time complexity of O(V + E) versus O(V^2) for adjacency matrices. Conversely, adjacency matrices excel when dealing with dense graphs or when checking	7 | 8.5

	for the existence of an edge between two vertices – this is O(1) for matrices compared to O(E) for lists.	
Reversing a singly linked list takes O(n) time because each node must be visited once, and its 'next' pointer updated. This is achieved through iterative or recursive approaches that traverse the entire list while modifying pointers efficiently. Extra space usage depends on the algorithm; iterative solutions typically require constant additional space, whereas recursion uses call stack memory, which may vary depending on implementation details but remains within O(depth of recursion).	Reversing a singly linked list in less than O(N) time is not possible because each node contains only a reference to the next node, requiring us to visit every node at least once to modify the links. Unlike arrays, where elements can be accessed in constant time using indexing, a singly linked list must be traversed sequentially since it lacks backward pointers. Consequently, any in-place reversal must iterate through all N nodes, making O(N) the lower bound for time complexity. The reversal process can be done without extra space using pointer manipulation. By maintaining three pointers—prev, current, and next—we iteratively reverse the next pointers of each node. Initially, prev is set to NULL, and current starts at the head. In each iteration, next stores current->next, then current->next is updated to point to prev. The prev and current pointers are then shifted forward, progressing through the list until current becomes NULL, at which point prev holds the new head. This iterative approach modifies links in place, ensuring O(N) time complexity with O(1) auxiliary space, making it the most efficient method for reversing a singly linked list.	6 | 9

Table 2: Comparison of Standard Gemma and Fine-Tuned Gemma

3.2.3 Comparison of Proposed System and GPT-4o

We compared the graded outputs of our proposed system with those of the state-of-the-art GPT-4o, a proprietary model developed by OpenAI with a parameter size of 145 billion. In contrast, our proposed model has a combined parameter size of 7 billion. Both systems, along with responses graded by CS Faculty were evaluated on three test images, which are available in the Appendix.

The results are presented below:

Image: A12.jpeg

Proposed System Response:

answers:
- student_id: "studentA12"
  marks: 40
  rubric_evaluation:
    clarity_of_thought: 8
    accuracy: 9
    depth_of_explanation: 8
    structure_and_organization: 9
    language_and_grammar: 8
  explanation:
    clarity_of_thought: "The student clearly explains the basic role of algorithms – finding the most efficient solution. The explanation is straightforward and easy to understand."
    accuracy: "The student accurately describes the benefit of using efficient algorithms and provides a relevant example (quicksort vs. bubble sort). The mention of time complexity is also correct."
    depth_of_explanation: "The student could have elaborated further on *why* efficient algorithms are important in real-world scenarios. Discussing different algorithmic complexities (e.g., O(n), O(log n)) and how they relate to resource usage would have added depth. Furthermore, mentioning different types of problems where algorithms are applied (beyond sorting) would be beneficial."
    structure_and_organization: "The answer is well-structured. It starts with a general statement, provides an example, and then describes the flowchart effectively. The flow is logical and easy to follow."
    language_and_grammar: "The language is clear, concise, and grammatically correct. The explanation is professional and well-written."

GPT-4o Response:

answers:
- student_id: "A12"
  marks: 40
  rubric_evaluation:
    clarity_of_thought: 8
    accuracy: 8
    depth_of_explanation: 8
    structure_and_organization: 8
    language_and_grammar: 8
  explanation:
    clarity_of_thought: "The response provides a clear explanation of how algorithms optimize computational tasks. The example of sorting algorithms effectively illustrates the concept."
    accuracy: "The answer correctly mentions time complexity differences among sorting algorithms. However, a deeper explanation of why quicksort and merge sort are more efficient than bubble sort could

strengthen the response."

depth_of_explanation: "The discussion covers key aspects but lacks a comparison of best, worst, and average cases for the sorting algorithms mentioned."

structure_and_organization: "The flowchart presents a structured approach, but the decision-making process could be made clearer, particularly when selecting sorting algorithms."

language_and_grammar: "The writing is mostly correct, with a few minor phrasing issues. Some sentences could be more concise for better readability."

CS Faculty Response:

answers:
- student_id: "A13"
  marks: 39
  rubric_evaluation:
    clarity_of_thought: 9
    accuracy: 8
    depth_of_explanation: 4
    structure_and_organization: 9
    language_and_grammar: 9
  explanation:
    clarity_of_thought: "Student has understood the concept on the role of algorithms and explained in an unambiguous manner."
    accuracy: "Few more points would have made the answer more precise"
    depth_of_explanation: "No detailed explanation is given on the points"
    structure_and_organization: "The figure gives more clarity"
    language_and_grammar: "Language is good, grammatically correct"

Image: A13.jpeg

Proposed System Response:

answers:
- student_id: "studentA13"
  marks: 42
  rubric_evaluation:
    clarity_of_thought: 9
    accuracy: 8
    depth_of_explanation: 7

structure_and_organization: 8

language_and_grammar: 9

explanation:

clarity_of_thought: "The student clearly states the fundamental role of algorithms in breaking down problems and achieving efficient solutions. The initial sentence is concise and easy to understand."

accuracy: "The explanation of algorithms optimizing for time and space complexity is accurate. The examples of A* and Dijkstra's are relevant and demonstrate the concept well. However, a more detailed discussion of different complexity classes (e.g., O(n), O(log n)) could improve accuracy."

depth_of_explanation: "The student provides a basic explanation of the process but lacks depth. They don't delve into the reasons *why* certain algorithms are chosen, or the trade-offs involved in selecting one over another. Discussing factors like data structure choice would have added depth."

structure_and_organization: "The answer is logically structured, following the flow of the flowchart description. The numbered steps clearly outline the process. However, it could benefit from a brief introductory paragraph summarizing the overall process before detailing each step."

language_and_grammar: "The language used is clear, professional, and grammatically correct. The writing is easily readable and understandable."

GPT-4o Response:

answers:

- student_id: "A13"

marks: 39

rubric_evaluation:

clarity_of_thought: 8

accuracy: 9

depth_of_explanation: 7

structure_and_organization: 8

language_and_grammar: 7

explanation:

clarity_of_thought: "The explanation effectively conveys the role of algorithms in optimization. However, the explanation could be more refined for clarity, particularly in structuring the thought process more precisely."

accuracy: "The student correctly explains key concepts like breaking down problems, optimization, and pathfinding with A* and Dijkstra's algorithm. However, some statements could be more rigorously defined."

depth_of_explanation: "The answer provides a solid foundation but lacks deeper discussions on trade-offs, edge cases, and how different algorithms perform under various constraints."

structure_and_organization: "The answer is logically structured with a combination of explanation and a flowchart. However, the flowchart could be slightly clearer in its transitions."

language_and_grammar: "The language is generally understandable, but minor grammar and syntax issues affect readability."

CS Faculty Response:

answers:
- student_id: "A13"
  marks: 29
  rubric_evaluation:
    clarity_of_thought: 5
    accuracy: 6
    depth_of_explanation: 4
    structure_and_organization: 7
    language_and_grammar: 7
  explanation:
    clarity_of_thought: "Student has written only explanation of the question. Example taken is fine"
    accuracy: "Few more points would have made the answer more precise. Example adds some clarity"
    depth_of_explanation: "No detailed explanation is given on the points"
    structure_and_organization: "The figure gives little clarity but can be made more clear"
    language_and_grammar: "Language is moderate, grammatically correct"

Image: A14.jpeg

Proposed System Response:

answers:
- student_id: "studentA14"
  marks: 40
  rubric_evaluation:
    clarity_of_thought: 8
    accuracy: 7
    depth_of_explanation: 8
    structure_and_organization: 7
    language_and_grammar: 8
  explanation:
    clarity_of_thought: "The student's initial explanation is reasonably clear in stating that algorithms optimize by finding efficient solutions and reducing redundant operations. However, the subsequent description of a flowchart feels somewhat tangential and adds unnecessary complexity without directly contributing to a deeper understanding of the core concept."

accuracy: "The explanation of DFS/BFS is accurate in its role, but the overall description of the optimization process is a bit simplistic. It accurately reflects the *intent* of algorithms but lacks nuance regarding algorithmic complexity and different optimization strategies."

depth_of_explanation: "The student demonstrates a basic understanding of the role of algorithms, mentioning reducing operations and node visits. However, it lacks discussion of algorithmic complexity (e.g., Big O notation), different algorithm families (sorting, searching, dynamic programming), or trade-offs between efficiency and other factors like memory usage. It could also benefit from discussing why certain algorithms are better suited for particular problem types."

structure_and_organization: "The flow chart description provides a loose structure, but it's not entirely coherent with the main explanation. The transition between describing algorithms and then explaining a flowchart is a little disjointed. A clearer connection between the concepts would improve the structure."

language_and_grammar: "The language is generally clear and grammatically correct. The writing is easy to understand."

GPT-4o Response:

answers:
- student_id: "A14"
  marks: 42
  rubric_evaluation:
    clarity_of_thought: 8
    accuracy: 9
    depth_of_explanation: 8
    structure_and_organization: 9
    language_and_grammar: 8
  explanation:

clarity_of_thought: "The response effectively explains the role of algorithms in optimizing computational tasks, with a focus on graph traversal."

accuracy: "The mention of DFS and BFS as efficient graph traversal algorithms is correct, though a brief comparison of their use cases (e.g., BFS for shortest path, DFS for deeper exploration) would enhance accuracy."

depth_of_explanation: "The explanation touches on optimization principles but lacks deeper insights into complexity analysis or specific cases where DFS or BFS might be preferable."

structure_and_organization: "The flowchart is well-organized and presents a logical decision-making process. However, the distinction between BFS and DFS could be made clearer."

language_and_grammar: "The response is generally well-written, with minor grammatical errors that slightly impact readability."

CS Faculty Response:

answers:
- student_id: "A14"
  marks: 26

rubric_evaluation:
clarity_of_thought: 5
accuracy: 5
depth_of_explanation: 4
structure_and_organization: 5
language_and_grammar: 7
explanation:
clarity_of_thought: "Student has written content which is not related to the question. Example taken is fine"
accuracy: "More relevant points would have made the answer more precise. Example adds some clarity"
depth_of_explanation: "No detailed explanation is given on the points"
structure_and_organization: "The figure added is good but not fully correct"
language_and_grammar: "Language is moderate, grammatically correct"

As demonstrated by the outputs above, the proposed model matches the performance of the proprietary 145B-parameter model while maintaining high accuracy and reliability in grading computer science answer scripts. Additionally, being open-source, the fast and efficient 7B-parameter grader offers a transparent and accessible solution, making it a highly attractive and capable alternative for widespread adoption.

Chapter 4

Discussion

4.1 CONCLUSION

The integration of Vision-Language Models (VLMs) with fine-tuned Large Language Models (LLMs) introduces a transformative approach to computational problem-solving, particularly in the domain of Data Structures and Algorithms (DSA). This framework leverages structured JSON files to represent domain-specific knowledge, ensuring precise and context-aware responses. By incorporating Qwen-VL models for advanced visual processing, the system facilitates seamless interaction between textual, visual, and structured data, enabling efficient interpretation of handwritten diagrams, pseudocode, and algorithmic queries. This multimodal integration significantly enhances the capabilities of coding assistants, educational tools, and automated assessment systems by providing more accurate and contextually relevant evaluations.

A key innovation of the proposed framework is the strategic use of prompt engineering to refine grading accuracy and adaptability. By carefully crafting prompts, the system dynamically retrieves and integrates relevant information, ensuring comprehensive assessments even for complex algorithmic solutions. This approach enables precise evaluation without relying on external retrieval mechanisms. Additionally, iterative prompt refinement enhances adaptability by improving response quality based on real-world performance metrics, making the system highly effective for large-scale academic evaluation.

4.2 FUTURE SCOPE

Future research will focus on optimizing the fine-tuning process to enhance response precision and computational efficiency. Expanding the model's capabilities to address a wider range of computational problems remains a key objective. Additionally, improving the integration between visual and textual modalities through sophisticated pre-processing and contextual understanding techniques will further refine performance. Investigating adaptive learning mechanisms to dynamically adjust grading criteria based on evolving educational requirements will ensure long-term scalability. As advancements in multimodal AI continue to evolve, this research contributes to the development of robust, intelligent frameworks for technical education and automated assessment, setting a foundation for next-generation AI-driven problem-solving solutions.

An area of particular interest is the integration of Retrieval-Augmented Generation (RAG) to enhance the reasoning capabilities of the grading model. RAG dynamically retrieves relevant grading criteria and exemplar answers, ensuring that grading decisions are well-grounded in established evaluation standards while maintaining contextual accuracy and fairness. This methodology significantly improves grading consistency, interpretability, and adaptability, particularly for handwritten responses. The process is structured into three key phases: retrieval, augmentation, and generation, each contributing to a structured and accurate grading workflow.

Retrieval-Augmented Generation (RAG): The incorporation of Retrieval-Augmented Generation (RAG) in automated grading systems can significantly enhance the accuracy, consistency, and interpretability of assessments by integrating retrieval mechanisms with Vision-Language Models (VLMs). RAG ensures that evaluations align with established grading criteria by leveraging historical grading data, rubric-based assessments, and exemplar responses.

Types of RAG for Enhancing Automated Grading:

Document-Retrieval RAG utilizes structured databases and indexed repositories to retrieve relevant grading rubrics, past student responses, and feedback. By leveraging predefined grading criteria and exemplar answers, this approach ensures that assessments align with established evaluation frameworks, improving the consistency and fairness of grading.

Embedding-Based RAG applies high-dimensional vector representations to student responses and historical grading data. This technique enables semantic similarity searches, allowing the system to compare new answers with previously graded submissions. By capturing contextual nuances, embedding-based retrieval enhances grading precision and minimizes discrepancies.

Hybrid RAG combines traditional keyword-based retrieval with embedding-based techniques to refine search accuracy. This approach ensures that assessments incorporate both explicit grading criteria and implicit contextual understanding. By balancing structured rule-based evaluation with semantic flexibility, hybrid RAG enhances the adaptability of automated grading models.

Multi-Modal RAG integrates text and visual data from Vision-Language Models (VLMs) to evaluate diverse response formats, including handwritten submissions, diagrams, and mathematical notations. By extending retrieval capabilities beyond textual inputs, this method ensures comprehensive assessment coverage across various subject areas and response types.

Contextual RAG dynamically adjusts retrieval strategies based on the grading scenario. By analyzing question complexity, response patterns, and assessment context, this approach refines search parameters in real time. This adaptability ensures that grading remains aligned with evolving evaluation standards, improving both accuracy and interpretability.

RAG-Enabled Grading Pipeline:

Retrieval Phase: The system queries a structured database containing manually graded responses, domain-specific rubrics, and standardized evaluation guidelines. This ensures alignment with past assessments and mitigates grader-specific biases.

Augmentation Phase: Retrieved references are incorporated into Google Gemma-3-4B-Instruct to provide contextual grounding, enhancing the model's ability to assess diverse handwriting styles, complex mathematical expressions, and diagrammatic responses with higher precision.

Generation Phase: A context-aware grading decision is synthesized based on structured evaluation metrics, ensuring fairness, explainability, and adherence to pedagogical standards.

Advantages of RAG:

1. The retrieval mechanism provides explicit grading justifications, enhancing transparency and explainability in automated scoring.

2. Unlike traditional fine-tuning approaches that require extensive retraining, RAG dynamically retrieves relevant knowledge, reducing computational overhead while supporting large-scale assessments such as MOOCs and institutional exams.

3. RAG enables more precise evaluation of handwritten responses, including complex equations and structured diagrammatic answers, making it particularly effective for STEM-based assessments.

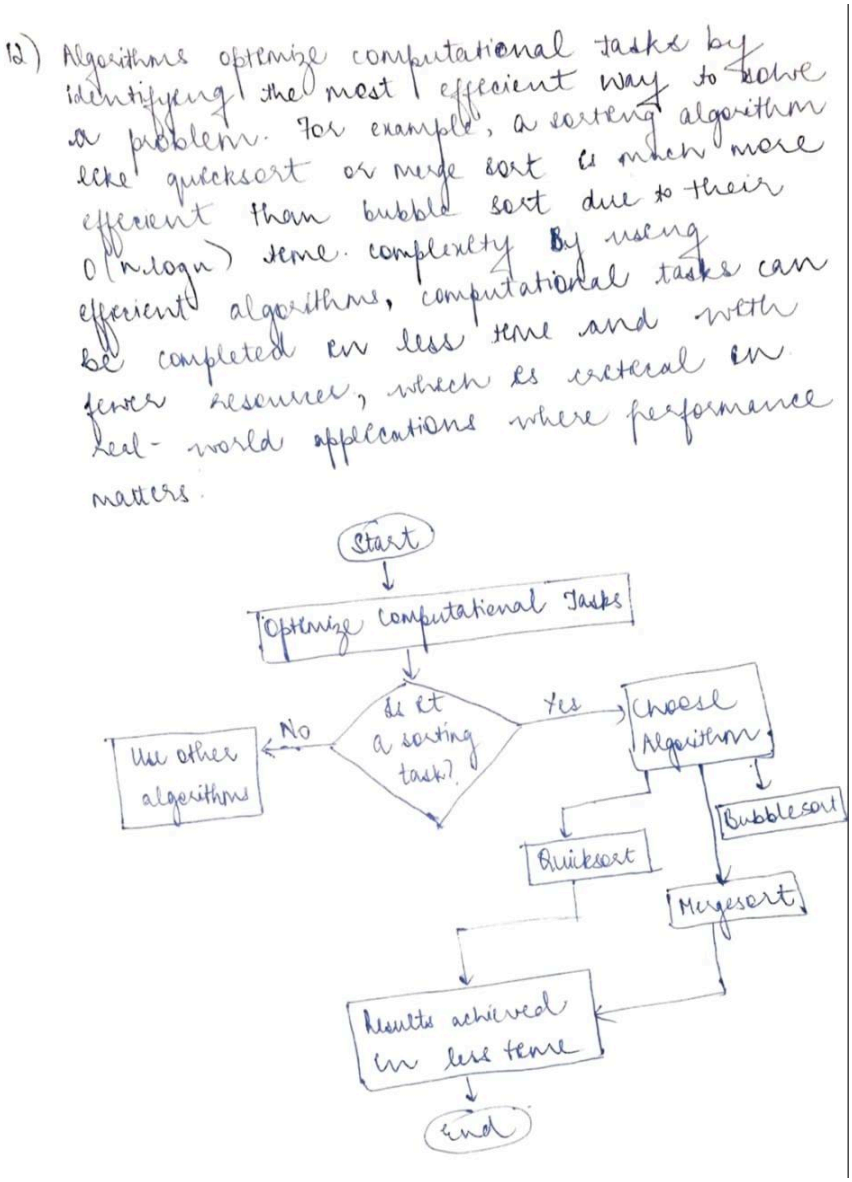
Appendix 1

Datasets

The Handwritten Answer Script Dataset comprises 40 training samples and 5 samples reserved for testing, with three distinct questions, each having 15 unique answer scripts. These answer scripts vary in quality, ranging from excellent to poor responses, ensuring diversity in the dataset. The answer scripts for each of the question are as follows :

Question 1: What is the role of algorithms in optimizing computational tasks?

A12.jpg (Test)

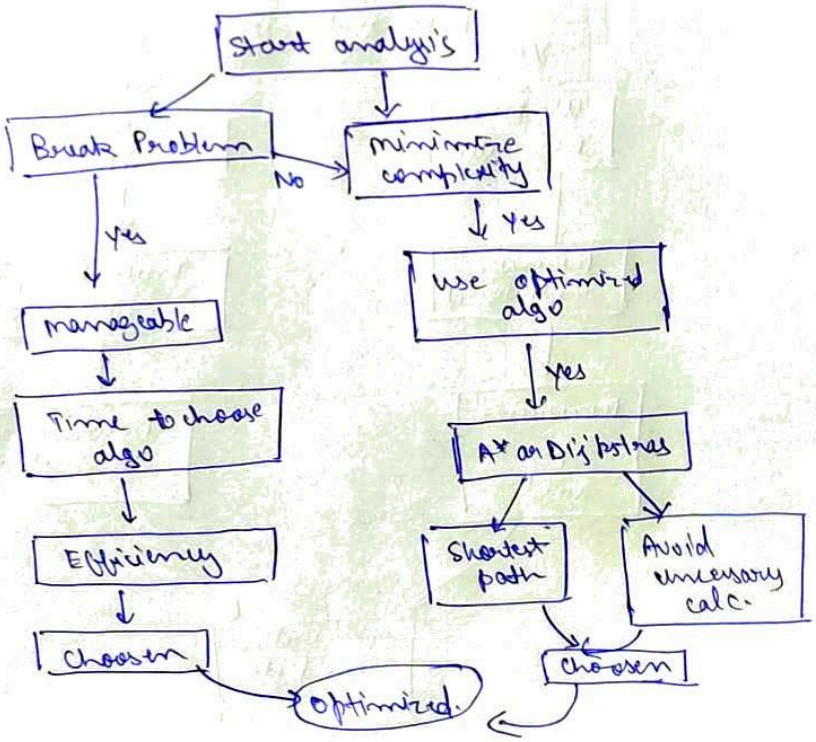


A13.jpg (Test)

13) Algorithms serve to optimize computational tasks by breaking down the problem into smaller, manageable components, allowing for more efficient solutions.

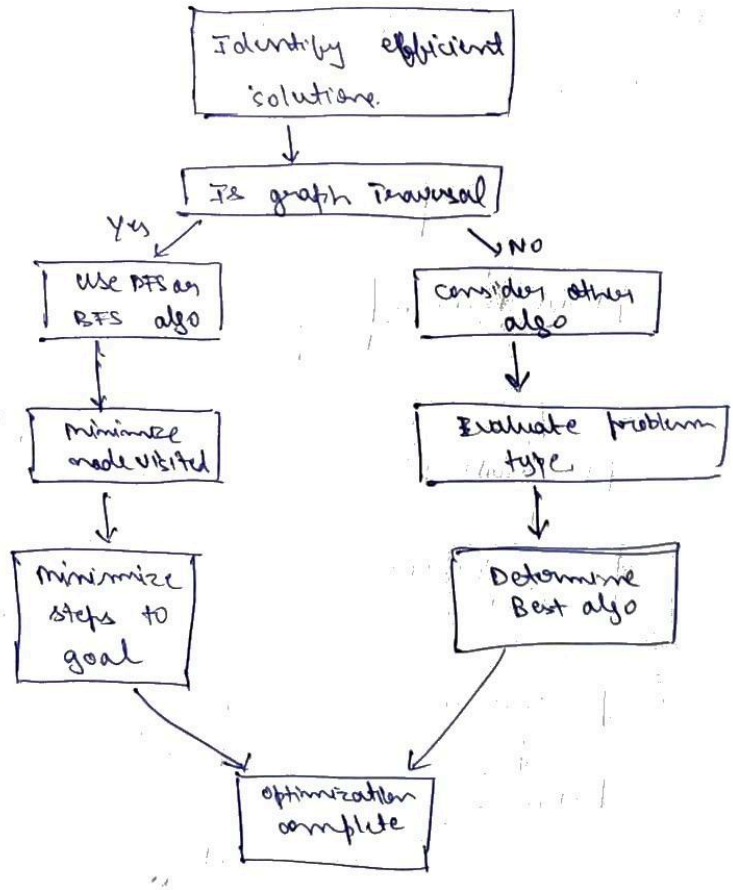
Optimization often involves using algorithms that minimize both time and space complexities.

For example, algorithms like A* or Dijkstra's algorithm for pathfinding are optimized to calculate the shortest path efficiently without unnecessary calculations.



A14.jpg (Test)

14) The role of algorithms in optimizing computational tasks is to indentify the most efficient way to solve a problem, often by reducing redundant operations. For eg: a DFS or BFS algorithm can optimize graph traversal tasks. These algorithms minimise the no-of costs and nodes visited and the steps required to reach goal.

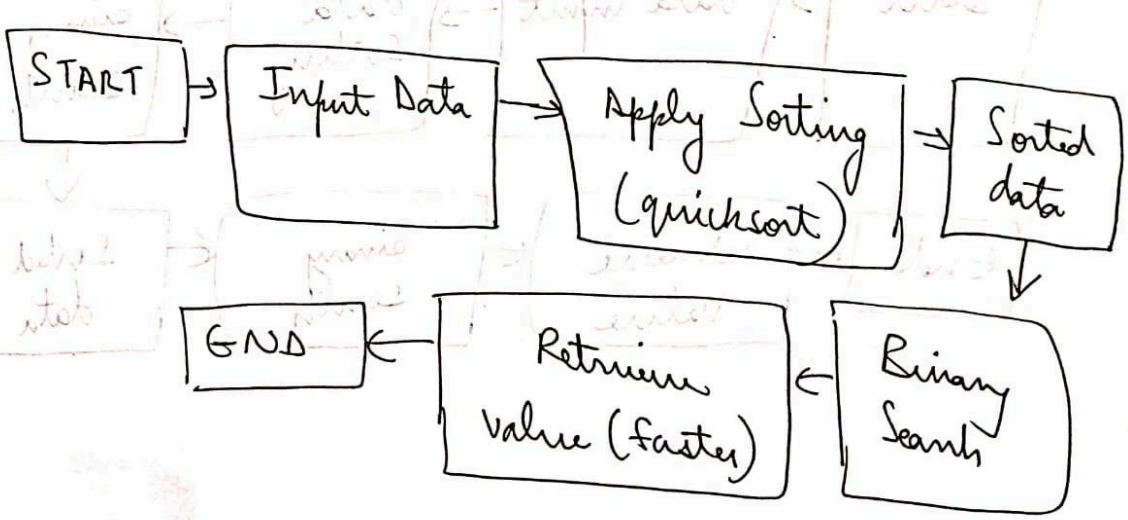


Question 2: How do sorting algorithms impact the performance of data retrieval?

V11.jpg

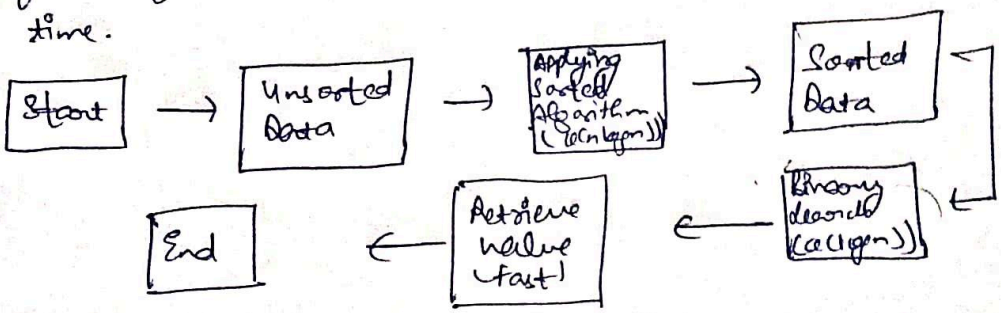
11

Sorting algorithms enhance data retrieval by structuring the data in way that supports efficient search methods. For example, after sorting binary search can be performed, which significantly reduces the number of comparisons compared to linear search. While sorting the data involves an initial overhead eg: O(nlogn) for quicksort or mergesort it ultimately leads to faster retrieval times for large datasets.



V5.jpg

5 The performance of data retrieval is significantly impacted by sorting algorithms. When data is unsorted, retrieving specific values may require a linear search, which is inefficient for large datasets. Sorting the data beforehand allows faster retrieval using binary search or other algorithms that operate on sorted data. The time complexity of the sorting algorithm (eg. O(nlogn) for merge sort) must be factored into the overall retrieval time.

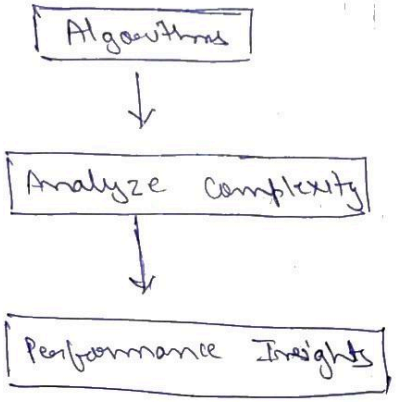


Question 3: What is the importance of analyzing algorithm complexity?

M3.jpg

3) Analyzing algorithm is crucial for understanding the scalability and efficiency of an algorithm. It allows us to identify potential performance bottlenecks when the input size grows and optimize them them accordingly.

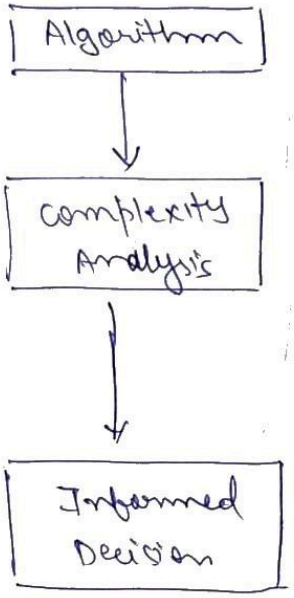
The analysis helps developers select the best algorithm for a given problem based on time and space constraints.



M4.jpg

4) Understanding algorithms complexity helps in evaluating the trade-offs between different algorithms. By analyzing their time and space complexity, we can make informed decisions about which algorithm to use it in a given context.

This is essential for ensuring that we choose an optimal solution that balances performance with resource usage.



The QA Dataset consists of 700 training samples, structured as question-answer pairs covering a wide range of topics in Data Structures and Algorithms (DSA). Below are some representative samples from the dataset:


```
{
 "questions": [
  {
   "question": "What is the time complexity of the Fast Fourier Transform (FFT) algorithm?",
   "answer": "The Fast Fourier Transform (FFT) algorithm has a time complexity of O(n log n), where n is the number of points in the input. This efficiency makes it widely used in signal processing, polynomial multiplication, and other applications requiring frequency domain analysis."
  },
  {
   "question": "How does the Knuth-Morris-Pratt (KMP) algorithm improve string matching?",
   "answer": "The Knuth-Morris-Pratt (KMP) algorithm improves string matching by preprocessing the pattern to create a partial match table (prefix function). This table helps avoid redundant comparisons, reducing the time complexity to O(n + m), where n is the text length and m is the pattern length."
  },
  {
   "question": "What is the purpose of the A* search algorithm?",
   "answer": "The A* search algorithm is used for finding the shortest path between two nodes in a graph. It combines the advantages of Dijkstra's algorithm (guaranteed shortest path) and greedy best-first search (efficiency) by using a heuristic function to estimate the cost to the goal."
  },
  {
   "question": "Explain the concept of dynamic programming in algorithm design.",
   "answer": "Dynamic programming is a method for solving complex problems by breaking them into simpler subproblems. It stores the results of subproblems to avoid redundant computations, optimizing both time and space. Examples include the Fibonacci sequence and the knapsack problem."
  },
  {
   "question": "What is the difference between greedy algorithms and dynamic programming?",
   "answer": "Greedy algorithms make locally optimal choices at each step, hoping to find a global optimum, but they may not always succeed. Dynamic programming, on the other hand, considers all possible choices and uses memoization to ensure an optimal solution."
  },
  {
   "question": "How does the Bellman-Ford algorithm handle negative weight edges?",
   "answer": "The Bellman-Ford algorithm handles negative weight edges by relaxing all edges repeatedly (up to V-1 times, where V is the number of vertices). It can also detect negative weight cycles, making it suitable for graphs where Dijkstra's algorithm fails."
  },
  {
   "question": "What is the significance of the P vs NP problem in computer science?",
   "answer": "The P vs NP problem asks whether every problem whose solution can be verified quickly (in polynomial time) can also be solved quickly. It is one of the most important unsolved problems in computer science, with implications for cryptography, optimization, and computational theory."
  },
  {
   "question": "What is the role of heuristics in the A* algorithm?",
   "answer": "Heuristics in the A* algorithm provide an estimate of the cost to reach the goal from a given node. A good heuristic balances accuracy and computational efficiency, guiding the search toward the most promising paths and improving overall performance."
```

```
  },
  {
   "question": "How does the Floyd-Warshall algorithm work?",
   "answer": "The Floyd-Warshall algorithm computes the shortest paths between all pairs of vertices in a weighted graph. It uses dynamic programming to iteratively improve path estimates, handling both positive and negative edge weights (but no negative cycles)."
  },
  {
   "question": "What is the difference between BFS and DFS in graph traversal?",
   "answer": "BFS (Breadth-First Search) explores nodes level by level, using a queue, and is ideal for finding the shortest path in unweighted graphs. DFS (Depth-First Search) explores as far as possible along each branch before backtracking, using a stack, and is useful for topological sorting and cycle detection."
  },
  {
   "question": "What is the importance of amortized analysis in algorithm design?",
   "answer": "Amortized analysis provides a realistic average-case performance of algorithms over a sequence of operations, even if some operations are expensive. It is commonly used to analyze dynamic arrays, hash tables, and disjoint-set data structures."
  },
  {
   "question": "How does the Rabin-Karp algorithm work for string matching?",
   "answer": "The Rabin-Karp algorithm uses hashing to find any of a set of pattern strings in a text. It computes hash values for the pattern and text substrings, comparing them to identify potential matches. Its average-case time complexity is O(n + m)."
  },
  {
   "question": "What is the purpose of the suffix array data structure?",
   "answer": "A suffix array is a data structure that stores all suffixes of a string in sorted order. It is used for efficient string searching, longest common substring problems, and other text-processing tasks, often outperforming suffix trees in practice."
  },
  {
   "question": "Explain the concept of NP-completeness.",
   "answer": "A problem is NP-complete if it is in NP (solutions can be verified quickly) and every problem in NP can be reduced to it in polynomial time. NP-complete problems, like the traveling salesman problem, are believed to have no efficient solutions."
  },
  {
   "question": "What is the role of randomization in the QuickSort algorithm?",
   "answer": "Randomization in QuickSort involves choosing a pivot element randomly to avoid worst-case scenarios (e.g., already sorted input). This ensures an average-case time complexity of O(n log n) and improves practical performance."
  },
  {
   "question": "How does the Hopcroft-Karp algorithm improve bipartite matching?",
   "answer": "The Hopcroft-Karp algorithm finds maximum bipartite matching in O(√V * E) time by using BFS to construct layered graphs and DFS to find augmenting paths. It is more efficient than the standard DFS-based approach."
  },
  {
   "question": "What is the significance of the master theorem in algorithm analysis?",
   "answer": "The master theorem provides a way to solve recurrence relations of the form T(n) = aT(n/b) + f(n), common in divide-and-conquer algorithms. It simplifies the analysis of algorithms like MergeSort and QuickSort by providing their time complexity directly."
  },
  {
```

"question": "How does the Johnson's algorithm work for all-pairs shortest paths?",
"answer": "Johnson's algorithm computes all-pairs shortest paths by reweighting edges to eliminate negative weights, running Dijkstra's algorithm from each vertex, and then reversing the reweighting. It is efficient for sparse graphs with negative edges."
},

Appendix 2
AutoGrader Source Code

```
import os
import random
from IPython.display import Image, display

def display_random_image(folder_path):
    """
    Choose and display a random image from the specified folder.

    Parameters:
    folder_path (str): Path to the folder containing images

    Returns:
    str: Name of the displayed image file
    """
    # Get list of files in the folder
    files = os.listdir(folder_path)

    # Filter for common image file extensions
    image_extensions = ['.jpg', '.jpeg', '.png', '.gif', '.bmp', '.tiff']
    image_files = [f for f in files if os.path.splitext(f.lower())[1] in image_extensions]

    if not image_files:
        print("No image files found in the specified folder.")
        return None

    # Choose a random image
    random_image = random.choice(image_files)

    # Display the image
    image_path = os.path.join(folder_path, random_image)
    display(Image(filename=image_path))

    return random_image

# Example usage:
```

```
selected_image = display_random_image('test-images')
print(f"Displayed: {selected_image}")
from transformers import Qwen2_5_VLForConditionalGeneration, AutoTokenizer, AutoProcessor
from qwen_vl_utils import process_vision_info

# default: Load the model on the available device(s)
model = Qwen2_5_VLForConditionalGeneration.from_pretrained(
    "aneesh-sathe/qwen2.5-3b-ft", torch_dtype="auto", device_map="auto"
)

processor = AutoProcessor.from_pretrained("aneesh-sathe/qwen2.5-3b-ft")



messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": f"test-images/{selected_image}",
            },
             {"type": "text", "text": "Perform OCR on the uploaded image and extract all text content exactly as it appears. Also describe the flowchart in the image in a paragraph format. Be descriptive with your explaination."},
        ],
    }
]

# Preparation for inference
text = processor.apply_chat_template(
    messages, tokenize=False, add_generation_prompt=True
)
image_inputs, video_inputs = process_vision_info(messages)
inputs = processor(
    text=[text],
    images=image_inputs,
    videos=video_inputs,
    padding=True,
    return_tensors="pt",
)
inputs = inputs.to("cuda")
```

```
# Inference: Generation of the output
generated_ids = model.generate(**inputs, max_new_tokens=512)
generated_ids_trimmed = [
    out_ids[len(in_ids) :] for in_ids, out_ids in zip(inputs.input_ids, generated_ids)
]
output_text = processor.batch_decode(
    generated_ids_trimmed, skip_special_tokens=True, clean_up_tokenization_spaces=False
)
print(output_text)


ext_text = "QUESTION: What is the role of algorithms in optimizing computational tasks? ANSWER:" + "
".join(output_text)


import torch
from transformers import AutoTokenizer, Gemma3ForCausalLM


ckpt = "aneesh-sathe/gemma-3-4b-ft"
gemma_model = Gemma3ForCausalLM.from_pretrained(
    ckpt, torch_dtype=torch.bfloat16, device_map="auto"
)
gemma_tokenizer = AutoTokenizer.from_pretrained(ckpt)



GEMMA_SYS_PROMPT = """
You are a professional grader specializing in Data Structures and Algorithms (DSA). Your task is to evaluate student answers on a scale of 0 to 50 based on five rubrics:


1. Clarity of Thought (10 points): Evaluate how clearly the student conveys their understanding of the problem and solution.
2. Accuracy (10 points): Assess the correctness of the concepts, algorithms, and techniques used.
3. Depth of Explanation (10 points): Determine the depth of understanding, covering edge cases, complexities, and trade-offs.
4. Structure and Organization (10 points): Check if the answer is logically structured, progressing coherently from problem to solution.
5. Language and Grammar (10 points): Evaluate the clarity, grammar, and readability of the response.


You must output a JSON object in the following format:


{
  "answers": [
```

```
  {
   "student_id": "<student_id>",
   "marks": <total_marks>,
   "rubric_evaluation": {
    "clarity_of_thought": <marks_out_of_10>,
    "accuracy": <marks_out_of_10>,
    "depth_of_explanation": <marks_out_of_10>,
    "structure_and_organization": <marks_out_of_10>,
    "language_and_grammar": <marks_out_of_10>
   },
   "explanation": {
    "clarity_of_thought": "<explanation_of_clarity>",
    "accuracy": "<explanation_of_accuracy>",
    "depth_of_explanation": "<explanation_of_depth>",
    "structure_and_organization": "<explanation_of_structure>",
    "language_and_grammar": "<explanation_of_language>"
   }
  }
 ]
}

Grading Rules:
- Assign marks proportionate to performance in each rubric.
- Provide a rationale behind each rubric score in the explanation field, addressing both strengths and areas for improvement.
- Ensure fairness, consistency, and impartiality in grading.
- Do not exceed 50 marks in total.

Your output must strictly follow the JSON structure provided.
"""

gemma_messages = [
  [
    {
      "role": "system",
      "content": [{"type": "text", "text": GEMMA_SYS_PROMPT},]
    },
    {
      "role": "user",
      "content": [{"type": "text", "text": ext_text},]
```

```
        },
    ],
]
gemma_inputs = gemma_tokenizer.apply_chat_template(
    gemma_messages, add_generation_prompt=True, tokenize=True,
    return_dict=True, return_tensors="pt"
).to(gemma_model.device)

gemma_input_len = gemma_inputs["input_ids"].shape[-1]

gemma_generation = gemma_model.generate(**gemma_inputs, max_new_tokens=256, do_sample=True)
gemma_generation = gemma_generation[0][gemma_input_len:]

gemma_decoded = gemma_tokenizer.decode(gemma_generation, skip_special_tokens=True)
print(gemma_decoded)

import json
import os
import re
from IPython.display import display, HTML

def extract_and_append_json(llm_response, file_path="llm_responses.json"):
    """
    Extracts JSON from raw LLM output and appends it to a file in a Jupyter environment.
    Creates parent directories if they don't exist.

    Args:
        llm_response (str): Raw text output from an LLM, potentially containing JSON
        file_path (str): Path to the output file

    Returns:
        bool: True if operation was successful, False otherwise
    """
    # Extract JSON from the LLM response
    extracted_json = extract_json_from_text(llm_response)

    if not extracted_json:
        display(HTML("<div style='color: red;'>Error: No valid JSON found in the response</div>"))
        return False
```

```
# Parse the extracted JSON string into a Python dictionary
try:
    json_data = json.loads(extracted_json)
except json.JSONDecodeError as e:
    display(HTML(f"<div style='color: red;'>Error parsing JSON: {str(e)}</div>"))
    display(HTML(f"<div style='color: gray;'>Extracted text: {extracted_json[:100]}...</div>"))
    return False

# Create directory if it doesn't exist
directory = os.path.dirname(file_path)
if directory and not os.path.exists(directory):
    try:
        os.makedirs(directory)
        display(HTML(f"<div style='color: blue;'>Created directory: {directory}</div>"))
    except Exception as e:
        display(HTML(f"<div style='color: red;'>Error creating directory: {str(e)}</div>"))
        return False

# Check if file exists
file_exists = os.path.isfile(file_path)

if not file_exists:
    # If file doesn't exist, create it with the first JSON object
    try:
        with open(file_path, 'w') as file:
            json.dump({"responses": [json_data]}, file, indent=2)
        display(HTML(f"<div style='color: green;'>Created new file {file_path} with initial JSON data</div>"))
    except Exception as e:
        display(HTML(f"<div style='color: red;'>Error creating file: {str(e)}</div>"))
        return False
else:
    # If file exists, read it, append the new data, and write it back
    try:
        with open(file_path, 'r') as file:
            existing_data = json.load(file)

        # Append new data to the responses list
        existing_data["responses"].append(json_data)

        # Write updated data back to file
```

```
            with open(file_path, 'w') as file:
                json.dump(existing_data, file, indent=2)

            display(HTML(f"<div style='color: green;'>Appended JSON data to {file_path}</div>"))
        except Exception as e:
            display(HTML(f"<div style='color: red;'>Error processing file: {str(e)}</div>"))
            return False

    return True

def extract_json_from_text(text):
    """
    Extracts JSON content from a text that might contain other formatting elements.

    Args:
        text (str): Text potentially containing JSON

    Returns:
        str or None: Extracted JSON string or None if no JSON found
    """
    # Remove code block markers and language identifiers
    # This handles formats like: ```json\n{...}\n``` or ```\n{...}\n```
    json_pattern = r'```(?:json)?\s*\n(.*?)\n```'

    # Use re.DOTALL to make '.' match newlines as well
    match = re.search(json_pattern, text, re.DOTALL)

    if match:
        return match.group(1).strip()

    # If no match with code blocks, try to find JSON directly
    # Look for text that starts with { and ends with }
    json_direct_pattern = r'(\{.*\})'
    match = re.search(json_direct_pattern, text, re.DOTALL)

    if match:
        return match.group(1).strip()

    return None
```

```
extract_and_append_json(gemma_decoded, "llm_responses.json")
```

References

[1]. Huber, S. E., Kiili, K., Nebel, S., Sailer, M., & Ninaus, M. (2024). Leveraging the Potential of Large Language Models in Education Through Playful and Game-Based Learning. Journal of Educational Technology Studies.

[2]. Alhafni, B., Vajjala, S., Bannò, S., Maurya, K. K., & Kochmar, E. (2024). LLMs in Education: Novel Perspectives, Challenges, and Opportunities. Education Research Journal.

[3]. Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D. (2024). Practical and Ethical Challenges of Large Language Models in Education: A Systematic Scoping Review. Educational Data Science Review.

[4]. Hasanbeig, H., Sharma, H., Betthauser, L., Vieira Frujeri, F., & Momennajad, I. (2024). ALLURE: Auditing and Improving LLM-Based Evaluation of Text Using Iterative In-Context Learning. Microsoft Research Reports.

[5]. Lyu, W., Wang, Y., Sun, Y., Chung, T. R., & Zhang, Y. (2024). Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. arXiv preprint arXiv:2404.13414.

[6]. Ismail, I. A., Mawardi, M., Qadriati, M., Humane, M., & Arif, K. (2024). Revolutionizing Science Education Evaluation Using a Vision-Language Model of Effective Assessment and Supervision. Education and AI Research Journal.

[7]. Kim, T. H., Kim, J. S., Yoon, H. I., Lee, J., Lee, J. J. B., Byun, H. K., Cho, Y., Kim, Y. B., Lee, I. J., Kim, K. H., & Chang, J. S. (2024). Medical Student Education Through Flipped Learning and Virtual Rotations in Radiation Oncology During the COVID-19 Pandemic: A Cross-Sectional Research. Journal of Medical Education.

[8]. Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2024). Improved Baselines with Visual Instruction Tuning. arXiv preprint arXiv:2403.00231.

[9]. Zhang, G., Zhang, Y., & Zhang, K. (2024). Can Vision-Language Models Be a Good Guesser? Exploring VLMs for Times and Location Reasoning. arXiv preprint arXiv:2307.06166.

[10]. Lamm, B., & Keuper, J. (2024). Can Visual Language Models Replace OCR-Based Visual Question Answering Pipelines in Production? A Case Study in Retail. arXiv preprint arXiv:2408.15626.

[11]. Cărțune, V., Mansoor, H., Baechler, G., Liu, F., Chen, J., Aralikatte, R., &

Sharma, A. (2024). Chart-Based Reasoning: Transferring Capabilities from LLMs to VLMs. Google Research Reports.

[12]. Oh, Y., Cho, J., Kim, D. J., Kweon, I. S., & Kim, J. (2024). Preserving Multi-Modal Capabilities of Pre-Trained VLMs for Improving Vision-Linguistic Compositionality. KAIST Research Papers.

[13]. Li, L., Wang, Y., Xu, R., Wang, P., Feng, X., Kong, L., & Liu, Q. (2024). Multimodal ArXiv: A Dataset for Improving Scientific Comprehension of Large Vision-Language Models. arXiv preprint arXiv:2403.00231.

[14]. Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., & Jia, J. (2024). Mini-Gemini: Mining the Potential of Multi-Modality Vision Language Models. arXiv preprint arXiv:2403.18814.

[15]. Zhou, G., Hong, Y., Wang, Z., Wu, Q., & Wang, X. E. (2024). NavGPT-2: Unleashing Navigational Reasoning Capability for Large Vision-Language Models. arXiv preprint arXiv:2407.12366.

[16]. Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2024). Visual Instruction Tuning. University of Wisconsin–Madison Research Papers.

[17]. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., et al. (2024). Qwen2.5 VL-3B-Instruct: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191.

[18]. Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., et al. (2024). PaliGemma: A Versatile 3B VLM for Transfer. Google Research.

[19]. Steiner, A., Pinto, A. S., Tschannen, M., Keysers, D., et al. (2024). PaliGemma 2: A Family of Versatile VLMs for Transfer. Google DeepMind.

[20]. Chi, J., Karn, U., Zhan, H., Smith, E., et al. (2024). LlamaGuard3Vision: Safeguarding Human-AI Image Understanding Conversations. arXiv preprint arXiv:2411.10414.

[21]. Bai, Jinze, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou and Jingren Zhou. "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond." (2023).

[22]. Wang, W., et al. (2024). CogVLM: Visual Expert for Pretrained Language Models. AI and Education Research.

[23]. Hong, W., et al. (2024). CogVLM2: Visual Language Models for Image and Video Understanding. AI Research Papers.

[24]. Tan, J. W., et al. (2024). Clinical-Grade Multi-Organ Pathology Report Generation. Medical Image Analysis.

[25]. Qu, L., et al. (2024). Pathology-Knowledge Enhanced Few-Shot WSI Classification. Pathology Research Journal.

[26]. Nguyen, A. T., et al. (2024). Text-Based Quantitative Histopathology Image Analysis. AI in Histopathology.

[27]. Liu, Q., et al. (2024). mTREE: End-to-End WSI Analysis. arXiv preprint.

[28]. Liu, Q., et al. (2024). mTREE: End-to-End WSI Analysis. arXiv preprint.

[29]. Pereira Júnior, C., et al. (2024). Children's Handwritten Math Recognition. Educational Technology Studies.

[30]. Chauhan, M., et al. (2024). Handwriting Verification Using Vision-Language Models. AI and Handwriting Recognition.

[31]. Boteanu, A., et al. (2024). Read-Write-Learn: Framework for Handwriting Recognition. Handwriting Technology.

[32]. Aguilar, S. T. (2024). HTR for Historical Documents Using GANs. Historical Document Analysis.

[33]. Baral, S., et al. (2024). DrawEduMath: Evaluating VLMs on Math Images. Educational AI.