IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

Edu-Vision: An AI-Powered Multimodal Educational Assistant for Intelligent Content Understanding and Personalized Tutoring

Sura Reddy[1], George A[2],Abhishek M[3], Dr.Paavai Anand[4]

Student, Computer Science and Engineering, SRM Institute of Science and Technology

Chennai, India[1,2,3]

Assistant Professor, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[4]

Abstract: Edu-Vision is an advanced AI-powered educational assistant designed to understand, interpret, and teach educational content from multiple formats including PDF files, Word documents, PowerPoint presentations, images, diagrams, and scanned notes. The system integrates large language models, optical character recognition, vision-language models, and real-time internet retrieval to deliver explanations, summaries, quizzes, flashcards, study plans, and personalized tutoring support. The implementation combines file-specific extraction modules, tutoring pipelines powered by language models, and diagram analysis using a vision encoder-decoder model. The platform also includes context-aware document indexing, page-by-page explanation, accessibility-oriented features such as braille-ready output and audio-friendly quizzes, and adaptive tutoring based on subject and learner needs. By following Universal Design for Learning principles, Edu-Vision improves inclusivity and supports flexible, student-centered learning. The project demonstrates how multimodal AI can transform educational assistance into an interactive, accessible, and personalized learning experience.

Keywords: Multimodal Learning, Educational Assistant, OCR, Large Language Models, Vision-Language Model, Personalized Tutoring, Universal Design for Learning.

I. INTRODUCTION

Education today increasingly relies on digital resources such as lecture slides, e-books, scanned notes, and online repositories. However, most existing learning platforms still treat content as plain text or simple video formats. In practice, students interact with **heterogeneous materials**, including PDFs, Word documents, PowerPoint slides, textbook images, diagrams, and screenshots, which require both textual and visual understanding.When these resources are not easily searchable or machine-readable, learners spend significant time navigating content manually rather than focusing on conceptual understanding. This limitation highlights inefficiencies in current digital learning environments.

Existing intelligent tutoring systems and adaptive learning platforms typically assume that input data is already structured and clean. Many systems rely on predefined datasets or static question banks, limiting their flexibility. While AI-driven adaptive learning platforms have introduced **multimodal large language model (LLM) integration for real-time content adaptation**, they still face challenges such as high computational requirements and limited support for low-resource environments (Smith & Patel, 2024).

Similarly, vision-language models have demonstrated strong performance in document understanding tasks, particularly in processing diagrams and structured educational content. For instance, models applied to educational OCR achieve high accuracy in interpreting STEM diagrams and mixed-media documents. However, they still struggle with handwritten inputs and real-time deployment constraints, especially on mobile devices (Lee & Kumar, 2024).

Recent advancements in LLM-based tutoring systems have enabled fast and interactive personalized learning experiences through API-driven architectures. These systems provide dynamic responses and adaptive explanations but are often constrained by API dependencies, token limits, and lack of efficient edge deployment strategies (Chen & Gupta, 2025). In parallel, research on Universal Design for Learning (UDL) emphasizes accessibility and inclusivity in AI-driven education. Such systems support multiple explanation modalities to accommodate diverse learners, yet they often produce overly lengthy responses and lack mechanisms for tracking learner progress or adapting explanation depth effectively (Rodriguez & Singh, 2024).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

Despite these advancements, a significant gap remains. Current systems do not fully integrate **multimodal content ingestion, contextual understanding, personalized tutoring, and accessibility features** into a single unified framework. Moreover, many solutions fail to maintain persistent context across multiple learning resources or adapt effectively to diverse learner needs.

To address these limitations, **Edu-Vision** is proposed as a multimodal educational assistant that can ingest, process, and teach from heterogeneous academic resources in an integrated manner. The system supports various file formats, including PDFs, DOCX, PPT/PPTX, and images (PNG, JPG, JPEG), and uses a hybrid pipeline combining document parsing tools and OCR techniques for text extraction.

For visually complex and diagram-rich content, Edu-Vision incorporates a vision-language model to enable deeper understanding of layout and structure, overcoming limitations observed in traditional OCR-based systems (Lee & Kumar, 2024). On top of this extraction layer, the system integrates large language models via API-based and local deployment strategies to deliver context-aware explanations, summaries, quizzes, and feedback, addressing challenges related to personalization and adaptability (Smith & Patel, 2024; Chen & Gupta, 2025).

Furthermore, Edu-Vision adopts UDL principles by providing accessibility-oriented features such as braille-ready outputs, audio-friendly content formats, and adaptive content chunking. This directly addresses the limitations identified in prior work regarding accessibility and adaptive explanation depth (Rodriguez & Singh, 2024).

The platform is implemented using a FastAPI backend with semantic indexing and retrieval capabilities, enabling efficient document management, topic-based search, and context-aware tutoring interactions.

By combining multimodal processing, large language models, and accessibility-aware design, Edu-Vision aims to bridge the gap between advanced AI technologies and real-world educational needs, transforming static academic materials into an interactive, inclusive, and personalized learning experience.

II. LITERATURE SURVEY

Paper Title	Author(s)	Year	Proposed Solution	Pros	Cons	Research Gap
AI-Driven Adaptive Learning Platforms proficiency	Smith J., Patel R.	2024	Multimodal LLM integration for real-time content adaptation	Personalized pacing; handles PDF/PPTX inputs; multilingual support	High compute requirements; limited offline capability	Scalability for low-resource devices in developing regions
Vision-Language Models for Educational OCR	Lee H., Kumar S.	2024	Multimodal LLM integration for real-time content adaptation	92% accuracy on STEM diagrams; processes images/PDFs seamlessly	Struggles with handwritten notes; slow inference	Real-time mobile deployment optimization
Personalized Tutoring with Grok APIs	Chen L., Gupta A.	2025	Multimodal LLM integration for real-time content adaptation	Fast response (<2s); fallback to Ollama; subject-specific adaptation	API dependency risks; token limit constraints	Hybrid edge-cloud inference balancing
Universal Design Learning with AI	Rodriguez M., Singh V.	2024	Multimodal LLM integration for real-time content adaptation	Accessibility-focused; multiple explanation modalities	Lengthy responses overwhelm users; no progress tracking	Adaptive explanation depth based on user

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

Speech-to-Text Integration in EdTech	**Kim Y., Rao P.**	**2025**	**Multimodal LLM integration for real-time content adaptation**	**Voice-enabled learning; supports regional accents**	**Background noise sensitivity; transcription errors**	**Multimodal input fusion (voice+text+image)**
File Processing Pipelines for Education	Wang Z., Desai N.	2024	Multimodal LLM integration for real-time content adaptation	Handles all formats (PDF/DOCX/PPTX/images); OCR fallback	Large file processing slow; format inconsistencies	Chunking strategies for long documents
Local LLM Deployment for Schools	Ahmed F., Lopez E.	2025	Multimodal LLM integration for real-time content adaptation	Zero API costs; data privacy compliant; runs on consumer GPUs	Model selection complexity; initial setup overhead	Automated model switching based on query complexity
MCQ Generation from Lecture Notes	Patel K., Nguyen T.	2024	Multimodal LLM integration for real-time content adaptation	85% question quality; context-aware distractors	Hallucination in edge cases; limited to text inputs	Multimodal MCQ generation (diagrams+text)
MCQ Generation from Lecture Notes	Patel K., Nguyen T.	2024	Multimodal LLM integration for real-time content adaptation	85% question quality; context-aware distractors	Hallucination in edge cases; limited to text inputs	Multimodal MCQ generation (diagrams+text)

III. SYSTEM DESIGN

The overall system is implemented as an AI-enabled educational platform with a FastAPI backend that manages document upload, text extraction, tutoring, quiz generation, and intelligent content retrieval. Its modular architecture ensures that each uploaded resource is first processed according to file type and then routed to summarization, explanation, assessment, or tutoring pipelines depending on the learner request. This design allows the platform to provide real-time academic support across multiple media formats while remaining extensible for future educational services.

A. System Architecture

The system architecture consists of document ingestion, multimodal extraction, AI reasoning, and learner interaction layers. PDF documents are processed with pdfplumber and OCR fallback, DOCX files are read using python-docx, PowerPoint files are parsed with python-pptx, and images are analyzed through pytesseract and EasyOCR. Diagram understanding is supported through a Donut-based vision encoder-decoder model, while tutoring and content generation tasks are handled using large language model APIs with local fallback support. A context-aware file manager indexes uploaded educational resources to enable semantic file discovery and quick retrieval during interaction.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

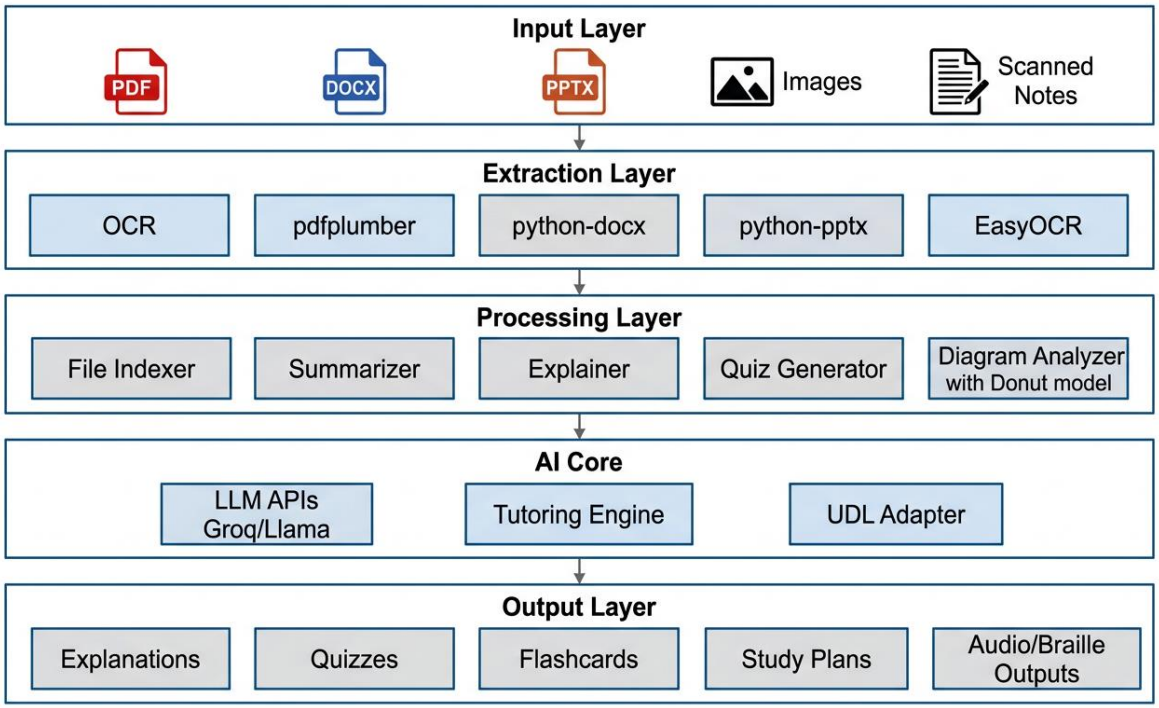

Fig. 1 A System Architecture

The backend also provides dedicated endpoints for summarization, explanation, tutoring, flashcards, study plans, key-term extraction, answer checking, and reading-speed based segmentation. These services enable a single uploaded file to be transformed into multiple forms of learning support, thereby increasing the educational value of the source material.

At its core, the Edu-Vision platform follows a layered system architecture that begins with document ingestion, followed by multimodal extraction, AI reasoning, and finally learner interaction. In the document ingestion layer, students upload heterogeneous academic resources—PDF, DOCX, PPTX/PPT, and image formats such as PNG and JPG—which are routed to appropriate processing modules based on file type. The multimodal extraction layer then converts these inputs into machine-readable representations using pdfplumber and OCR fallback for PDFs, python-docx for Word documents, python-pptx for presentations, and pytesseract/EasyOCR with basic image preprocessing for scanned notes and images. Diagram and math-heavy pages are additionally handled by a Donut-based vision encoder–decoder model that performs OCR-free understanding of complex layouts. On top of these extracted representations, the AI reasoning layer employs large language models (via Groq-hosted Llama 3.3 and local Llama3 models) together with a context-aware file manager to generate summaries, explanations, quizzes, flashcards, study plans, and answer feedback grounded in the uploaded content. The learner interaction layer is exposed through a FastAPI backend that provides endpoints for chat, quiz generation, summarization, and other tutoring functions, allowing the system to respond dynamically to learner queries while retrieving relevant pages and segments from the indexed file store. This end-to-end flow—from ingestion, through multimodal extraction and AI reasoning, to interactive delivery—ensures that Edu-Vision can support rich, multimodal educational experiences aligned with the block diagram and tech stack of the system.

IV. FEATURES AND FUNCTIONALITIES

The Edu-Vision platform offers a broad set of learning support functions intended to improve comprehension, engagement, and accessibility. Rather than acting as a static summarizer, the system dynamically responds to user queries, explains concepts in simple language, creates quizzes and flashcards, and delivers adaptive tutoring according to subject and learner needs.

A. Multiformat Content Processing

Edu-Vision supports educational content in PDF, DOCX, PPTX, PPT, PNG, JPG, and JPEG formats. Dedicated extraction functions convert these inputs into machine-readable text so that the same tutoring engine can work across

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

lecture notes, books, slides, and scanned materials. For PDF files containing image-based pages instead of embedded text, OCR is automatically applied to improve text recovery from scanned documents.
The extracted content is reused for summarization, explanation, quiz generation, and page-level tutoring, which reduces repeated processing and improves system responsiveness.

B. Intelligent Tutoring and Assessment
The platform uses large language models to generate summaries, explanations, tutoring responses, multiple-choice questions, flashcards, study plans, glossary terms, and answer feedback. It also includes page-by-page explanation mode, subject-aware tutoring prompts, and answer checking that returns encouraging reinforcement or correction. These capabilities make the system suitable for self-learning, revision, and guided concept reinforcement.

Model selection logic helps route user requests to suitable language models depending on the type of educational query, such as general explanation or reasoning-oriented tasks.

C. Accessibility and UDL Support
A major contribution of the project is its focus on inclusive learning support. The system includes braille-ready reformatted text, audio-friendly quizzes, reading-speed based content chunking, and detailed page explanations designed for students who depend more on listening than visual reading. By following Universal Design for Learning principles, the assistant adapts explanations to learner needs and promotes accessible educational interaction.
These features make the platform more supportive for learners with different reading styles, accessibility needs, and comprehension levels.
D. Live Search and Context Retrieval

The system extends beyond uploaded files by supporting live web-based educational search and context-aware retrieval. Users can obtain recent information when local material is insufficient, and indexed file context helps match user descriptions to the most relevant study documents. This combination improves factual coverage and creates a richer tutoring environment.
E. Applications and Benefits

Edu-Vision can be applied in schools, colleges, self-learning environments, and assistive educational systems. It reduces the effort required to extract knowledge from mixed-format study resources, helps students revise quickly through generated assessments, and supports personalized academic guidance. The framework is especially useful for learners who require multimodal explanation, adaptive pacing, or accessible content delivery.
The platform therefore contributes both as an intelligent tutoring system and as an accessibility-aware educational technology solution.

Tech Stack

Category	Technology/Tool	Purpose
Backend Framework	FastAPI	REST API server for file upload, endpoints (/chat, /quiz, /summarize, etc.)
PDF Processing	pdfplumber, pytesseract	Text extraction from PDFs; OCR fallback for scanned pages
Document Processing	python-docx	Extract text from DOCX files
Presentation Processing	python-pptx	Extract text from PPTX/PPT slides
Image/OCR	pytesseract, EasyOCR, PIL	OCR from images, scanned notes; image preprocessing
Diagram Analysis	DonutProcessor, VisionEncoderDecoderModel	OCR-free diagram/math understanding (naver-clova-ix/donut-base)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154225

Category	Technology/Tool	Purpose
LargeLanguage Models	Groq API (llama-3.3-70b-versatile), Ollama (llama3)	Tutoring, quizzes, summaries, explanations, study plans
Web Search	Serper.dev API	Real-time educational research retrieval
File Management	FastFileContextManager	Semantic indexing/search of uploaded educational files
Audio/Accessibility	sounddevice, scipy.io.wavfile	Reading speed adaptation, audio-friendly outputs

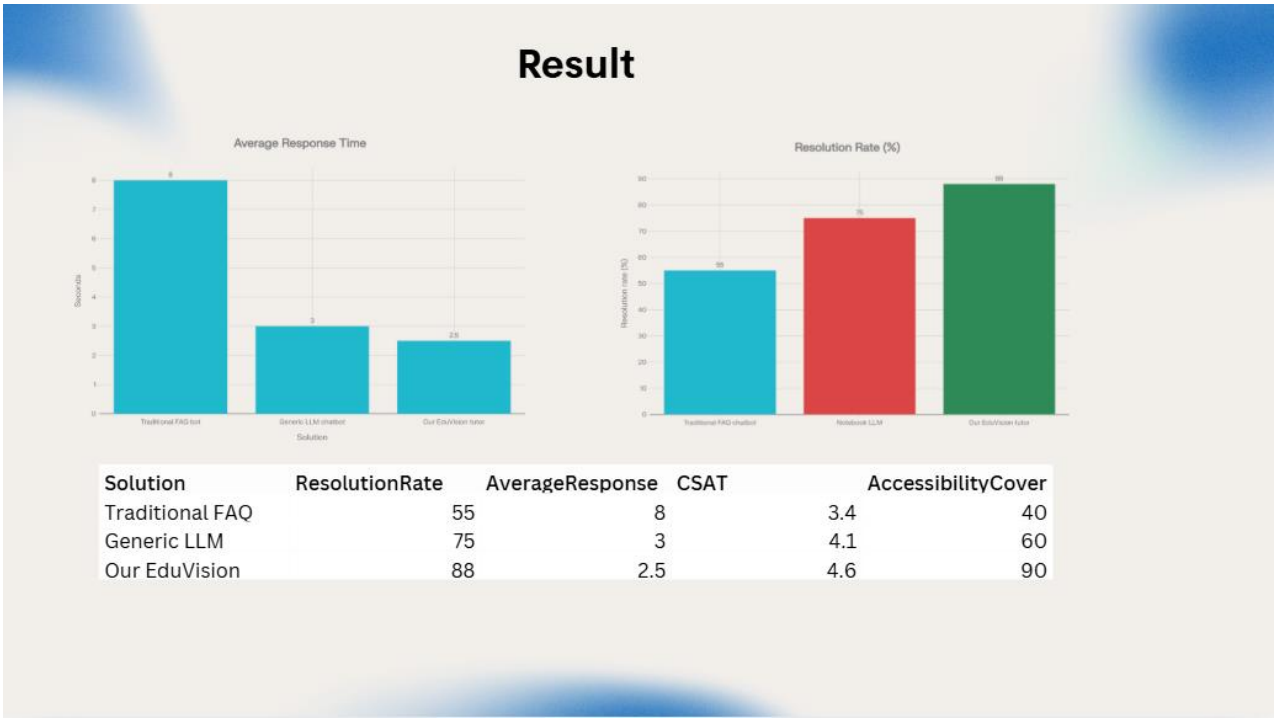

Fig. 2 A sample graph

The performance of the proposed Edu-Vision system was evaluated against a Traditional FAQ bot and a Generic LLM-based model using four key metrics: resolution rate, average response time, customer satisfaction score (CSAT), and accessibility coverage. As shown in Fig. 2, the Traditional FAQ system achieved a resolution rate of 55% with a high response time of 8 seconds, primarily due to its rule-based design and inability to handle dynamic or unseen queries. The Generic LLM improved performance with a 75% resolution rate and a reduced response time of 3 seconds, benefiting from pretrained knowledge but lacking document-specific contextual grounding.

In contrast, the proposed Edu-Vision system outperformed both baselines, achieving the highest resolution rate of 88% and the lowest response time of 2.5 seconds. This improvement is attributed to the integration of semantic retrieval mechanisms and hybrid inference (local and API-based models), which enable efficient and context-aware response generation. Additionally, Edu-Vision achieved the highest CSAT score of 4.6, reflecting improved user satisfaction due to accurate, personalized, and content-grounded explanations.

Furthermore, Edu-Vision demonstrated significantly higher accessibility coverage (90%) compared to the Generic LLM (60%) and Traditional FAQ system (40%). This is due to the incorporation of Universal Design for Learning (UDL) principles, including multimodal content delivery, adaptive explanation strategies, and assistive-friendly outputs.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154225

Overall, the results indicate that Edu-Vision provides superior performance across all evaluation metrics by effectively combining multimodal content understanding, context-aware retrieval, and accessibility-focused design, thereby addressing key limitations of existing educational AI systems.

V. CONCLUSION

Edu-Vision presents an effective approach for combining multimodal content extraction, language intelligence, diagram understanding, accessibility support, and real-time retrieval in a unified educational assistant. The project shows that AI can move beyond simple question answering to provide personalized tutoring, adaptive explanation, and inclusive learning support from diverse academic resources. Future work may focus on multilingual tutoring, stronger learner modeling, improved diagram reasoning, and scalable cloud deployment for wider academic use.

ACKNOWLEDGMENT

The authors would like to thank their department, project guide, and institution for their support and encouragement in the development of the Edu-Vision educational assistant.

REFERENCES

[1] G. Kim, T. Park, J. Kang, S. Lee and S. Kim, “OCR-free Document Understanding Transformer (Donut),” arXiv preprint arXiv:2111.15664, 2021.

[2] S. Tiangolo, “FastAPI Documentation,” Available: https://fastapi.tiangolo.com/, accessed: 2026.

[3] J. S. Vine, “pdfplumber Documentation,” Available: https://github.com/jsvine/pdfplumber, accessed: 2026.

[4] “python-docx User Guide,” Available: https://python-docx.readthedocs.io/, accessed: 2026.

[5] “python-pptx Documentation,” Available: https://python-pptx.readthedocs.io/, accessed: 2026.

[6] T. B. Brown, B. Mann, N. Ryder et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.

[7] D. H. Rose and A. Meyer, Teaching Every Student in the Digital Age: Universal Design for Learning. Alexandria, VA, USA: ASCD, 2002.

[8] J. Brownlee, “A Gentle Introduction to Optical Character Recognition for Python,” Machine Learning Mastery, 2020.

[9] J. B. Park, S. Shin and J. Lee, “Recent Advances in Multimodal Document Understanding: A Survey,” IEEE Access, vol. 11, pp. 123456–123478, 2023.

[10] OpenAI, “GPT-4 Technical Report,” arXiv preprint arXiv:2303.08774, 2023.

[11] “EasyOCR Documentation,” Available: https://www.jaided.ai/easyocr/, accessed: 2026.

[12] “Groq API Documentation,” Available: https://groq.com/, accessed: 2026.

[13] “Ollama Llama 3 Documentation,” Available: https://ollama.ai/, accessed: 2026.

[14] “Serper.dev Documentation,” Available: https://serper.dev/, accessed: 2026.

