iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80286

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Neural Docs: An Intelligent AI-Driven Document Management System Using OCR and Retrieval-Augmented Generation

Sahil Sanjay Haryan[1], Maruthu Devendrar[2], Gupta Rahul Shitlaprasad[3], Tanvirali Hadginal[4], Prof. Nargis Shaikh[5]
Department of Electronics & Computer Science, RCOE, Mumbai, Maharashtra, India

Abstract: ***Managing large volumes of documents remains a challenging task for individuals and organizations, especially when data exists in unstructured formats such as scanned files, PDFs, and images. Traditional document management approaches rely heavily on manual effort for sorting, searching, and extracting information, which leads to inefficiencies and increased processing time. To address these challenges, this paper introduces Neural Docs, an AI-enabled document management system that focuses on automating document understanding and interaction.***

The system is designed to transform static documents into interactive and searchable knowledge sources. It utilizes Optical Character Recognition (OCR) to convert visual document content into machine-readable text. This extracted information is further processed using Natural Language Processing (NLP) techniques to identify key entities, generate metadata, and organize documents intelligently. In addition, a Retrieval-Augmented Generation (RAG) mechanism is incorporated to enable users to query documents through a conversational interface, providing responses based on relevant contextual information rather than simple keyword matching.

The overall architecture is divided into multiple layers, including a user interface for interaction, a backend system for processing and coordination, and a dedicated AI module for advanced analysis. A vector-based storage mechanism is used to maintain semantic representations of documents, allowing efficient similarity-based retrieval. The system is implemented using modern technologies such as Node.js, FastAPI, and containerized deployment for flexibility and scalability.

The developed solution demonstrates improved accessibility and reduced manual workload in document handling tasks. It allows users to quickly retrieve meaningful insights from stored documents and interact with them in a more intuitive way. The approach presented in this work highlights the potential of combining multiple AI techniques to create smarter and more efficient document management systems suitable for real-world applications.

Keywords: Artificial Intelligence, Document Management System, Optical Character Recognition (OCR), Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG), Semantic Search, Intelligent Automation, Information Extraction

I. INTRODUCTION

In the digital era, the volume of documents generated and processed by individuals, businesses, and government institutions has increased significantly. These documents often exist in diverse formats such as scanned images, PDFs, and printed records, making their management a complex and time-consuming task. Despite the availability of digital storage solutions, many document-centric workflows still rely on manual efforts for organization, verification, and retrieval of information. This not only reduces efficiency but also increases the chances of human error and data inconsistency.

Traditional document management systems primarily focus on storage and basic search capabilities, lacking the ability to understand and interpret the content within documents. As a result, users are often required to manually search through multiple files to locate relevant information. This limitation becomes more prominent in scenarios involving compliance processes, identity verification, and administrative tasks, where quick and accurate access to document data is essential.

To overcome these challenges, there is a growing need for intelligent systems that can automate document processing and provide meaningful insights. In this context, Neural Docs is proposed as an AI-driven document management solution that transforms static documents into interactive and searchable resources. The system leverages Optical Character Recognition (OCR) to extract textual data from unstructured documents and applies Natural Language Processing (NLP) techniques to analyse and organize the extracted content. Furthermore, the integration of Retrieval-Augmented Generation (RAG) enables users to interact with their documents through natural language queries, offering context-aware responses instead of simple keyword-based results.

By combining these technologies within a modular and scalable architecture, the proposed system aims to simplify document handling, reduce manual workload, and enhance overall accessibility of information.

II. RELATED WORK

In recent years, significant research has been carried out in the areas of document processing, natural language understanding, and intelligent information retrieval. These advancements have contributed to the development of systems capable of automating various document-centric tasks. However, most existing solutions focus on specific aspects rather than providing a unified and intelligent framework.

Several studies have explored the use of Optical Character Recognition (OCR) for extracting textual content from scanned documents and images. OCR-based systems have been widely adopted in applications such as digitization of records and automated data entry. While these systems effectively convert visual data into text, they often lack the capability to interpret and organize the extracted information in a meaningful way.

Research in Natural Language Processing (NLP) has further enhanced document understanding by enabling tasks such as entity recognition, classification, and summarization. NLP-based systems can process textual data and extract relevant insights, improving the efficiency of information handling. However, these systems are generally limited to structured text inputs and may not perform effectively when integrated with raw, unstructured document data.

In addition, conversational AI and chatbot-based systems have been developed to improve user interaction and accessibility. These systems allow users to perform queries using natural language, reducing the complexity of traditional search mechanisms. Despite their advantages, many chatbot implementations rely on predefined rules or simple keyword matching, which limits their ability to provide context-aware responses.

More recently, Retrieval-Augmented Generation (RAG) has emerged as a promising approach that combines information retrieval with generative models to deliver more accurate and context-driven responses. RAG-based systems have shown improved performance in knowledge-intensive tasks by retrieving relevant data before generating responses. However, their application in document management systems is still limited.

From the analysis of existing work, it is evident that most solutions address individual components such as OCR, NLP, or chatbot interaction independently. There is a lack of integrated systems that combine these technologies into a single platform capable of end-to-end document automation and intelligent interaction. The proposed Neural Docs system aims to bridge this gap by integrating OCR, NLP, and RAG into a unified architecture for efficient and intelligent document management.

III. METHODOLOGY

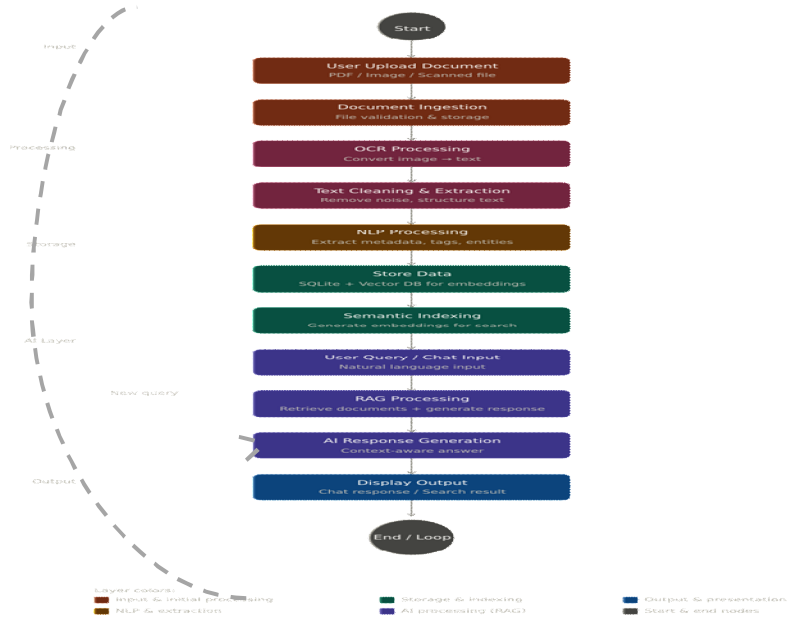


Fig.3.1 Flowchart of AI Powered Document Management System

The methodology of the *Neural Docs* system describes the complete workflow followed for processing, managing, and interacting with documents using AI-based techniques. The system is designed as a multi-stage pipeline where each stage performs a specific function, ensuring efficient and intelligent document handling.
The overall working of the system is divided into the following sequential steps:

1) User Upload
The process begins when the user uploads a document through the system interface. The document can be in various formats such as PDF, scanned image, or digital file.

2) Document Ingestion
The uploaded document is received by the backend server, where it is stored and prepared for further processing. This step ensures proper file handling and format validation.

3) OCR Processing
Optical Character Recognition (OCR) is applied to the document to extract textual content from images or scanned files. This step converts visual data into machine-readable text.

4) Text Extraction
The extracted text is cleaned and structured to remove noise and irrelevant characters. This prepares the data for further analysis.

5) NLP Processing (Metadata Extraction)
Natural Language Processing techniques are used to analyse the text and extract important information such as document title, entities, tags, and key attributes.

6) Storage (Database + Vector Database)
The processed data is stored in a relational database for structured information, while semantic embeddings of the document are stored in a vector database for advanced retrieval.

7) Semantic Indexing
The system generates vector representations of the document content, enabling similarity-based search and efficient retrieval of relevant information.

8) User Query (Chat/Search)
The user interacts with the system through a search interface or chatbot by entering natural language queries.

9) RAG Processing
The Retrieval-Augmented Generation mechanism retrieves relevant document data and combines it with a language model to generate context-aware responses.

10) Response Generation
The system processes the query and generates a meaningful and accurate response based on retrieved information.

11) User Output
Finally, the response is displayed to the user through the interface, providing relevant information in an easy-to-understand format.

IV. SYSTEM ARCHITECTURE

A. Architecture Overview
The architecture of *Neural Docs* is designed to provide an efficient and scalable framework for intelligent document management by integrating traditional document processing techniques with advanced artificial intelligence capabilities.

The system follows a modular and service-oriented architecture, where different components operate independently while maintaining seamless communication through APIs.

At a high level, the system consists of a user-facing interface, a backend application layer, an AI processing module, and multiple data storage mechanisms. The frontend provides an interactive platform for users to upload documents, search information, and interact with the system through a conversational interface. The backend, implemented using a server-side framework, acts as the central controller that manages document ingestion, processing workflows, and communication between various services.

A key aspect of the architecture is the integration of an AI processing layer that handles tasks such as text extraction, metadata generation, and intelligent querying. This layer incorporates Optical Character Recognition (OCR) to convert document content into machine-readable text, followed by Natural Language Processing (NLP) techniques to analyze and structure the extracted information. Additionally, a Retrieval-Augmented Generation (RAG) mechanism is employed to enable context-aware interaction with stored documents, allowing users to retrieve relevant information through natural language queries.

The system also includes a dual storage strategy, where structured data such as metadata is maintained in a relational database, while semantic representations of documents are stored in a vector-based database to support efficient similarity search. External services and APIs are integrated to enhance functionality and enable extensibility.

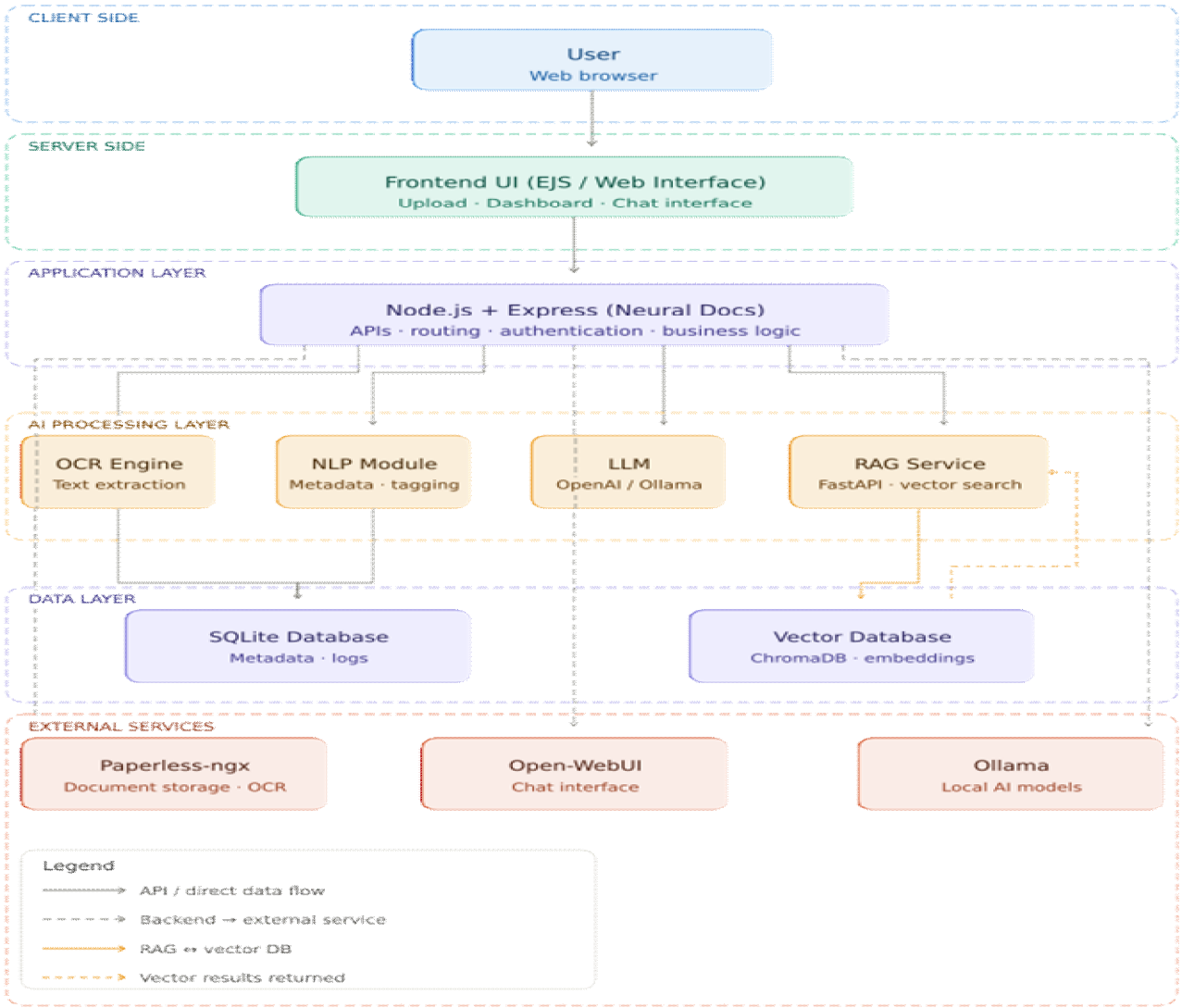


Fig. 4.1 Architecture Overview Diagram

B. Architectural Layers

The Neural Docs system is structured using a layered architecture approach to ensure modularity, scalability, and efficient separation of concerns.

Each layer is responsible for a specific set of functionalities, allowing independent development, maintenance, and scalability of system components. The architecture is divided into five primary layers: Presentation Layer, Application Layer, AI Processing Layer, Data Layer, and External Services Layer.

1) Presentation Layer

The presentation layer serves as the interface between the user and the system. It is responsible for handling user interactions and displaying system outputs in an intuitive manner. This layer includes the web-based user interface, which allows users to upload documents, view processed data, and interact with the system through a chatbot interface. It ensures a smooth and responsive user experience by managing input requests and rendering outputs such as search results and AI-generated responses.

2) Application Layer (Backend)

The application layer acts as the core control unit of the system. It is responsible for processing user requests, managing workflows, and coordinating communication between different components. Implemented using a server-side framework, this layer handles tasks such as document ingestion, API routing, authentication, and business logic execution. It serves as a bridge between the presentation layer and the underlying AI and data layers.

3) AI Processing Layer

The AI processing layer is the intelligence core of the system. It performs advanced operations such as text extraction, data analysis, and response generation. This layer integrates multiple AI techniques, including OCR for converting documents into text, NLP for extracting meaningful information, and Retrieval-Augmented Generation (RAG) for enabling context-aware query responses. The integration of these components allows the system to transform raw documents into structured and interactive data.

4) Data Layer

The data layer is responsible for storing and managing all system data. It includes both structured and unstructured storage mechanisms. Structured data such as metadata, logs, and user information is stored in a relational database, while semantic representations of document content are stored in a vector database. This dual-storage approach enables efficient querying, fast retrieval, and support for advanced search capabilities.

5) External Services Layer

The external services layer includes third-party tools and services that enhance the functionality of the system. These services provide capabilities such as document storage, additional AI processing, and model inference. The system communicates with these services through APIs, enabling seamless integration and extensibility without increasing system complexity.

C. Component Description

The Neural Docs system is composed of several interconnected components, each responsible for handling specific functionalities within the overall architecture. These components work together to enable efficient document processing, intelligent analysis, and seamless user interaction.

1) Document Service

The Document Service manages the complete lifecycle of documents within the system. It handles document ingestion, storage, retrieval, and updates. This component communicates with external document management platforms to fetch document data and ensures that uploaded files are properly processed and organized for further analysis.

2) AI Service

The AI Service is responsible for performing intelligent operations on document data. It integrates with language models and processing engines to analyze extracted text, generate metadata, and produce meaningful outputs. This component supports multiple AI providers, enabling flexibility in choosing between cloud-based and local models for processing tasks.

3) OCR Module

The OCR module plays a crucial role in converting non-text documents into machine-readable formats. It processes scanned images and PDFs to extract textual content, which serves as the foundation for further analysis. This component ensures that even unstructured documents can be processed effectively.

4) RAG Service (Retrieval Module)

The RAG (Retrieval-Augmented Generation) service enables intelligent querying and document interaction. It retrieves relevant information from stored documents using semantic search and combines it with generative models to produce context-aware responses. This component enhances the system's ability to provide accurate and meaningful answers to user queries.

5) Search and Indexing Service

This component is responsible for creating and managing searchable indexes of document data. It generates semantic embeddings and maintains efficient retrieval mechanisms, allowing users to perform fast and accurate searches across large document collections.

6) User and Authentication Module

The User Module manages user accounts, authentication, and access control. It ensures secure interaction with the system by validating user credentials and maintaining session information. This component also supports role-based access to enhance system security.

7) Database Module

The Database Module handles data storage and management. It stores structured data such as metadata, processing logs, and user information, while also supporting storage for vector embeddings used in semantic search. This ensures efficient data organization and retrieval.

D. Data Flow Architecture

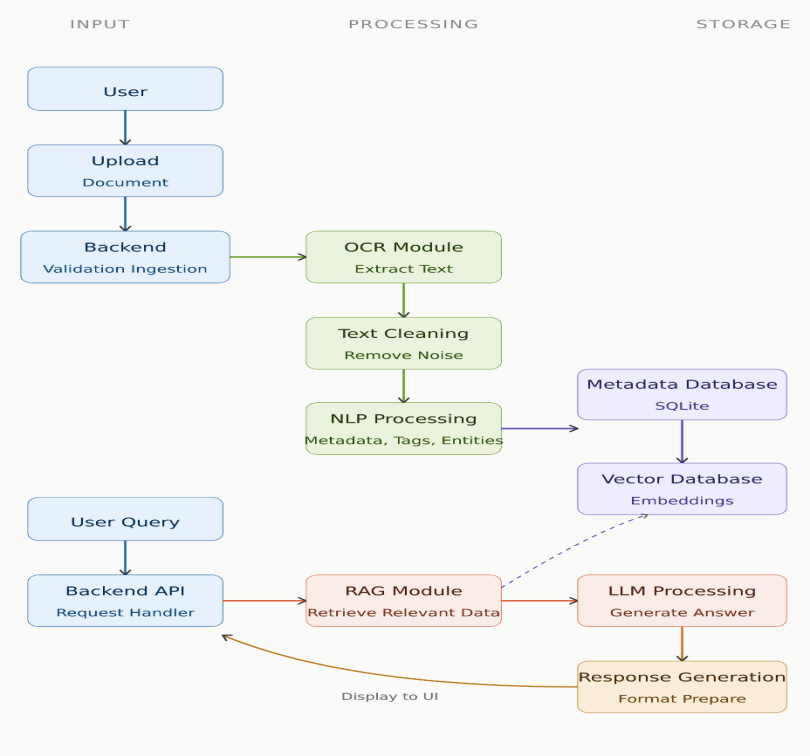


Fig. 4.2: Detailed Data Flow of the Proposed System

The data flow architecture of *Neural Docs* defines how information moves through the system from document input to final user interaction. The system follows a sequential yet modular pipeline where each stage processes and enhances the data before passing it to the next stage.

The process begins when a user uploads a document through the frontend interface. The document is received by the backend, where it undergoes validation and ingestion. If the document contains non-textual content such as scanned images or PDFs, the OCR module extracts textual information and converts it into a machine-readable format.

Once the text is extracted, it is processed using Natural Language Processing techniques to identify key entities, generate metadata, and classify the document. The processed data is then stored in the database, while semantic embeddings are generated and stored in a vector database for efficient retrieval.

When a user submits a query, the system processes the request through the backend, which interacts with the retrieval module. The RAG mechanism retrieves relevant document data using semantic similarity and forwards it to the language model for response generation. The generated output is then sent back to the user interface in a structured and understandable format.

This flow ensures that raw documents are transformed into meaningful information, enabling efficient search, intelligent querying, and improved user experience.

E. Deployment Architecture

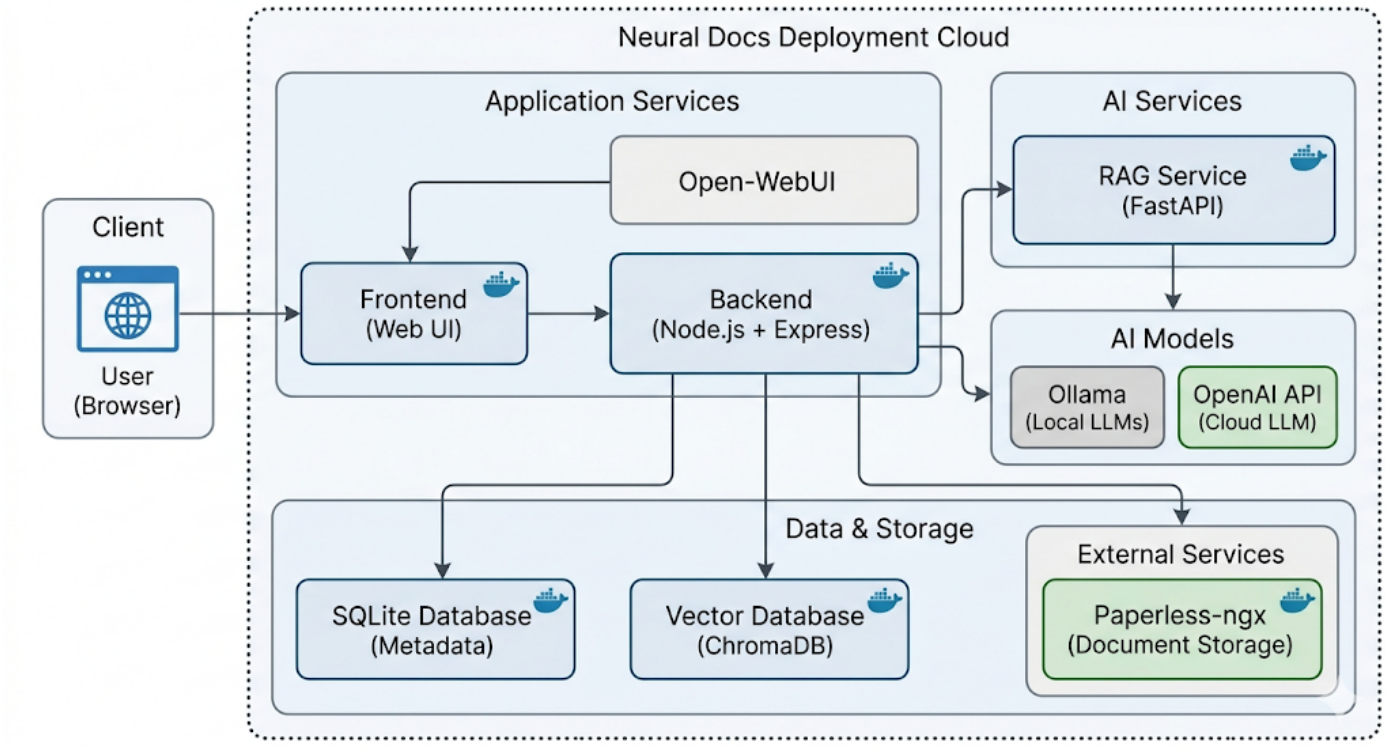


Fig. 4.3: Deployment Architecture Diagram

The deployment architecture of Neural Docs is designed to ensure scalability, modularity, and efficient resource utilization. The system is deployed using a containerized environment, where different services run independently and communicate through defined interfaces.

The core application, including the backend server, is deployed as a containerized service that handles API requests, document processing coordination, and system logic. The frontend interface is served through the same application or a separate service, providing user access via a web browser.

The AI processing components, including the RAG service and language models, are deployed as independent services. This separation allows scalable AI processing and flexibility in choosing between local or cloud-based models. The vector database used for semantic search operates as a dedicated service, ensuring efficient storage and retrieval of embeddings.

In addition, external services such as document management systems and model providers are integrated through APIs. All services are orchestrated using container management tools, enabling easy deployment, monitoring, and scaling of the system.

This architecture ensures that each component can be updated or scaled independently, making the system robust and adaptable to increasing workloads and future enhancements.

V. IMPLEMENTATION

A. Implementation Overview

The implementation of the Neural Docs system is carried out using a modular and service-oriented approach, ensuring flexibility, scalability, and efficient integration of multiple components. The system is designed to handle document processing tasks in a structured pipeline, where each module performs a specific function and communicates with other components through well-defined interfaces.

At a high level, the system consists of a frontend interface, a backend server, an AI processing module, and data storage layers. The frontend provides users with an interactive platform to upload documents, view processed results, and interact with the system through a conversational interface. The backend serves as the central controller, managing request handling, document processing workflows, and communication with both AI services and storage systems.

The implementation emphasizes the separation of concerns, where each layer operates independently. The backend is developed using a server-side framework that supports API-based communication, allowing seamless interaction between the frontend and underlying services. Document ingestion, validation, and processing are handled at this layer, ensuring that incoming data is structured before further analysis.

A significant aspect of the implementation is the integration of AI-based processing modules. These modules are responsible for extracting meaningful information from documents and enabling intelligent interaction. Optical Character Recognition is used to convert non-textual data into readable content, which is then processed using natural language techniques for metadata extraction and classification. Additionally, a retrieval-based mechanism is implemented to support context-aware query handling.

The system also incorporates dual storage mechanisms, including structured storage for metadata and vector-based storage for semantic representations. This approach allows efficient data retrieval and supports advanced search capabilities.

Overall, the implementation is designed to ensure robustness, scalability, and efficient handling of large volumes of documents. The use of containerized deployment further enhances system portability and simplifies integration across different environments.

B. Frontend Implementation

The frontend of the Neural Docs system is designed to provide a user-friendly and interactive interface for document management and intelligent querying. It serves as the primary point of interaction between the user and the system, enabling seamless access to various functionalities such as document upload, data visualization, and chatbot interaction.

The interface is developed using a lightweight templating approach, ensuring efficient rendering of dynamic content. The design focuses on simplicity and usability, allowing users to perform operations without requiring technical expertise. The main interface consists of multiple pages, including a document upload page, a dashboard for viewing processed documents, and a chat interface for querying stored information.

The document upload feature allows users to submit files in various formats, including PDFs and images. Once uploaded, the frontend sends the data to the backend through API requests. The system provides feedback to the user regarding the status of the upload and processing stages, ensuring transparency in operations.

The dashboard component displays a structured view of stored documents along with their metadata. Users can browse, search, and filter documents based on tags, categories, or extracted attributes. This enhances usability by enabling quick access to relevant information.

A key feature of the frontend is the chatbot interface, which allows users to interact with the system using natural language. The interface captures user queries and forwards them to the backend, where they are processed using AI models. The responses are displayed in a conversational format, improving user engagement and accessibility.

The frontend communicates with the backend using asynchronous HTTP requests, ensuring real-time interaction without page reloads. Token-based authentication is implemented to secure user access and protect sensitive data.

Overall, the frontend implementation prioritizes usability, responsiveness, and seamless integration with backend services, enabling efficient and intuitive document management.

C. Backend Implementation

The backend of the Neural Docs system is implemented as the core processing unit responsible for handling application logic, managing workflows, and coordinating communication between different system components. It is designed using a server-side framework that supports scalable and modular development, allowing efficient handling of multiple concurrent requests.

The backend provides a set of API endpoints that facilitate communication between the frontend, AI services, and data storage layers. These APIs handle operations such as document upload, retrieval, processing, and query handling. The server manages incoming requests, validates data, and routes it to the appropriate services for further processing.

A key function of the backend is document ingestion and processing. When a document is uploaded, the backend validates its format and stores it in the system. It then triggers the processing pipeline, which includes text extraction, metadata generation, and indexing. This ensures that all documents are systematically analysed and prepared for retrieval.

The backend also integrates with AI modules to perform advanced operations such as text analysis and response generation. It communicates with external services through well-defined interfaces, enabling flexibility in selecting different AI providers. This abstraction allows the system to switch between local and cloud-based models without affecting overall functionality.

Authentication and security are handled at the backend level, ensuring that only authorized users can access system resources. Token-based mechanisms are used to manage user sessions and protect sensitive data.

Additionally, the backend maintains logs and metrics related to document processing and system performance. This information is useful for monitoring system behaviour and optimizing performance.

Overall, the backend implementation acts as the central orchestrator of the system, ensuring smooth integration between different modules and efficient execution of document processing workflows.

D. Database Implementation

The database implementation in the Neural Docs system is designed to efficiently store, manage, and retrieve both structured and unstructured data. The system adopts a hybrid storage approach, combining relational data storage with vector-based storage to support advanced search and retrieval capabilities.

Structured data, such as document metadata, processing logs, and user information, is stored in a relational database. This includes attributes such as document titles, tags, categories, timestamps, and extracted entities. The relational model ensures data consistency and enables efficient querying based on specific attributes.

The database schema is designed to support the lifecycle of documents within the system. Each document entry is associated with metadata that describes its content and classification. Additional tables are used to store processing history and performance metrics, allowing the system to track document processing activities.

In addition to relational storage, the system incorporates a vector database to store semantic representations of document content. These representations are generated using embedding techniques, which convert textual data into numerical vectors. The vector database enables similarity-based search, allowing the system to retrieve relevant documents based on meaning rather than exact keyword matches.

The integration of vector storage is essential for implementing Retrieval-Augmented Generation, as it allows efficient retrieval of contextually relevant information. This enhances the system's ability to provide accurate and meaningful responses to user queries.

Data consistency and integrity are maintained through proper indexing and validation mechanisms. The database is optimized to handle large volumes of documents while ensuring fast retrieval and minimal latency.

Overall, the database implementation supports both traditional data management and advanced semantic search capabilities, making it a critical component of the system architecture.

E. AI Module Implementation (MOST IMPORTANT)

The AI module in the Neural Docs system is responsible for enabling intelligent document processing and interaction. It integrates multiple artificial intelligence techniques to transform raw documents into structured, searchable, and interactive data.

The first stage of the AI pipeline involves Optical Character Recognition, which converts scanned images and PDF documents into machine-readable text. This step is essential for processing documents that do not contain embedded textual data.

Once the text is extracted, it is processed using natural language techniques to identify key information. This includes extracting entities such as names, dates, and document types, as well as generating metadata for classification and tagging. This structured information is used to organize documents and improve search efficiency.

A key feature of the AI module is the implementation of a Retrieval-Augmented Generation mechanism. In this approach, document content is converted into vector representations and stored in a vector database. When a user submits a query, the system retrieves relevant document segments based on semantic similarity.

The retrieved data is then combined with a language model to generate context-aware responses. This allows the system to provide meaningful answers rather than simple keyword-based results. The use of generative models enhances the system's ability to handle complex queries and deliver accurate information.

The AI module supports integration with both cloud-based and local language models, providing flexibility in deployment. This ensures that the system can adapt to different performance and cost requirements.

Overall, the AI module forms the intelligence core of the system, enabling automation, advanced search, and natural language interaction.

F. System Integration

The Neural Docs system is implemented as an integrated platform where multiple components interact seamlessly to provide efficient document management and intelligent querying. System integration plays a crucial role in ensuring that all modules work together in a coordinated manner.

The frontend communicates with the backend through API requests, enabling users to perform actions such as document upload and query submission. The backend processes these requests and interacts with the AI modules and database systems to perform the required operations.

The AI module is integrated as a separate service, allowing independent processing of document data. The backend communicates with this module through defined interfaces, ensuring efficient execution of tasks such as text extraction, metadata generation, and response creation.

The database layer is connected to both the backend and AI modules, allowing storage and retrieval of structured and semantic data. This ensures that all processed information is readily available for search and analysis.

The system is deployed using containerization, where each component runs as an independent service. This approach simplifies deployment and enables scalability by allowing individual components to be updated or scaled without affecting the entire system.

Overall, the integration ensures smooth data flow, efficient communication, and reliable system performance.

VI. RESULT & DISCUSSION

The Neural Docs system was implemented and evaluated to analyze its effectiveness in handling document processing, semantic search, and intelligent interaction. The system demonstrates a complete pipeline that transforms unstructured documents into structured and interactive data.

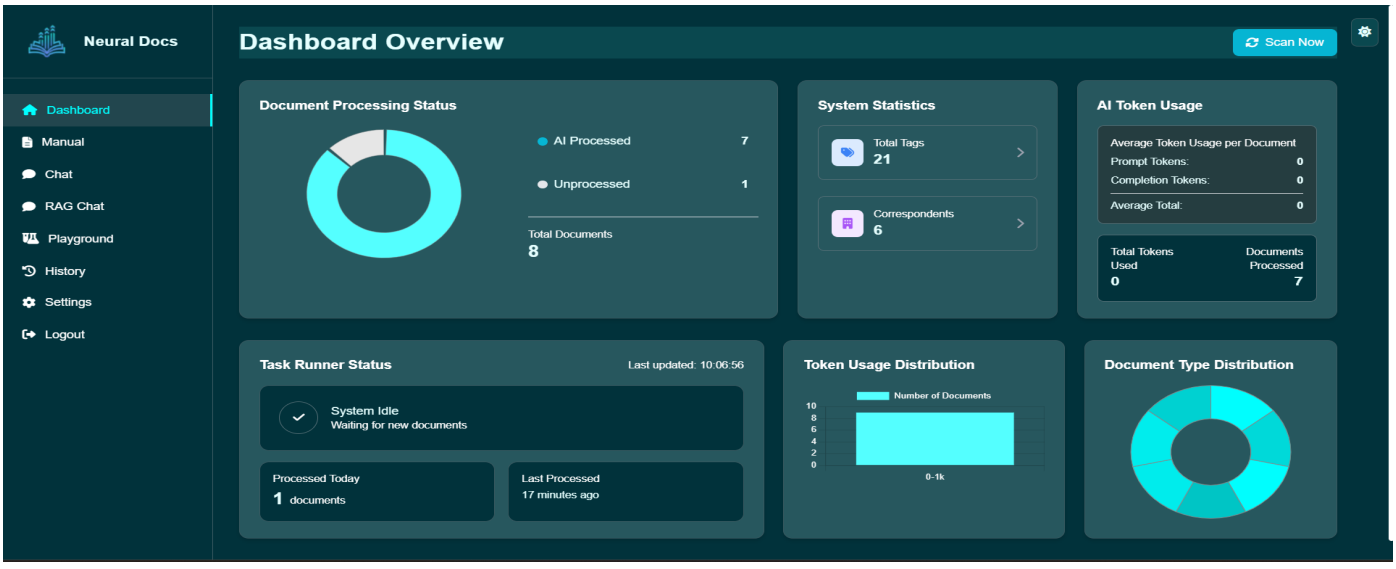


Fig. 6.1: Dashboard Interface

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

The overall user interface of the system is shown in Fig. 6.1, where the dashboard provides a centralized view of uploaded and processed documents. The interface allows users to navigate through documents efficiently, highlighting the usability and accessibility of the platform. The clean and structured layout ensures that users can easily access document information without requiring technical expertise.

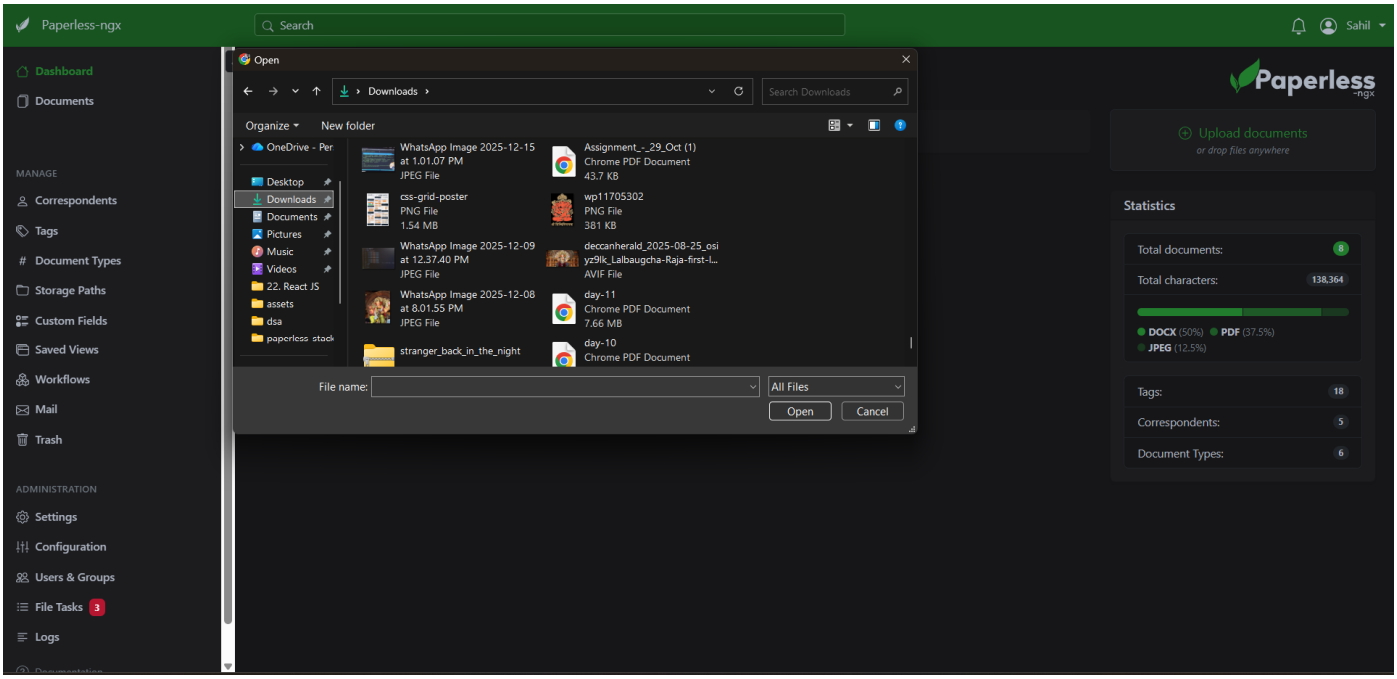


Fig. 6.2 – Document Upload Interface

The document upload functionality is illustrated in Fig. 6.2, where users can upload files in various formats such as PDF and images. Once a document is uploaded, it is processed through the backend pipeline. The system provides real-time feedback, ensuring transparency during the processing stage.

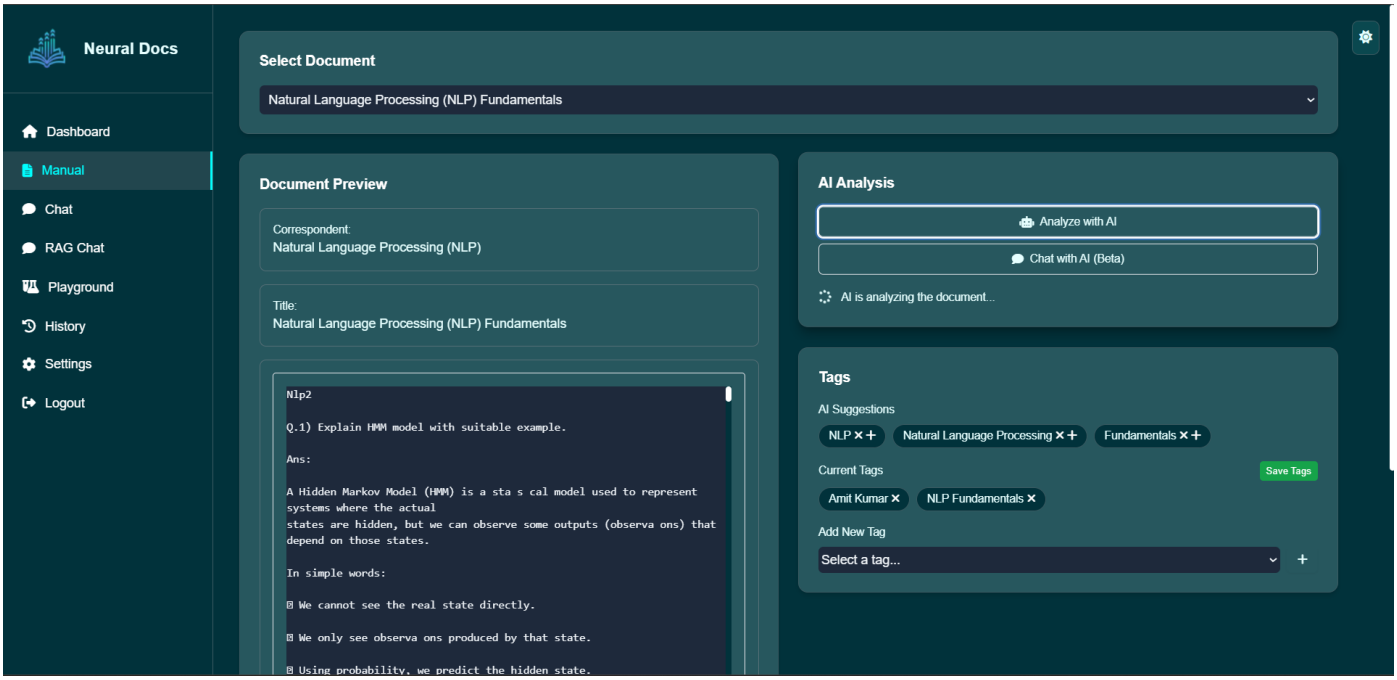


Fig. 6.3 – Extracted Metadata & Processing Output

The effectiveness of the AI processing module is demonstrated in Fig. 6.3, which shows the extracted metadata from a document. The system successfully identifies key attributes such as document title, tags, and entities. This automated extraction reduces manual effort and enhances document organization. The accuracy of metadata generation plays a crucial role in improving retrieval performance.

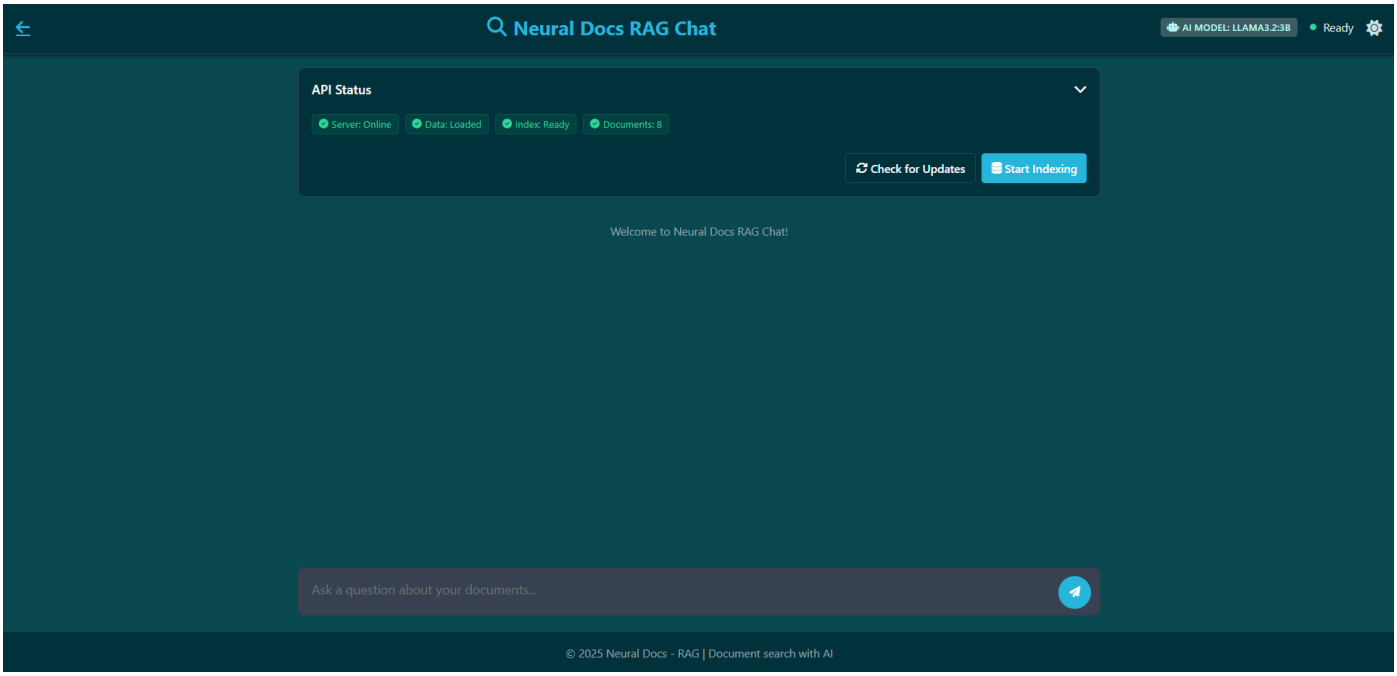


Fig. 6.3 – Extracted Metadata & Processing Output

The semantic search capability of the system is shown in Fig. 6.4, where relevant documents are retrieved based on contextual similarity rather than exact keyword matching. This demonstrates the advantage of embedding-based retrieval over traditional search methods. The system is able to identify relevant documents even when the query does not exactly match the stored content, indicating improved search efficiency. The chatbot processes natural language queries and generates context-aware responses using the Retrieval-Augmented Generation mechanism. The integration of retrieval and generative models enables the system to provide meaningful answers based on document content. This significantly improves user interaction and reduces the need for manual document searching.

Despite its advantages, the system has certain limitations. The accuracy of OCR depends on the quality of input documents, and poorly scanned images may lead to errors in text extraction. Additionally, the performance of the chatbot is influenced by the quality of retrieved data, and irrelevant retrievals may affect response accuracy. The system also requires computational resources for AI processing, which may impact performance in large-scale deployments.

Overall, the results demonstrate that Neural Docs effectively combines document processing, semantic search, and conversational AI to provide an intelligent document management solution.

VII. CONCLUSION

The Neural Docs system presents an integrated approach to document management by combining traditional storage mechanisms with advanced artificial intelligence techniques. The system successfully addresses the challenges associated with handling unstructured documents by automating tasks such as text extraction, metadata generation, and intelligent retrieval.

As demonstrated through the implementation and results, the system is capable of converting raw documents into structured and searchable information. The use of Optical Character Recognition enables processing of scanned and image-based documents, while Natural Language Processing techniques enhance the understanding and organization of extracted content. The integration of semantic search and Retrieval-Augmented Generation further improves the system's ability to provide context-aware responses to user queries.

The inclusion of a user-friendly interface, as shown in Fig. 6.1 and Fig. 6.4, enhances accessibility and interaction, allowing users to perform document-related tasks efficiently. The system architecture and data flow, illustrated in Fig. 4.1 and Fig. 4.2, demonstrate a scalable and modular design that supports efficient integration of multiple components.

However, certain limitations were identified, including dependency on OCR accuracy and computational overhead associated with AI processing. These challenges highlight opportunities for future improvements, such as enhancing text extraction accuracy and optimizing system performance.

In conclusion, Neural Docs provides a scalable, efficient, and intelligent solution for document management by integrating automation and AI-driven technologies. The system demonstrates significant potential for real-world applications in domains requiring efficient document handling and information retrieval, thereby contributing to the advancement of intelligent document processing systems.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, pp. 4171–4186, 2019.

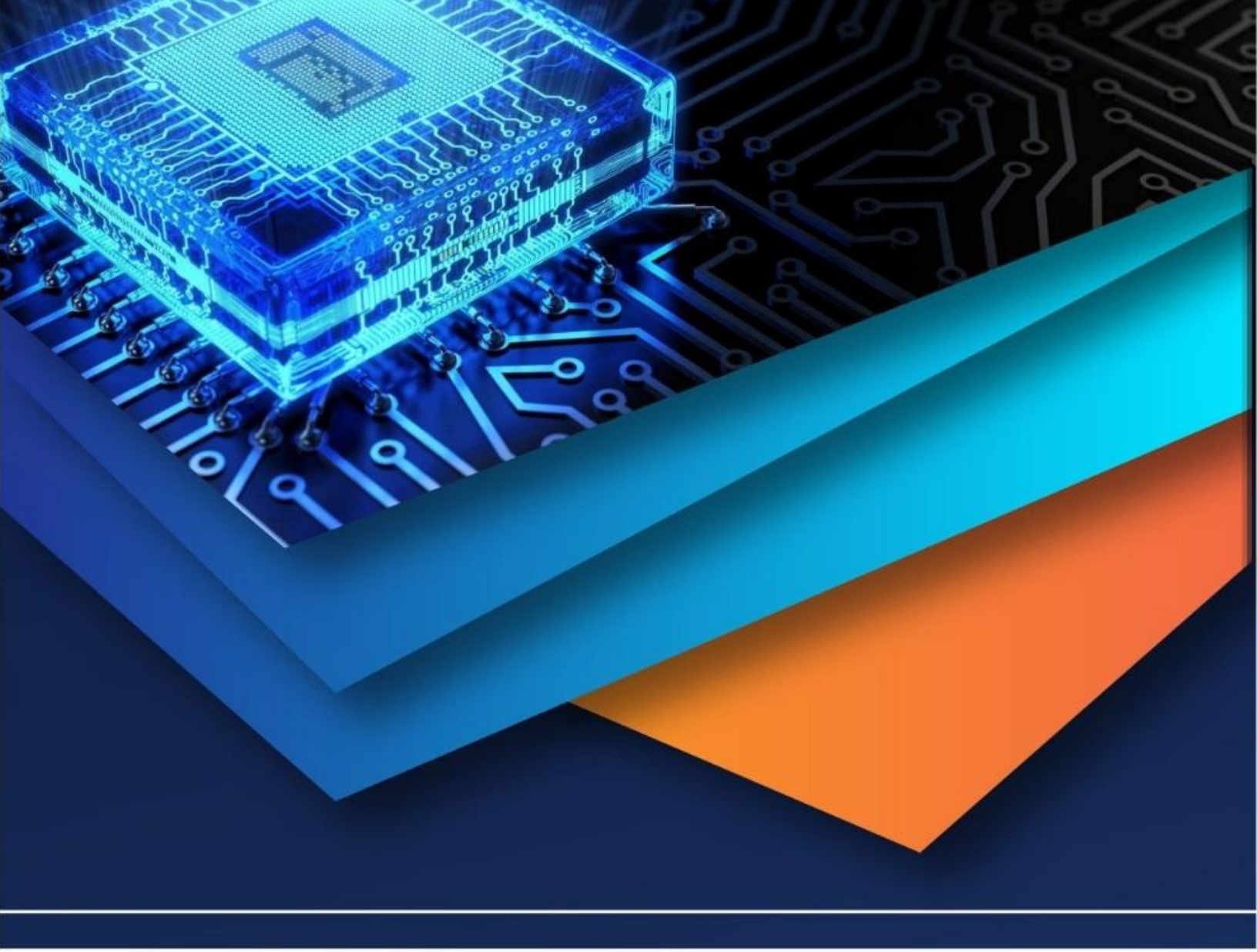
[3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT Networks," Proceedings of EMNLP, pp. 3982–3992, 2019.

[4] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," Proceedings of EMNLP, pp. 6769–6781, 2020.

[5] R. Smith, "An Overview of the Tesseract OCR Engine," Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), pp. 629–633, 2007.

[6] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2019.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY