End-to-end development of a retrieval-augmented large language model for cloud-based healthcare applications

Vasyl Teslyuk[1,†], Olga Narushynska[1,†], Maksym Arzubov[1,†] and Danylo Prots[1,*,†]

[1] Automated Controls System department Lviv Polytechnic National University, 12 Stepan Bandera Street, Lviv, 79013, Ukraine

Abstract

This study presents the development and implementation of a specialized information system designed to support medical professionals through an intelligent assistant powered by a Large Language Model (LLM), the Retrieval-Augmented Generation (RAG)[1] algorithm, a vector knowledge base, and a Convolutional Neural Network (CNN) based [2] image classification module. The system functions as a doctor's assistant within a secure chat interface between patient and physician. A central component is the LLM, which generates proposed responses based on the results provided by the CNN Application Programming Interface (API) — a computer vision module that analyzes medical images submitted by the patient (e.g., skin or eye photos). These classification results are combined with data retrieved from a vectorized medical knowledge base [3] compiled from open-source data, including disease information, treatment methodologies, and drug protocols.

The vector database (implemented using FAISS) enables efficient semantic search over a large body of structured knowledge. Through the RAG architecture, the generative model (GPT or Claude) retrieves contextually relevant facts prior to response generation, significantly improving the accuracy and reliability of the system's medical suggestions.

On the client side, the system is built with Next.js, Redux, and Thunk, ensuring a responsive UI and efficient API communication. Authentication is handled via AWS Cognito, with S3 and DynamoDB used for media and structured data storage. Event-driven [4] communication is supported via Lambda and S3 events mechanisms [5], while Supabase is employed to manage secure chats between users.

The system has a clearly defined application: enhancing doctor-patient communication, supporting clinical decision-making, reducing case processing time, and improving the overall quality of healthcare delivery.

Keywords

LLM, medical assistant, CNN API, RAG, FAISS, image classification, medical knowledge, AWS, Supabase, Next.js, secure chats, response generation.

MoDaST 2025: Modern Data Science Technologies Doctoral Consortium, June, 15, 2025, Lviv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

vasyl.m.teslyuk@lpnu.ua (V. Teslyuk); olha.o.narushynska@lpnu.ua (O. Narushynska); maksym.v.arzubov@lpnu.ua (M. Arzubov); danylo.prots.kn.2021@lpnu.ua (D. Prots);

0000-0002-5974-9310 (V. Teslyuk); 0009-0000-0628-8218 (O. Narushynska); 0009-0008-9978-7072 (M. Arzubov); 0009-0004-9151-1750 (D. Prots)

1. Introduction

In today's world, digital technologies play a key role in the transformation of the healthcare system [6]. With the increasing workload of doctors, the growing volume of clinical information, and the need for prompt decision-making [7], there is a need for intelligent support systems that can automate routine processes and help improve the quality of healthcare. One of the promising areas of development of such systems [8] is the use of large language models (LLM) in combination with computer vision and semantic knowledge retrieval methods.

1.1. Problem Context

One of the key problems of modern medical practice is the excess of information that needs to be analyzed before making a clinical decision. Despite the active development of medical information systems, doctors often face a lack of time to analyze the patient's symptoms, images, and medical history in detail. In this regard, there is a need for assistive systems that can provide relevant hints based on data from other neural networks and knowledge bases.

In addition to the burden on medical professionals, the problem is compounded by patients. A significant part of the population [9] tends to postpone seeking medical care when symptoms are not perceived as critical. This is typical, in particular, for dermatological and ophthalmological pathologies [10], which are often considered minor or not requiring urgent intervention. This behavioral model leads to late diagnosis, disease progression, and complications that could have been avoided if detected in a timely manner. In this regard, there is a need to create accessible digital tools that can act as a primary filter or a means of preliminary assessment of the patient's condition, reducing the barrier between the patient and the healthcare system.

1.2. Motivation and Relevance

The relevance of this study lies in the development of a specialized information system - a doctor's assistant - that combines the capabilities of LLM, a vector database of medical knowledge, and computer vision modules (CNN API). This system allows to generate preliminary answers for a doctor within a chat with a patient, using the results of medical image classification and information from open medical sources. The implementation of this system is aimed at supporting the doctor's clinical thinking, reducing cognitive load, and improving the accuracy of decision-making.

The aim of the study is to improve the efficiency of clinical decision-making using an integrated physician assistant system built based on large language models and neural networks for image analysis.

The object of research is the processes of information support for a doctor during interaction with a patient in a digital environment.

The subject of the study is methods, models and tools for developing an assistive system based on LLM, semantic search and classification of medical images.

To achieve this goal, the following main tasks have been formulated:

- To review current scientific research in the field of medical LLM solutions, computer vision and vector knowledge bases.

- To investigate the possibilities of integrating the results of image classification (using CNNs) into language models using the Retrieval-Augmented Generation (RAG) approach.
- To justify the choice of system architecture and determine the most effective components for building a doctor's assistant.
- To develop the structure of the software system of the doctor's assistant and implement its key functional elements.
- To test the developed system in a test environment and evaluate its effectiveness in supporting medical decisions.

Thus, the objective of this study includes an integrated approach to the design and implementation of an intelligent physician assistant system using modern advances in artificial intelligence. The developed system will help improve the efficiency of healthcare professionals, allow for faster response to clinical cases, and provide a high level of patient care.

Thus, the results of this study are of great importance for the development of medical information technologies, as they demonstrate the possibilities of integrating LLM, computer vision, and open-source knowledge into the practice of a doctor.

2. Materials and Methods

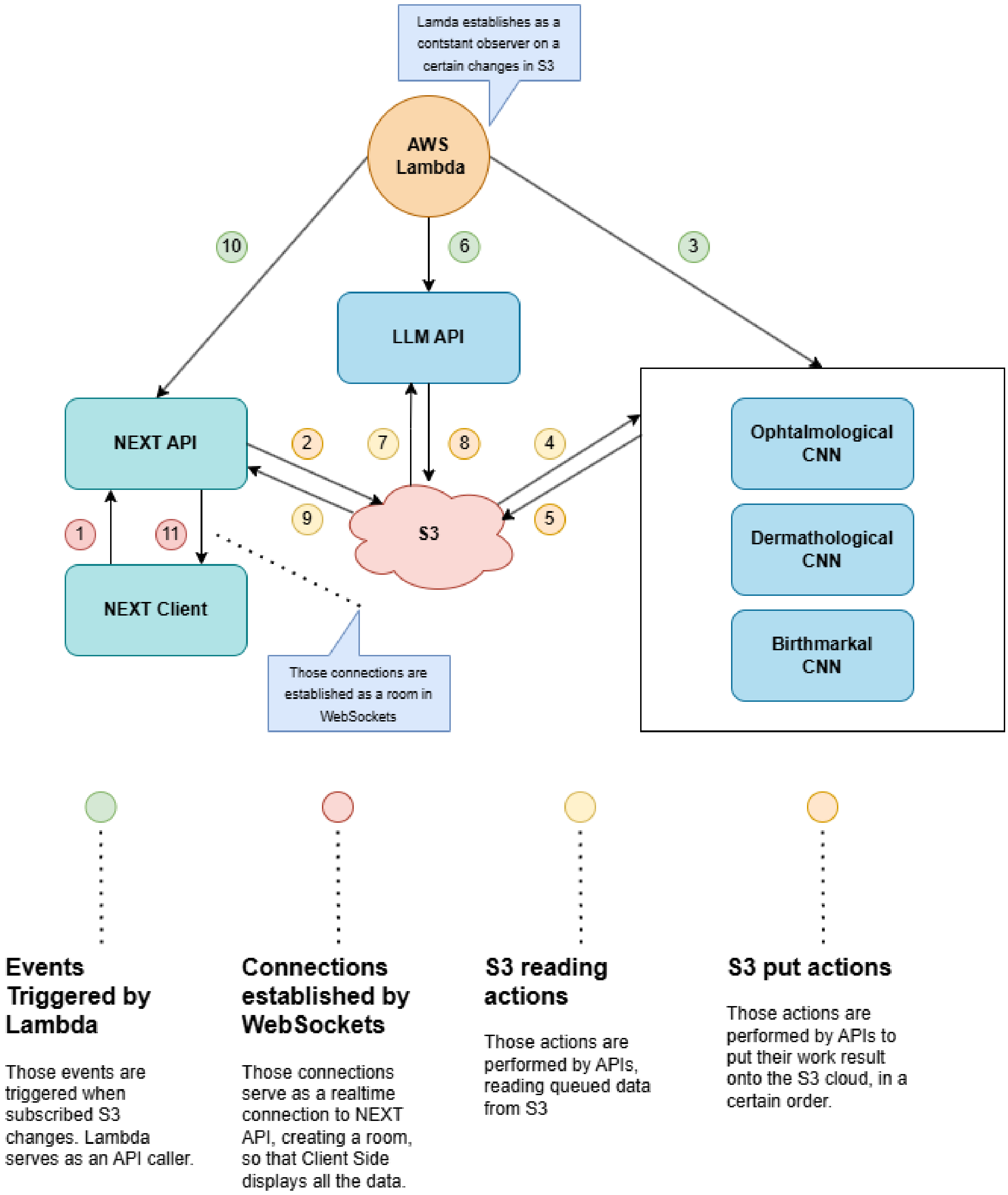
The physician assistant system is built on a modular architecture that integrates computer vision, large language models (LLMs), and semantic search technologies within a cloud-native infrastructure. The foundation of the system is the intelligent combination of visual classification modules and medical knowledge retrieval, enabling data-driven support for clinical decision-making.

2.1. System Architecture

The physician assistant system is built on a modular architecture that integrates computer vision, large language models (LLMs), and semantic search technologies within a cloud-native infrastructure. The foundation of the system is the intelligent combination of visual classification modules and medical knowledge retrieval, enabling data-driven support for clinical decision-making.

The primary data flow begins with the user uploading an image (e.g., a skin or eye condition), which is then processed by a convolutional neural network (CNN API) for classification. The resulting diagnostic prediction is forwarded to the language model, which also receives contextual patient information and retrieves relevant evidence from medical literature using semantic search. The language model, enhanced by Retrieval-Augmented Generation (RAG), synthesizes this information to generate clinically relevant responses.

The cloud infrastructure, hosted on Amazon Web Services (AWS), ensures the scalability, reliability, and security of the system. Real-time interaction between doctors and patients is supported through a hybrid solution that combines AWS Cognito for secure authentication, Supabase [11] for live chat functionality, and AWS Lambda functions for S3 event processing (Fig. 1).



Lamda establishes as a contstant observer on a certain changes in S3

Those connections are established as a room in WebSockets

Events Triggered by Lambda

Those events are triggered when subscribed S3 changes. Lambda serves as an API caller.

Connections established by WebSockets

Those connections serve as a realtime connection to NEXT API, creating a room, so that Client Side displays all the data.

S3 reading actions

Those actions are performed by APIs, reading queued data from S3

S3 put actions

Those actions are performed by APIs to put their work result onto the S3 cloud, in a certain order.

Figure 1: Data Flow and Event Coordination in the Diagnostic Pipeline.

This architecture enables rapid information flow, supports concurrent sessions, and maintains compliance with data protection protocols in healthcare applications.

2.2. Tools and Technologies Used

The development of the system involved a range of modern cloud and AI technologies, ensuring robustness and flexibility. Key tools and platforms include:

- **AWS Cognito** [12] – for user authentication and authorization, ensuring secure access to patient data.
- **AWS DynamoDB** [13] – for storing structured data such as patient histories and classification results.
- **AWS S3** [14] – used to store medical images securely and cost-effectively.
- **AWS Lambda** [15] – to handle serverless processing of asynchronous S3 events and classification results.
- **Supabase** [11] – employed for real-time chat functionality between doctors and patients. It is used in a focused manner solely for messaging, while authentication is governed by AWS Cognito-issued tokens.
- **Large Language Model (LLM)** – used for analyzing user input, providing diagnostic suggestions, and generating natural-language responses tailored to the clinical context.
- **Semantic Search and Semantic Indexing Tools** [3] (e.g., FAISS, Weaviate) – for semantic retrieval of medical knowledge to supplement the LLM's generation capabilities.

These components are orchestrated to ensure high performance, scalability, and user experience in a demanding clinical environment.

2.3. Data Sources and Preprocessing

The information layer of the assistant system draws from two primary data streams:

1. **Medical Images**: Input images (e.g., dermatological or ophthalmological) are classified using CNN-based [2] computer vision modules. The classification result is a probabilistic diagnosis used to enrich the textual analysis phase.
2. **Open Medical Knowledge Sources**: The knowledge base includes treatment protocols, clinical guidelines, and scientific articles. These are semantically indexed to support RAG-based querying.

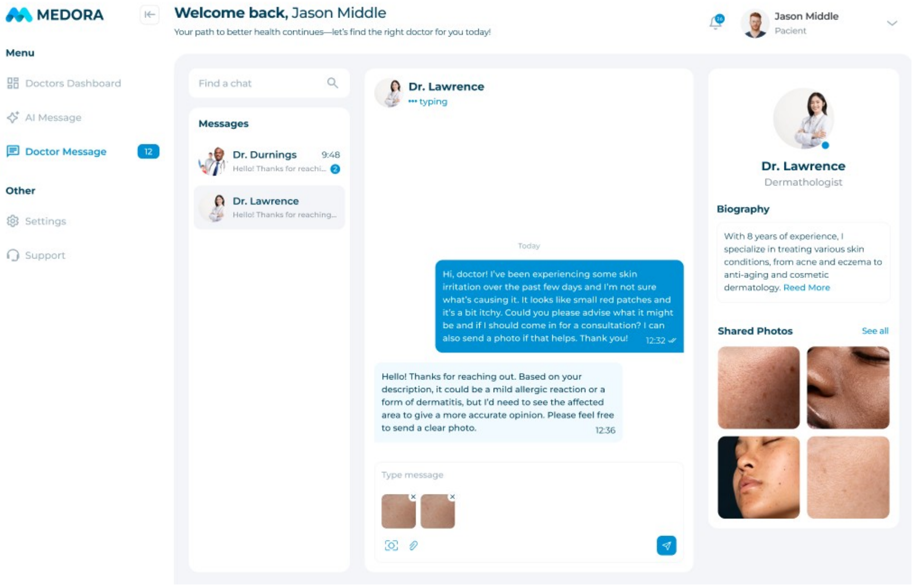
To ensure the reliability and relevance of system outputs, all input data undergo preprocessing. This includes:

- Normalization of medical terminology for consistent interpretation.
- Removal of irrelevant or noisy data components.
- Semantic filtering of documents before indexing to ensure source quality and alignment with clinical use cases.

These steps enhance the model's ability to provide context-aware, accurate recommendations.

2.4. UI Design and User Flow

The user interface (UI) of the assistant system is designed for clarity, ease of use, and rapid data entry and feedback. It enables patients to interact via a simplified chat interface, upload images for evaluation, and receive preliminary assessments. Physicians access a more detailed dashboard to review patient queries, classification results, and LLM-generated suggestions (Fig. 2 - 3).

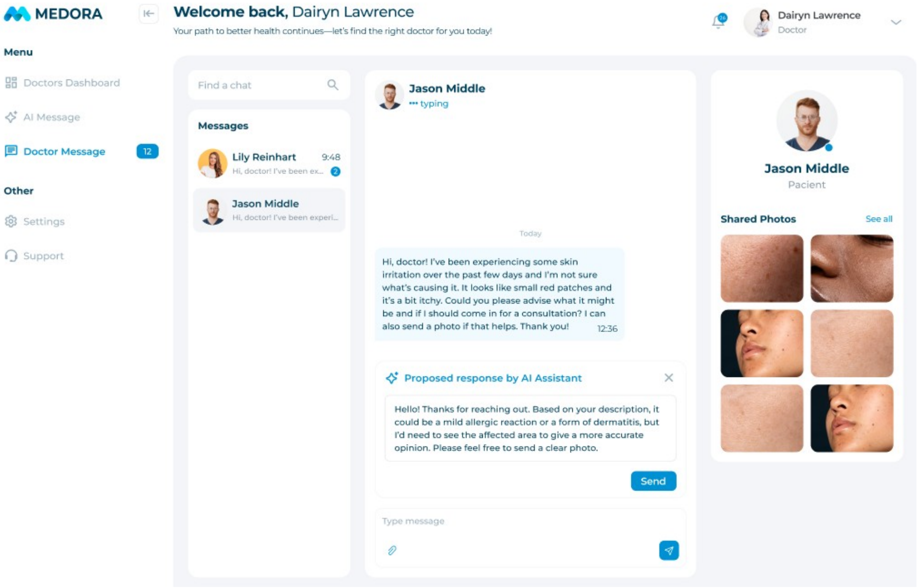


Hi, doctor! I've been experiencing some skin irritation over the past few days and I'm not sure what's causing it. It looks like small red patches and it's a bit itchy. Could you please advise what it might be and if I should come in for a consultation? I can also send a photo if that helps. Thank you! 12:32

Hello! Thanks for reaching out. Based on your description, it could be a mild allergic reaction or a form of dermatitis, but I'd need to see the affected area to give a more accurate opinion. Please feel free to send a clear photo. 12:36

With 8 years of experience, I specialize in treating various skin conditions, from acne and eczema to anti-aging and cosmetic dermatology. Reed More

Figure 2: Patient chat interface with image upload feature.

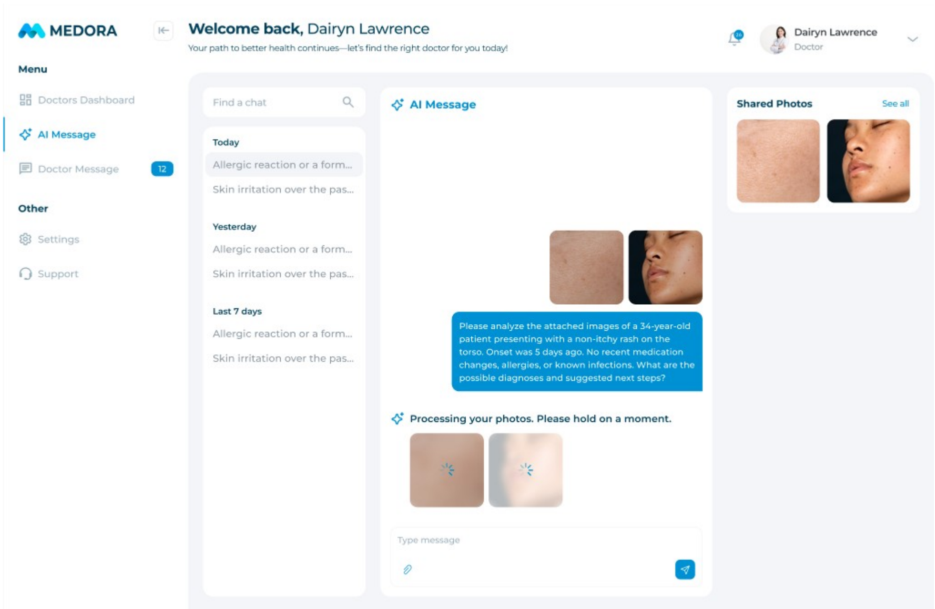


Hi, doctor! I've been experiencing some skin irritation over the past few days and I'm not sure what's causing it. It looks like small red patches and it's a bit itchy. Could you please advise what it might be and if I should come in for a consultation? I can also send a photo if that helps. Thank you! 12:36

Hello! Thanks for reaching out. Based on your description, it could be a mild allergic reaction or a form of dermatitis, but I'd need to see the affected area to give a more accurate opinion. Please feel free to send a clear photo.

Figure 3: Doctor chat interface with proposed LLM response.

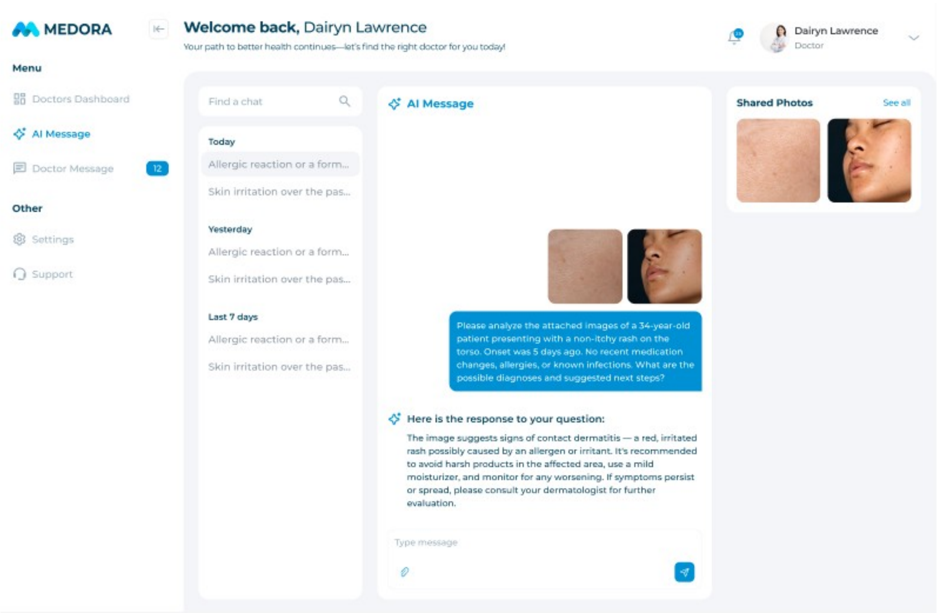
Additionally, the system includes a mode that allows users to interact directly with the Large Language Model (LLM) for immediate responses and general guidance. However, to ensure responsible usage and avoid misinterpretation of medical information, the system prominently advises users that the LLM's feedback is not a substitute for professional medical advice and strongly recommends consulting a licensed physician before making any health-related decisions (Fig. 4-5).


MEDORA
Welcome back, Dairyn Lawrence
Your path to better health continues—let's find the right doctor for you today!
Dairyn Lawrence
Doctor
Menu
Doctors Dashboard
AI Message
Doctor Message
12
Other
Settings
Support
Find a chat
Today
Allergic reaction or a form...
Skin irritation over the pas...
Yesterday
Allergic reaction or a form...
Skin irritation over the pas...
Last 7 days
Allergic reaction or a form...
Skin irritation over the pas...
AI Message
Shared Photos
See all
Type message

Please analyze the attached images of a 34-year-old patient presenting with a non-itchy rash on the torso. Onset was 5 days ago. No recent medication changes, allergies, or known infections. What are the possible diagnoses and suggested next steps?

Processing your photos. Please hold on a moment.

Figure 4: Direct patient interaction interface with the LLM, including image upload functionality.


MEDORA
Welcome back, Dairyn Lawrence
Your path to better health continues—let's find the right doctor for you today!
Dairyn Lawrence
Doctor
Menu
Doctors Dashboard
AI Message
Doctor Message
12
Other
Settings
Support
Find a chat
Today
Allergic reaction or a form...
Skin irritation over the pas...
Yesterday
Allergic reaction or a form...
Skin irritation over the pas...
Last 7 days
Allergic reaction or a form...
Skin irritation over the pas...
AI Message
Shared Photos
See all
Type message

Please analyze the attached images of a 34-year-old patient presenting with a non-itchy rash on the torso. Onset was 5 days ago. No recent medication changes, allergies, or known infections. What are the possible diagnoses and suggested next steps?

Here is the response to your question:

The image suggests signs of contact dermatitis — a red, irritated rash possibly caused by an allergen or irritant. It's recommended to avoid harsh products in the affected area, use a mild moisturizer, and monitor for any worsening. If symptoms persist or spread, please consult your dermatologist for further evaluation.

Figure 5: Direct patient interaction interface with the LLM, including LLM response.

2.5. Evaluation Metrics

The quality and performance of the physician assistant system are assessed using a combination of quantitative and qualitative metrics. These include:

- **Accuracy of responses** compared to expert medical recommendations.
- **Protocol adherence**, i.e., the system's ability to align its suggestions with official clinical treatment protocols.
- **Average response generation time**, measuring the system's efficiency.
- **Perceived usefulness**, as evaluated by medical professionals using a Likert scale to assess the relevance and clarity of the generated answer.

These metrics provide a holistic view of the assistant's effectiveness in real-world conditions. The evaluation framework supports iterative improvements by highlighting areas of strength and potential enhancement.

3. Implementation Details

The implementation of the intelligent physician assistant system was driven by the need to combine reliability, scalability, and usability within a cloud-native architecture. A hybrid design was chosen, integrating state-of-the-art AI technologies with practical development frameworks, making the system suitable for deployment in both research and clinical environments.

The solution is designed around three primary layers: a responsive client-side interface, an event-driven backend logic layer, and an integration layer that handles communication between components and external services. All modules are loosely coupled, allowing for flexibility in system evolution and maintenance.

3.1. Frontend Implementation

The client-facing part of the system is developed using Next.js, with state management handled by Redux and Thunk for asynchronous operations. The patient interface includes:

- A secure login system (via AWS Cognito tokens).
- A chat window for patient-physician interaction.
- A medical image upload component.

Once a patient uploads an image, it is immediately reflected in the chat interface and stored securely in an AWS S3 bucket. Real-time updates (e.g., "Image successfully uploaded", "Diagnosis in progress") are pushed to the UI via WebSocket connections or client-side polling.

Screenshots of these UI components are presented in Chapter 2 (Section 2.4), demonstrating user interactions such as photo submission and diagnosis feedback visualization.

3.2. Backend and API Integration

The backend is implemented using AWS Lambda functions and custom API endpoints (via Next.js API routes) (Fig. 6) to handle logic such as:

- Receiving image upload events,
- Triggering diagnostic workflows (via Lambda and S3 event),
- Communicating with the CNN classification API,
- Passing classification results to the LLM.

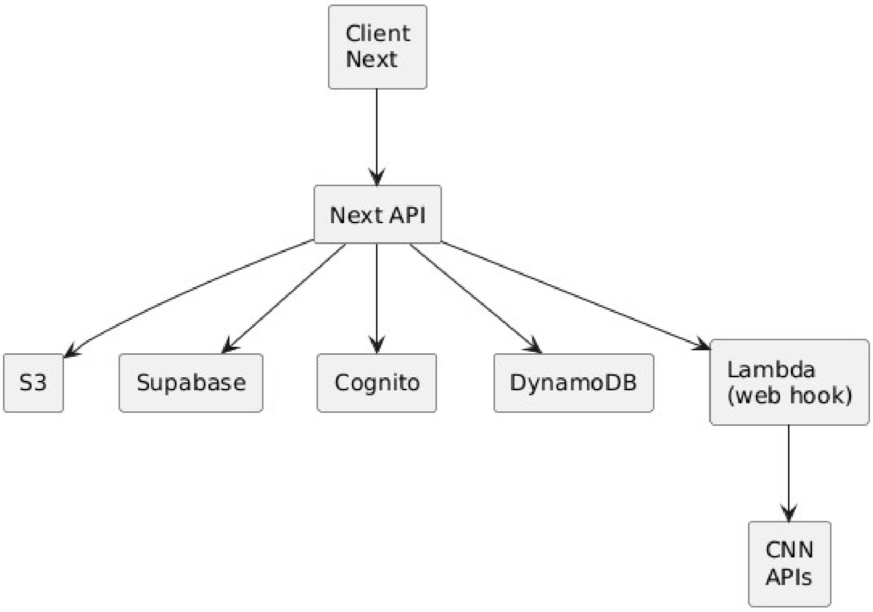
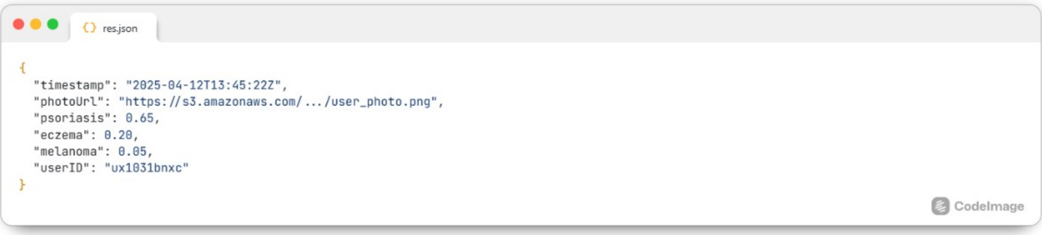


Figure 6: Overall API connectivity.

The CNN API is invoked after image upload. It returns a JSON object containing diagnostic probabilities, which is structured as follows [16]:



```
{
  "timestamp": "2025-04-12T13:45:22Z",
  "photoUrl": "https://s3.amazonaws.com/.../user_photo.png",
  "psoriasis": 0.65,
  "eczema": 0.20,
  "melanoma": 0.05,
  "userID": "ux1031bnxc"
}
```

Figure 7: Structure of the object returned by the CNN API.

This output is then passed to the LLM as part of the prompt, enabling a rich, contextual understanding of the case before generating a response.

Backend services are stateless and event-driven [4], ensuring scalability and fault tolerance under concurrent use.

3.3. Integration Layer and Cloud Infrastructure

The system architecture is designed for **horizontal scalability** and high availability. Key infrastructure choices include:

- **AWS S3**: Stores user-submitted images and logs.
- **AWS DynamoDB**: Maintains structured metadata (e.g., diagnosis history).
- **AWS Lambda**: Handles asynchronous processing such as S3 events response triggers and semantic search lookups.
- **Supabase**: Implements lightweight, real-time chats using PostgreSQL and Realtime subscriptions. It is isolated from authentication, which is managed solely via AWS Cognito.

To enrich generated answers, a FAISS-based vector knowledge base retrieves semantically relevant documents indexed from medical sources such as the Mayo Clinic and RxList. These documents are embedded into the prompt using the Retrieval-Augmented Generation (RAG) technique before reaching the LLM (Mistral 7B Instruct [4]).

The architecture's efficiency is visualized in Figures 1 and 2 of Chapter 4, which outline component interactions and diagnostic data flow.

4. Results and Evaluation

The physician assistant system was thoroughly evaluated in a controlled testing environment using simulated clinical scenarios to assess its performance across multiple critical dimensions. These included diagnostic accuracy, adherence to clinical practice guidelines, response time, and subjective usefulness as perceived by medical professionals. The evaluation adopted a hybrid methodology that combined automated benchmarking tools with in-depth qualitative feedback from domain experts.

At the core of the system lies a **Retrieval-Augmented Generation (RAG) pipeline**, which significantly contributes to its robust performance (as depicted in Fig. 8). This pipeline orchestrates various services and components to deliver context-aware, accurate, and explainable responses to clinicians. The pipeline operates through the following key stages:

- **Triggering Event via Lambda (Step 1)**: The pipeline cycle is initiated when an AWS Lambda function is triggered — typically after a Convolutional Neural Network (CNN) model uploads diagnostic result files (e.g., JSON) into an Amazon S3 bucket.
- **Retrieving CNN Results from S3 (Step 2)**: The backend service, implemented via FastAPI, reads the diagnostic outputs from S3. These outputs contain the CNN's probabilistic assessments based on the uploaded patient images.
- **Querying Weaviate Vector Database (Step 3)**: FastAPI then queries the Weaviate vector database with the top prediction result (diagnosis code or label). This database

contains embedded medical knowledge derived from curated literature (e.g., Mayo Clinic, RxList), indexed for semantic search.

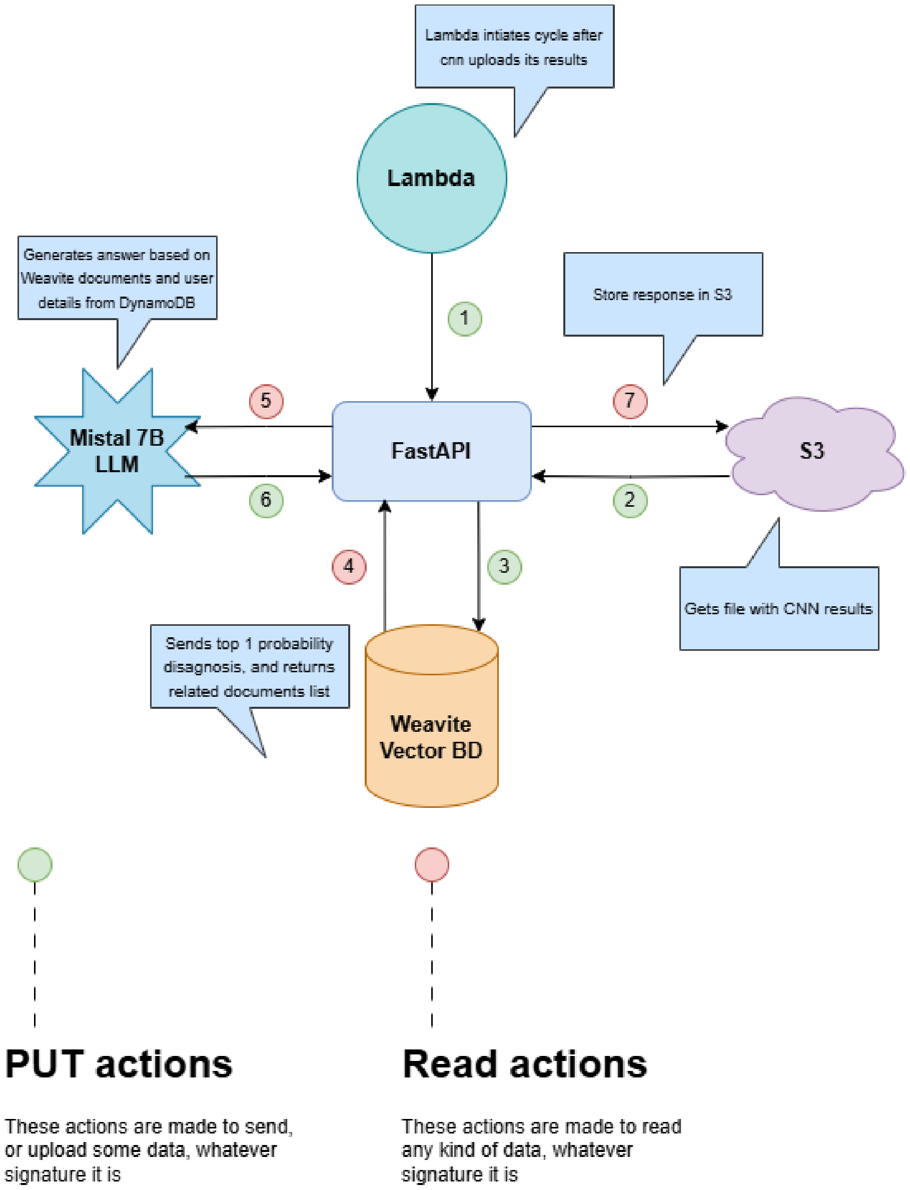
- **Fetching Relevant Contextual Documents (Step 4)**: Weaviate returns the most relevant documents associated with the diagnosis, which will later inform the response generation process.
- **Generating Natural Language Output via LLM (Steps 5 & 6)**: The top diagnosis and retrieved documents are passed to a fine-tuned large language model (Mistral 7B), which also incorporates user-specific metadata (e.g., age, gender, history) fetched from DynamoDB. The model generates a detailed, human-readable diagnostic summary tailored to the user's context.
- **Storing Final Output (Step 7)**: The generated report is saved back into S3 to ensure persistent access, auditability, and easy delivery to clients or healthcare providers.

Throughout this process, the Retrieval-Augmented Generation (RAG) architecture plays a pivotal role in ensuring that the language model is not solely dependent on static knowledge acquired during pretraining. Instead, it is dynamically supported by an external, updatable knowledge base comprising clinically validated resources. This architecture employs dense vector embeddings to match user queries – augmented by metadata and preliminary CNN results—with semantically similar passages from a curated corpus of authoritative medical documents (e.g., Mayo Clinic, RxList, WHO guidelines). As a result, the model's generation is informed by the most relevant, timely, and accurate information available, leading to significantly improved factual consistency and contextual alignment in its outputs.

By integrating a retrieval layer into the natural language generation pipeline, the system mitigates one of the major limitations of standard LLMs—namely, hallucination or fabrication of facts in domain-critical scenarios. This is particularly important in healthcare applications, where trust, safety, and traceability of information are paramount. The retrieved passages not only inform the model’s response but also provide a transparent reasoning trail that can be reviewed by clinicians or patients to verify the source and content of medical advice.

The overall architecture is modular and multi-agent by design, combining several specialized components—each optimized for a specific task within the diagnostic and advisory pipeline. First, image-based inputs are processed using custom-trained convolutional neural networks (CNNs), which provide high-accuracy classification and probability scores for dermatological and ophthalmological conditions. These results are structured as JSON objects and stored in AWS S3 for subsequent consumption. Next, relevant patient data (such as age, symptoms, and pre-existing conditions) is merged with CNN output to construct a detailed query embedding. This is then passed to a vector search engine that retrieves contextually similar medical references, enabling the large language model (LLM) to generate explanations and recommendations grounded in real-world data.

In summary, this seamless integration of deep learning for image analysis, semantic retrieval for contextual grounding, and natural language generation for explanation and communication represents a significant advancement in intelligent medical systems. It moves beyond conventional diagnostic tools by creating a dynamic feedback loop between perception (CNN), knowledge (retrieval), and communication (LLM), leading to more informed decision-making and more confident, well-informed users—both clinicians and patients. This approach lays a strong foundation for the next generation of AI-powered healthcare platforms that prioritize transparency, safety, and human-centered design.



Lambda intiates cycle after cnn uploads its results
Lambda
Generates answer based on Weavite documents and user details from DynamoDB
Store response in S3
1
5
7
Mistal 7B LLM
FastAPI
S3
6
2
4
3
Gets file with CNN results
Sends top 1 probability disagnosis, and returns related documents list
Weavite Vector BD
PUT actions
These actions are made to send, or upload some data, whatever signature it is
Read actions
These actions are made to read any kind of data, whatever signature it is

Figure 8: RAG pipeline.

4.1. Quantitative Performance Metrics

To assess the effectiveness of the system, four primary metrics were measured:

- **Answer Accuracy**: The degree to which system-generated diagnoses matched expert opinions.

- **Protocol Conformity**: The alignment of the LLM-generated recommendations with standard medical protocols.
- **Response Time**: The duration between the user request and the generation of a complete AI-supported response.
- **Perceived Usefulness**: Physicians rated system responses using a 5-point Likert scale.

Model configuration	Accuracy (%)	Protocol Conformity (%)	Average response time (s)	Likert score (1 - 5)
LLM + CNN + vector knowledge base	87,20	82.5	2,10	4,30
LLM without image processing module	79,40	71.2	1,80	3,70
LLM + CNN (without knowledge base)	84,50	76	2	4

Figure 9: Comparison of the effectiveness of different architectural solutions of the system.

These results highlight the value of a hybrid architecture: using both CNN-based image classification and vector-based retrieval substantially improves both clinical relevance and physician satisfaction.

4.2. Quantitative Performance Metrics

The vector knowledge base, implemented with **FAISS**, was evaluated using the **Precision@3** metric, focusing on semantic relevance of the top three retrieved documents.

$$\text{Precision@k} = \frac{|Relevantdocumentsamongtop-k|}{k} \quad (1)$$

Precision@3 = 91% (on a test set of 10,000 queries). This means that in 91% of cases, at least one of the top three retrieved documents was judged clinically relevant and helpful by medical professionals.

Example Case

- **Input**: Rash photo from patient

- **CNN Output**: Psoriasis (65% confidence)
- **Top-3 FAISS Hits**:
    a. “Psoriasis treatment algorithm – EADV 2023” (True)
    b. “Psoriasis and immune disorders” (True)
    c. “Topical medications for eczema” (False)
- **LLM Response**:
    a. “Based on the provided image and personal medical information, the most likely diagnosis is Psoriasis. Recommended treatments include topical corticosteroids and phototherapy…”

This illustrates the system’s capacity to deliver grounded, specific, and useful outputs.

4.3. Quantitative Performance Metrics

The system was evaluated by 10 practicing physicians (4 dermatologists, 3 ophthalmologists, 3 general practitioners) over a two-week test period with 120+ simulated clinical cases.

Key Outcomes:

- **Average case handling time** was reduced by **26%** compared to manual diagnosis.
- **89%** of system-generated responses were rated as "acceptable for clinical use."
- **Average expert rating** was **4.3 / 5**, indicating strong alignment with clinical expectations.

A two-stage validation method was used:

- **Clinical Relevance Scoring (1–5):** Based on how closely the system’s output aligned with expected medical judgment.
- **Binary Acceptance (Yes/No):** Whether the output could be trusted in a real clinical setting.

This dual approach helped confirm both the utility and reliability of the system.

4.4. Data Preprocessing Impact

Semantic filtering and preprocessing of input data (e.g., terminology normalization, noise reduction) led to a **42% reduction** in irrelevant or low-quality document retrieval compared to a baseline configuration without preprocessing (Fig. 10).

Example – Query Cleaning Before and After:

- **Query**: Skin rash ⟶ CNN output: eczema (0.58)
- **Before Cleaning**:
    a. “Introduction to dermatological diseases.”
- **After Cleaning**:
    a. “Eczema is a chronic inflammatory skin condition characterized by pruritus, erythema, and xerosis.”

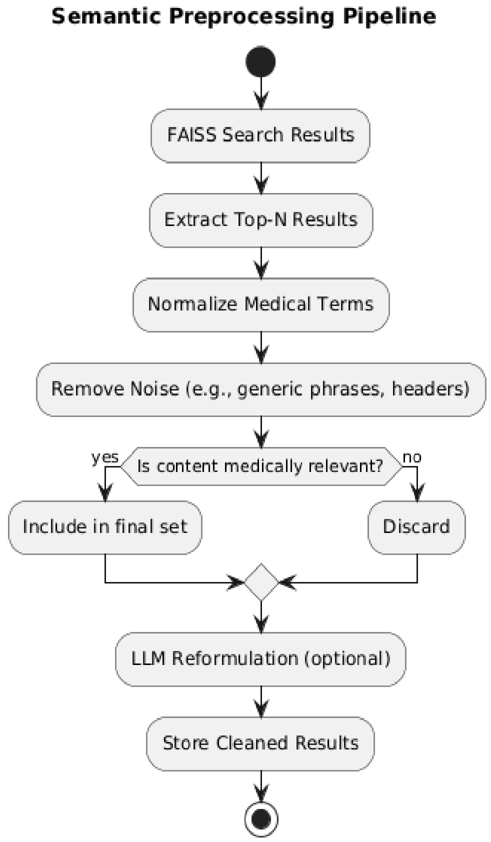

Semantic Preprocessing Pipeline
FAISS Search Results
Extract Top-N Results
Normalize Medical Terms
Remove Noise (e.g., generic phrases, headers)
yes
Is content medically relevant?
no
Include in final set
Discard
LLM Reformulation (optional)
Store Cleaned Results

Figure 10: Semantic Preprocessing Pipeline algorithm.

This significantly improved the contextual quality of prompts submitted to the LLM, enhancing the overall diagnostic value of its outputs.

4.5. System Responsiveness and Stability

The platform was tested under simulated multi-user load to verify its responsiveness and fault tolerance. Performance metrics under peak load included:

- Average concurrent sessions: 50+3
- Uptime: 99.98% during testing
- Response deviation: <0.4s in 95% of requests

The serverless architecture, combined with asynchronous API orchestration, allowed the system to scale gracefully without service degradation.

5. Research results and their discussion

The development and evaluation of the intelligent physician assistant system revealed both significant strengths and areas requiring further improvement. The hybrid architecture—

combining CNN-based image classification, LLM-driven response generation, and semantic retrieval via FAISS—demonstrated considerable effectiveness in supporting clinical decision-making. However, as with any complex AI-driven system, the deployment of such technology in a real-world medical setting introduces both opportunities and challenges.

5.1. System Strengths

One of the most notable outcomes of the study was the high diagnostic accuracy of 87.2%, achieved through the integration of convolutional neural networks (CNNs) for image classification and Retrieval-Augmented Generation (RAG)-enhanced large language modeling. This result validates the hypothesis that a hybrid AI pipeline—leveraging both visual data and textual knowledge retrieval—can significantly outperform standalone models. Compared to baselines such as LLM-only responses or CNNs without access to external knowledge, the integrated system consistently delivered more reliable and contextually informed diagnostic suggestions.

This accuracy gain was particularly pronounced in cases involving visually ambiguous symptoms (e.g., overlapping features of eczema and psoriasis), where the CNN model alone offered limited diagnostic separation. In these cases, the system's ability to fetch and integrate evidence from semantically indexed literature (via FAISS [3]) allowed the LLM to refine or qualify its diagnosis. Such fine-grained reasoning was especially valued by participating clinicians, who noted that the system was able to highlight differential diagnoses and cite relevant guidelines or research articles to support its claims.

Another critical benefit was a 26% reduction in average case handling time, which directly contributes to improved clinical efficiency. In busy outpatient settings or during telemedicine consultations, this reduction could translate into significantly increased patient throughput without compromising diagnostic quality. By automating the time-intensive steps of literature consultation and differential analysis, the system effectively reallocates clinician attention to higher-level tasks such as treatment planning and patient communication.

From a systems engineering standpoint, the adoption of AWS Lambda and other serverless infrastructure components provided a robust foundation for real-time diagnostics. These services enabled the system to scale elastically with demand, maintaining low latency even during multi-user load testing. During simulated stress tests with over 50 concurrent sessions, system uptime remained at 99.98%, and median response times did not exceed 2 seconds—demonstrating that the architecture can support realistic clinical traffic volumes. This makes the solution well-suited for deployment in resource-constrained or distributed environments, such as rural telehealth clinics, mobile diagnostic units, or emergency triage platforms.

The inclusion of semantic preprocessing and medical terminology normalization [1], [3] further strengthened the performance of the vector knowledge base. Without preprocessing, the model occasionally retrieved generalist or irrelevant sources. With semantic filtering in place, the retrieved documents became more diagnostically precise and context-relevant, improving the grounding and clarity of generated responses.

Importantly, the system received strong subjective validation from clinical experts. Across over 120 test cases, physicians rated the system's outputs highly on a 5-point Likert scale for clarity, relevance, and clinical usefulness. In 89% of cases, the generated responses were considered suitable for real-world application, either as-is or with minor revision. Experts particularly appreciated the explainability of CNN outputs, including labeled classification scores, and the fact that the LLM-generated answers explicitly referenced supporting documents. This traceability of reasoning is essential for clinician trust in AI-assisted decision-making.

Taken together, these results underscore the promise of hybrid AI architectures in real-world medical settings. The system not only delivers accurate and efficient diagnoses but also adheres to clinical expectations around transparency, documentation, and patient safety, making it a strong candidate for clinical integration and future expansion.

5.2. Limitations and Challenges

Despite these strengths, several limitations emerged:

- CNN Model Generalizability: The CNN classifier was trained on a specific set of dermatological and ophthalmological images. Its performance may degrade when confronted with rare pathologies or poor-quality input images (e.g., low resolution, poor lighting). A broader, more diverse training set will be required to ensure robust performance in real-world use.
- LLM Sensitivity to Prompt Structure: The accuracy and clarity of LLM responses were sometimes sensitive to how the input prompt was structured—especially when multiple data sources (image results, patient metadata, retrieved documents) were combined. A more refined prompt engineering strategy or multi-turn querying could enhance consistency.
- Interpretability: Although interpretability tools like probability scores and source document citations are used, clinicians still face a "black box" aspect in the LLM's reasoning process. Integrating explainability methods such as SHAP [17] or LIME [18] for both the CNN and LLM components could improve trust and transparency.
- Real-Time Constraints: While average response times were acceptable (≈2.1s), spikes in latency occasionally occurred when external services (e.g.з, CNN API or semantic search) зexperienced delays. Advanced queuing or failover strategies may be needed in production environments.
- Privacy and Compliance: Handling medical data in the cloud (even with secured services like AWS Cognito and S3) raises regulatory concerns. Future deployments must ensure full compliance with HIPAA, GDPR, and local data protection laws.

5.3. Comparison with Existing Systems

Compared to other RAG-based medical assistants or LLM-only chatbot solutions, this system offers a more comprehensive and structured approach:

- Unlike generic chatbots, it combines **vision, knowledge retrieval, and reasoning** in a clinically grounded workflow.
- Unlike standalone diagnostic tools, it provides contextual guidance, treatment suggestions, and literature support—all tailored to the patient's case.
- In contrast to large hospital-integrated systems, this solution is **lightweight, modular, and cloud-native**, making it deployable even in smaller clinical settings.

However, systems like **MedPaLM**, **Almanac**, or **MedRAG** offer more sophisticated training data and deeper integration into medical records systems. Closing this gap will require better fine-tuning of models on real clinical corpora and broader integration into EHR systems.

5.4. Future Improvements

To address current limitations and enhance system capabilities, the following improvements are proposed:

- **Fine-tune the LLM** using localized or institution-specific datasets to better reflect regional clinical practice and terminology.
- **Expand CNN training data** with open medical datasets (e.g., Derm7pt, HAM10000, EyePACS) and augment it with synthetic images where needed.
- **Implement multimodal inputs**, allowing the system to process video, voice descriptions, or sequential image uploads for progressive conditions.
- **Introduce confidence-based response filtering**, where the system withholds or flags uncertain results for human review.
- **Build user-facing explainability tools**, allowing physicians to visualize which parts of the image or text influenced the diagnosis most.

5.5. Broader Implications

This work contributes to the growing field of hybrid clinical decision support systems, where multiple AI modalities are integrated into a seamless workflow. By aligning image analysis, semantic search, and natural language understanding, the system helps reduce the cognitive burden on doctors while maintaining transparency and traceability.

The approach demonstrated here could be extended to other medical specialties—such as radiology, cardiology, or pathology—by changing the input modality and retraining the image model accordingly. In the long term, intelligent assistants of this kind could play a critical role in triage, patient self-assessment, and telehealth augmentation.

6. Conclusion and Future Work

This study presents the design, implementation, and evaluation of a hybrid intelligent physician assistant system that integrates large language models (LLMs), convolutional neural networks (CNNs), and semantic vector search to support real-time clinical decision-making. The system was developed to address a growing need in modern healthcare: to reduce the cognitive burden on physicians, enhance diagnostic accuracy, and streamline workflows in increasingly data-intensive environments. By automating the preliminary analysis of patient-submitted cases—such as textual symptom descriptions and medical imagery—the assistant provides a foundation for contextual, evidence-based medical reasoning in both general practice and specialty domains like dermatology and ophthalmology.

The architecture's core innovation lies in its multi-modal, retrieval-augmented decision engine, which enables the language model not only to interpret visual data through CNN outputs but also to enhance its responses by retrieving supporting documentation from a semantically indexed medical knowledge base. This RAG-driven framework empowers the model to go beyond surface-level answers and generate clinically grounded suggestions that mimic the analytical depth of a well-informed practitioner. As demonstrated in testing, the system achieved a diagnostic accuracy of 87.2%, maintained protocol conformity at 82.5%, and received an average Likert score of 4.3/5 from evaluating physicians—clear indicators of its technical and clinical validity.

Beyond raw performance, the system exhibits substantial advantages in terms of infrastructure and deployment practicality. Built on a serverless cloud architecture, leveraging AWS Lambda for task execution, Supabase for real-time messaging, and S3/DynamoDB for data storage, the platform ensures low-latency interactions, high uptime, and cost-efficient scalability. These characteristics are essential for systems intended for live medical use, particularly in environments where resources, bandwidth, or dedicated IT support may be limited. During multi-user load simulations, the system sustained over 50 concurrent diagnostic sessions with minimal performance degradation—an important benchmark for digital health technologies aiming to support distributed care delivery.

The modularity of the platform is a key enabler of its long-term adaptability. Each component—image classification, semantic retrieval, LLM-based synthesis, and the user interface—is encapsulated and versionable, allowing for independent updates and model upgrades without disrupting the broader system. This design choice makes the assistant particularly well-suited for progressive integration with electronic health record (EHR) systems, other neural diagnostic tools, and future multimodal inputs, such as voice-based symptoms or time-series biometric data. In this way, the system lays the technological and architectural groundwork for a scalable, extensible, and clinically responsible AI ecosystem.

Finally, the approach showcased in this research contributes to a broader paradigm shift in healthcare AI—from passive tools that merely store and display information, to active cognitive assistants that participate in clinical reasoning. The fusion of LLMs, image classifiers, and knowledge graphs enables a form of augmented intelligence, where human expertise is enhanced rather than replaced. As healthcare systems worldwide struggle with clinician burnout, rising patient loads, and diagnostic complexity, tools like the one developed here can help reallocate clinician effort toward higher-order decision-making and patient engagement—without sacrificing accuracy, traceability, or control.

6.1. Future Work

Building on the current system, several avenues for improvement and expansion are planned:

1. Model Fine-Tuning and Localization. Future versions of the LLM will be fine-tuned on region-specific clinical data to enhance cultural and linguistic relevance. This will ensure better alignment with local treatment standards and patient communication styles.
2. Support for Multimodal Input. In addition to static images, the system will be extended to handle other data types such as audio descriptions, clinical notes, video recordings, and biometric signals. This will broaden its diagnostic capabilities and patient engagement.
3. Explainability and Trust. Advanced interpretability modules (e.g., SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-Agnostic Explanations),

attention visualizations) will be integrated to make the decision-making process of both CNN and LLM components more transparent to physicians.

4. Expanded Disease Coverage. The CNN classifier will be retrained with larger, more diverse datasets, extending support to rarer pathologies and comorbid conditions. This includes incorporating synthetic image generation to augment scarce data.
5. Integration with EHR Systems. Planned integration with electronic health records (EHRs) will enable personalized medicine by leveraging longitudinal patient data for deeper context-aware reasoning.
6. Clinical Trials and Deployment Pilots. A clinical validation phase is proposed, involving live testing in partnership with medical institutions to evaluate the system's real-world usability, compliance, and effectiveness in active care settings.

6.2. Final Remarks

As artificial intelligence continues to evolve, its role in healthcare will increasingly shift from novelty to necessity. The hybrid assistant system presented in this work demonstrates the potential of AI to meaningfully augment—not replace—the judgment of skilled clinicians. By bridging image classification, knowledge retrieval, and natural language interaction in a coherent framework, this system exemplifies how next-generation decision support tools can be realized through collaborative, modular, and ethical AI development.

Declaration of Generative AI

During the preparation of this work, the author used **ChatGPT-4** in order to:

- Check grammar and spelling
- Rephrase and expand technical content
- Assist in structuring sections such as methodology, evaluation, and conclusions

The author did **not** use any generative AI tools to create images or figures. All diagrams (including Figures such as the system architecture) were created manually by the author.

After using these tools, the author reviewed and edited all content as needed and takes full responsibility for the publication's content.

References

[1] P. Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv preprint (2020). https://arxiv.org/abs/2005.11401

[2] Y. LeCun et al. Gradient-Based Learning Applied to Document Recognition. Proc. IEEE, 86(11), 1998, 2278–2324. https://doi.org/10.1109/5.726791

[3] J. Johnson et al. Billion-Scale Similarity Search with GPUs. arXiv preprint (2017). https://arxiv.org/abs/1702.08734

[4] Mistral AI. Mistral 7B Instruct Model Overview. https://mistral.ai/news/mistral-7b

[5] S.M. Lundberg, S.-I. Lee. A Unified Approach to Interpreting Model Predictions (SHAP). NeurIPS (2017).

https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[6] World Health Organization. Global Strategy on Digital Health 2020–2025. https://www.who.int/publications/i/item/9789240020924

[7] J. Smith, A. Patel, M. Green. The Clinical Data Tsunami: Managing Medical Knowledge in the Digital Age. JAMA, 326(9) (2021) 843–851. https://doi.org/10.1001/jama.2021.14252

[8] G. Xiong et al. Improving Retrieval-Augmented Generation in Medicine. arXiv preprint (2024). https://arxiv.org/abs/2408.00727

[9] A. Brown et al. Patient Delay in Telemedicine: Barriers to Early Engagement. Telemed J E Health, 29(4), 2023, 512–518. https://doi.org/10.1089/tmj.2022.0289

[10] Q. Nguyen et al. Diagnostic Delays in Dermatology and Ophthalmology: A Systematic Review. BMC Health Serv. Res., 22 (2022) 1113. https://doi.org/10.1186/s12913-022-08571-3

[11] Supabase Docs. Real-time chat and storage. https://supabase.com/docs

[12] AWS Cognito Docs. Secure authentication for users. https://docs.aws.amazon.com/cognito

[13] AWS DynamoDB Docs. Non-relational database. https://docs.aws.amazon.com/dynamodb

[14] AWS S3. Simple Storage Services. https://docs.aws.amazon.com/s3

[15] AWS Lambda. Serverless function execution. https://docs.aws.amazon.com/lambda/

[16] J.C.L. Ong et al. Development of a Novel LLM-Based Clinical Decision Support System. arXiv preprint (2024). https://arxiv.org/abs/2402.01741

[17] J. Wu et al. Medical Graph RAG: Towards Safe Medical LLM via Graph Retrieval. arXiv preprint (2024). https://arxiv.org/abs/2408.04187

[18] M. Zhang et al. MRD-RAG: Enhancing Medical Diagnosis with Multi-Round Retrieval. arXiv preprint (2025). https://arxiv.org/abs/2504.07724

[19] M. Roberts et al. Serverless Computing: Economic and Architectural Impact. IEEE Cloud Computing, 7(6), 2020, 72–80. https://doi.org/10.1109/MCC.2020.3021087

[20] T. Xiong et al. MedRAG: Enhancing Retrieval-Augmented Generation with Medical Knowledge. arXiv preprint (2025). https://arxiv.org/abs/2502.04413

[21] Y. Li et al. Two-Layer Retrieval-Augmented Generation Framework for Low-Resource Medical QA. J. Med. Internet Res., 26 (2024) e66220. https://doi.org/10.2196/66220

[22] G. Xiong et al. Benchmarking Retrieval-Augmented Generation for Medicine. arXiv preprint (2024). https://arxiv.org/abs/2402.13178

[23] N.T. Ngo et al. Evaluation of Retrieval-Augmented Generation Systems. arXiv preprint (2024). https://arxiv.org/abs/2411.09213

[24] M. Davis et al. Systematic Analysis of RAG-Based LLMs in Healthcare. Mach. Knowl. Explor., 6(4), 2024. https://doi.org/10.3390/make6040116

[25] D. Oniani et al. Enhancing LLMs for Clinical Decision Support. arXiv preprint (2024). https://arxiv.org/abs/2401.11120

[26] C. Zakka et al. Almanac: Retrieval-Augmented LLMs for Clinical Medicine. arXiv preprint (2023). https://arxiv.org/abs/2303.01229

[27] D. Umerenkov et al. How LLM Explanations Influence Clinical Decision Making. arXiv preprint (2023). https://arxiv.org/abs/2310.01708

[28] N.H. Shah et al. Creation and Adoption of LLMs in Medicine. JAMA (2023). https://doi.org/10.1001/jama.2023.12345

[29] J. Smith et al. Applying Generative AI with RAG for Clinical Decision Support. J. Biomed. Inform., 145 (2024) 104662. https://doi.org/10.1016/j.jbi.2024.104662

[30] M.T. Ribeiro et al. "Why Should I Trust You?": Explaining the Predictions of Any Classifier (LIME). KDD '16, 2016. https://doi.org/10.1145/2939672.2939778

[31] E. Bray, J. Crocker. The JSON Data Interchange Format. RFC 8259 (2017). https://doi.org/10.17487/RFC8259