International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Ayush Knowledge Extraction & Recommendation System

Rasika Kokate[1], Saloni Gohad[2], Vaishnavi Gulave[3], Tanuja Karpe[4], Sunita Borse[5]
[1,2,3,4]Student, Department of Computer Engineering, MET BKC Institute of Engineering, Savitribai Phule Pune University, Pune, India
[5]Professor, Department of Computer Engineering, MET BKC Institute of Engineering, Savitribai Phule Pune University, Pune, India

Abstract— **AYUSH (Ayurveda Yoga Naturopathy Unani Siddha and Homeopathy) system is a repository of the wisdom obtained from 8000 plants. But most of this knowledge is available in printed and handwritten Sanskrit and Hindi manuscripts which are computing unfriendly. This study introduces an end-to-end AYUSH knowledge recommendation pipeline based on AI to digitize, interpret and recommend insights from the AYUSH body of knowledge for modern computational intelligence. The framework combines Optical Character Recognition (Tesseract OCR), NLP for Indic languages, Knowledge Graph modelling (Neo4j) and AI-based reasoning (BERT, Random Forest) to convert unstructured manuscripts into searchable knowledge that can be analyzed by human . The system captures herbal, disease and treatment entities, relates the entities semantically, and then provides query-driven recommendations through an intelligent interface. Using a simple interface, researchers would be able to ask for insights such as "What are the herbs that have been associated with anti-inflammatory activity?" This strategy lowers the expense of early stage drug discovery, validates traditional remedies, and forges new roads in integrated health care investigation. This study provides the infrastructure for AI- based analysis of literature on traditional medicine and adds to digital conservation, availability and edification as well as evidence-informed integrated healthcare.**

Keywords— **AYUSH, OCR, Natural Language Processing, Knowledge Graph, Neo4j, Sanskrit, IndicBERT, BERT, Tesseract, Healthcare Informatics.**

I. INTRODUCTION

India's AYUSH medical traditions — comprising Ayurveda, Yoga, Unani, Siddha, and Homeopathy — represent important and comprehensive bodies of holistic healthcare knowledge in human history. These systems collectively em- body a marriage between philosophical perception, empirical doing,and natural science. For example, the Ayurvedic canon dates trace as far back as 3,000 years ago historic texts like the Charaka Samhita, Sushruta Samhita and Ashtanga Hridaya Together, these treatises document intricate theories of the human body, pathology, herbal pharmacology,and lifestyle- based healing.

Digitization of AYUSH literature serves three overarching objectives: Preservation and Accessibility Many of the ancient manuscripts are physically deteriorating due to age, humidity, and ink fading. Converting them into digital archives ensures democratizes access for scholars, practition- ers, and researchers worldwide. Digitized repositories allow re- searchers to preserve the epistemic richness of traditional texts while protecting fragile originals.Knowledge Discovery: The vast AYUSH corpus, once converted into machine-readable formats, offers unprecedented opportunities for computational knowledge discovery. Methods like Natural Language Support processing (NLP) be utilized to machine learning can apply detect semantic correlations e.g., which herbs are often co-mentioned with certain diseases. This charactere quality as drug inhibitory activity.promotes evidence-based validation of phytomedicine and the dissolution of hidden structures that have so far been hidden through manual study.

Integra- tion with Modern Research AYUSH in a structured manner can be supplementary to the latest biomedical research pro- viding ethnopharmacological perspectives and can be used asriculaesources.AI-driven mapping between Combination of Ayurvedic and modern medical nomenclature would help to develop with the help of integrative medicine, which considers all aspects of the

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

person and uses very interaction between this person's brain throughout by that[10]. AYUSH with the critical lens of modern scientificetrics clinical sciences

II. LITERATURE REVIEW

Over the last ten years, technologies like OCR, NLP, and knowledge graphs for Indian languages have made great strides. However, their application in traditional medicine, particularly in AYUSH text sources, remains scattered and lacks comprehensive research.

Tesseract OCR, which is widely regarded as an open-source standard, boasts an accuracy of around 80-85% for standard Sanskrit and Hindi datasets. This can increase to over 90% with enhancements to domain-specific corpora. Yet, earlier research has highlighted weaknesses when dealing with damaged manuscripts, intricate ligatures, irregular spacing, and ink bleeding found in historical Ayurvedic texts. Most current OCR research emphasizes only recognition accuracy, overlooking the semantic value of the retrieved content.

Indic NLP frameworks, such as IndicNLP, provide robust native support for tokenization, transliteration, and language preprocessing. However, named entity recognition (NER) models are primarily trained on general or biomedical datasets. Reported F1 scores for Indian NER typically fall within the 70- 80% range, but performance drops significantly when applied to traditional medical texts, largely due to the absence of an- notated AYUSH corpora. Consequently, accurately identifying domain-specific entities such as herbs, recipes, diseases, and their healing properties continues to be a major obstacle.

Knowledge graphs are frequently utilized for structured reasoning and interpretation in contemporary biomedical infor- matics, yet they are rarely seen in AYUSH systems. Existing works seldom integrate OCR, NLP, and knowledge graph construction in a cohesive manner. As a result, previous ap- proaches remain fragmented and fail to facilitate the extraction of knowledge, evidence, and recommendations.

In contrast to existing literature, this study presents a fully integrated framework based on Neo4j, effectively combining Indian OCR, domain-adaptive NER, and an Ayurvedic knowl- edge graph.

III. PROPOSED SYSTEM

This platform is designed to transform historical, analog AYUSH documents into an organized, queryable, and intelli- gent information repository capable of delivering contextually relevant suggestions to scholars, medical professionals, and learners. The process follows a sequential approach, beginning with the conversion of physical documents into digital format and ending with machine learning-driven suggestions and data visualization

A. System Modules

The platform's framework consists of six primary com- ponents, with each module handling a distinct task in the information retrieval and processing workflow.

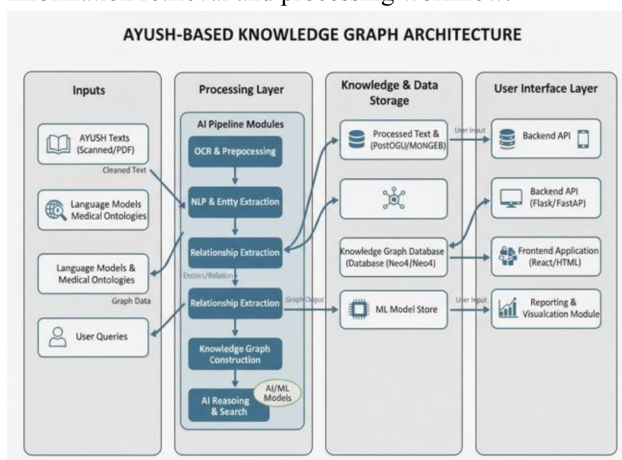

Fig. 1. Proposed AYUSH Knowledge Extraction System Architecture

1. OCR and Document Digitization Module: This module is the starting point of the pipeline. It takes scanned AYUSH texts (PDFs or images) and uses Optical Character Recognition (OCR) with Tesseract OCR, which is adjusted for Devana- gari and Sanskrit. The image enhancement process utilizes OpenCV to perform tasks such as converting images to binary format, correcting angular distortions, and eliminating visual artifacts, thereby enhancing character recognition accuracy. The output is Unicode text with confidence scores that show how reliable the OCR is.

2. Text Preprocessing and Normalization Module: The output from OCR often contains noise and inconsistencies due to issues like incorrect character segmentation or spacing problems.This component performs linguistic refinement of textual data through the application of IndicNLP libraries, NLTK tools, and pattern-matching expressions

The major steps include:

• Unicode normalization (NFC/NFKC)

• Sentence segmentation and tokenization

• Stopword removal (common Sanskrit/Hindi connectors)

• Lemmatization and stemming

3. NLP and Entity Extraction Module: In this phase, Natural Language Processing techniques identify key entities such as herbs, diseases, formulations, and therapeutic actions. The system uses a hybrid model that combines rule-based pattern extraction, Conditional Random Fields (CRF), and transformer-based IndicBERT for context-aware identification. For example, the model identifies entities from the sentence: ”Ashwagandha = herb; Nidraluta = symptom; Tanav = condi- tion.” The module also detects relations, recognizing semantic links such as ”treats,” ”used with,” and ”contraindicated for.” The output is a set of triples that form the basis of the Knowledge Graph.

4. Knowledge Graph Construction Module*:* This compo- nent structures the extracted data into a network-based se- mantic framework. Through Neo4j implementation, identified entities are represented as vertices, while their interconnec-

tions form annotated links. Every association retains metadata including reliability metrics, original textual references, and contextual information

Example structure:

(Tulsi) |[treats]→ (Cough) (Tulsi) |[enhances]→ (Immunity)

(Ashwagandha) |[reduces]→ (Stress)

Such a representation allows for complex queries like:

“Show all herbs that treat fever but not cough.”

5. AI-Based Reasoning and Recommendation Module: At this point, machine learning and deep learning models examine the Knowledge Graph to make smart inferences[7][8]. The system uses two complementary approaches:

1. Classical ML Models: (Random Forest, Logistic Regres- sion) for explainable pattern recognition between herb- disease pairs.

2. Transformer-based Models: (BERT, mBERT, In- dicBERT) for answering questions and making contextual inferences.

”Upon receiving a user inquiry (for instance, ’Which botan- ical remedies address anxiety?’), the artificial intelligence component identifies pertinent data points and prioritizes sug- gestions according to certainty levels, applicability, and pattern frequency. Every suggestion is validated with excerpts from the source AYUSH manuscripts.

Technology Stack

The proposed system uses a modular combination of open-source AI frameworks and web technologies. Python serves as the main development language, integrating Tesseract OCR

[2]through Pytesseract for extracting text from scanned San- skrit and Hindi manuscripts. Text preprocessing and linguistic normalization are done using NLTK and IndicNLP, supported by Regex for cleaning and formatting. The identification of entities and comprehension of context employ a combination of SpaCy, Conditional Random Fields, and IndicBERT tech- nologies, guaranteeing precise recognition of medicinal plants, ailments, and therapeutic preparations. The identified elements and their associations are managed within the Neo4j graph database[5], facilitating meaning-based searches and graph- ical representations.

For intelligent analysis and suggestion generation, the platform incorporates BERT, IndicBERT, and traditional machine learning algorithms from Scikit-learn such as Random Forest classifiers. Server-side functionality relies on Flask, while client-side interaction is powered by React.js, offering a user-friendly multilingual experience. Information persistence is achieved through SQLite and Neo4j databases.

In summary, this technological infrastructure guarantees seam- less operation from manuscript digitization through intelligent information extraction to recommendation delivery

IV. COMPARISON BETWEEN EXISTING AND PROPOSED SYSTEM ARCHITECTURE

The limitations of existing Ayurvedic digitization systems arise primarily from their dependence on general OCR[2]tools, small and manually curated datasets, and basic linguistic processing methods that fail to capture the complexity of Indic scripts and classical medical terminology. These systems typically lack the capability to handle degraded manuscripts, compound Sanskrit structures, and contextual variations found across Ayurvedic texts.In Existing System ,There was not proper symptom representation and the frequent mistransla- tion of domain-specific terminology result in fragmented and unreliable data. The absence of semantic relationship model- ing among key Ayurvedic elements—including herbs, symp- toms, formulations, and therapeutic interventions—leaves the extracted information fragmented and inadequate for com- plex analytical tasks or research support. Furthermore, the prevailing manual approach to digitization poses significant scalability challenges, consequently limiting the accessibility of systematically structured Ayurvedic knowledge within aca- demic and clinical domains.

TABLE I Comparative Analysis of Existing and Proposed Architectures

Aspect	**Existing Systems**	**Proposed System (This Work)**
OCR Capability	Works mainly for modern printed text. It has poor accuracy for Sanskrit and Hindi manuscripts.	OCR[2] optimized for San- skrit and Hindi AYUSH manuscripts, with better accuracy
Dataset Size & Quality	Small datasets collected by hand, with few sources.	Large dataset created from multiple sources using OCR and NLP from classical Ayurvedic texts.
Translation Qual- ity	Seq-to-seq models have trouble with long and complicated Ayurvedic sentences and terms.	Fine-tuned Transformer models designed for Ayurvedic terminology and context
Semantic Under- standing	No connections or con- textual mapping between herbs, symptoms, and treat- ments.	Uses NLP and a knowledge graph (Neo4j) for under- standing context and mod- eling relationships.
Standardization of Symptoms	Descriptions of symptoms are inconsistent and don't match classical references.	Creates a standardized symptom database aligned with classical Ayurvedic literature..
Domain Terminology	Ayurvedic terms are of- ten overlooked or wrongly translated.	Creates custom vocabulary and domain-specific em- beddings to keep the mean- ing intact.
Scalability & Au- tomation	lacks clinical deployment.	Mostly manual digitiza- tion, limiting scalability.

The proposed system introduces a comprehensive and domain-optimized framework designed to overcome these challenges through an integrated OCR–NLP–Knowledge Graph pipeline. By employing script-specific OCR mod- els, a large multi-source dataset, and refined preprocessing techniques, the system significantly improves text recogni- tion accuracy for Sanskrit and Hindi manuscripts. The use of Transformer-based language models enables high- quality translation and contextual understanding of long and complex Ayurvedic sentences, while the construction of a Neo4j knowl- edge graph captures semantic relationships among symptoms, herbs, formulations, and therapeutic procedures. The use of curated symptom databases alongside domain-specific em- beddings serves a dual purpose: maintaining terminological consistency and protecting the cultural and medicinal integrity of classical literature.The automated system greatly enhances scalability, enabling mass digitization, systematic storage, and intelligent querying of Ayurvedic literature[5] .This provides a robust foundation for research, clinical decision support, and future machine learning applications in the AYUSH domain.

V. METHODOLOGY

The proposed AI-powered AYUSH Knowledge Extraction and Recommendation System follows a structured, multi- stage pipeline to transform traditional manuscripts into ma- chinereadable, semantically enriched data. The methodology includes six sequential, but connected, stages. Each stage helps with clear knowledge representation and smart inference.

Data Collection and OCR Conversion: The initial phase involves collecting ancient and modern AYUSH texts, typi- cally available as scanned PDFs or image files. These might consist of classical Sanskrit works like the Charaka Samhita and Sushruta Samhita, or modern AYUSH publications in Hindi. The Tesseract OCR[2] engine, specifically optimized for Devanagari script, transforms scanned pages into Unicode text[6]. Prior to the OCR process, OpenCV techniques such as converting to grayscale, binarization, and correcting skew are utilized to improve clarity and enhance recognition precision. The output from OCR includes confidence scores that help evaluate reliability, allowing uncertain sections of text to be flagged for manual examination or retraining This step effectively digitizes centuries-old handwritten or printed texts into machine-readable format[3].

Text Preprocessing and Cleaning: Text Preprocessing and Cleaning: Once texts are obtained, it generally contains artifacts, such as space error, mixed characters, OCR induced distortions etc. The preprocessing module is concerned with converting this raw text into linguistically normalised data, using the Indic NLP and NLTK libraries[1].

Critical operations are:

1. Unicode normalization (NFC/NFKC) to normalize the encoding of scripts in use

2. tokenization which breaks text into semantically meaningful tokens, e.g., words and expressions greater than a word

3. removal of stopwords that could help to simplify grammar but provide redundancy in our context

4. lemmatization/stemming so as to reduce words to their most basic form, widening the keyword list considered for search at input time; and Sandhi splitting for Sanskrit compound words[6]

All of this is done while keeping some structural information from the original text, allowing concrete semantics analysis.

NLP and Entity Recognition: During this stage, Natural Language Processing (NLP) methods are utilized to extract significant entities and their connections. This process includes recognizing herbs, ailments, symptoms, and treatments from the text[4].

1. Rule-based patterns: SpaCy and Regex are used for identifying domain-specific terms.

2. CRF models: Conditional Random Fields label sequences for entity classification.

3. Transformer-based models: IndicBERT fine-tuning cap- tures contextual nuances unique to Sanskrit and Hindi.

Relations such as *treats*, *contains*, or *usedwith* are extracted using dependency parsing and semantic pattern matching.

For example, from the sentence: *"Tulsi is beneficial for cough"*

→ (Tulsi) —[treats]→ (Cough).

The output is a structured set of entities and their semantic relations, forming the foundation of the Knowledge Base.

Knowledge Graph Construction: LeapNLP knowledge graph is built on a Neo4j platform where entities and relations extracted from the documents are maintained as a center source for structured information[6]. Each molecule is a node and each relationship between molecules becomes the directed edge, with a semantically defined labeltreats, enhances, or contraindicatedfor.

Example of graph structure:

(Ashwagandha) |[reduces]→ (Stress) (Tulsi) |[treats]→ (Cough) (Ginger) |[enhances]→ (Digestion)

Each of these relations is labeled with the metadata such as the source document, Sentence ID and confidence score. The Knowledge Graph enables Cypher queries like:

1. "Return all herbs that treat fever, but do nottreat cough."

2. "Show me all drugs that include Ashwagandhain their formulations."

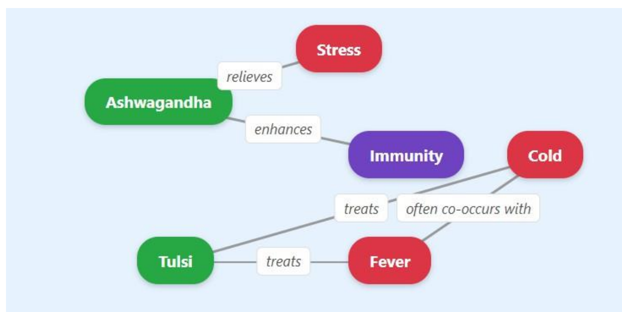

Fig. 2. Knowledge Graph

5. AI-Based Reasoning and Recommendation: Once the Knowledge Graph is established, machine learning and deep learning models are employed to infer new insights and respond to user queries[9].

Classical models such as Random Forest and Logistic Regression identify statistically significant herb–disease rela- tionships, while **Transformer-based models** like

BERT and IndicBERT[7][8] interpret natural language questions such as "What helps with fever?"

These models analyze graph connections and textual embed- dings to produce ranked recommendations based on frequency, context, and semantic similarity. Every suggestion is justified connected to the source text for validation, guaranteeing both scientific responsibility and clarity, which are essential in healthcare contexts.

6. User Interface and Visualization: The final stage focuses on system accessibility. The Flask-based backend communi- cates with the AI and graph layers, while a React.js frontend provides an user friendly interface. Users have the option to enter queries by typing or by speaking in either English or Hindi, with results shown as both text-based lists and interactive graphs. Visualization tools like Neo4j Browser and PyVis allow users to investigate the relationships between entities (for instance, herbs associated with various diseases). Each node and connection in the visualization is clickable, providing access to relevant text excerpts, which enhances transparency and builds user trust.

VI. DATASET DESCRIPTION

A. *Dataset Source*

The dataset used in this study consists of a collection of Ayurvedic and Unani medical manuscripts collected from public digital libraries, academic repositories, and scanned printed texts. The dataset includes:

1. Classical Ayurvedic treatises in Sanskrit, such as Ash- tanga Hrudayam

2. Articles on modern and semi-classical Unani medicine (e.g., treatises on medicinal plants such as Aloe Vera)

• Digitized Hindi AYUSH publications used for further verification

These documents were selected to reflect the diversity of writing styles, historical periods, and medical traditions within the AYUSH system.

B. Representative Examples

1. Ashtanga Hrudayam (Sanskrit, Devanagari Script): A classical Ayurvedic text explaining foundational princi- ples, diagnostic methods, and therapeutic formulations.

2. Aloe Vera – Unani Treatise (Perso-Arabic/Urdu Script): A Unani medical manuscript describing the pharmacolog- ical properties and medicinal applications of Aloe Vera.

3. Yunani-Adwia-Mufrida (Unani, Urdu/Perso-Arabic Script): A classical Unani text focusing on single medicinal ingredients and their therapeutic indications.

4. Makhzan ul Advia (Unani, Urdu/Perso-Arabic Script): A well-known Unani pharmacopoeia documenting medic- inal substances, their properties, and clinical uses.

C. *Dataset Preprocessing Issues*

Several preprocessing challenges were identified in the dataset:

• Variability of scripts between Devanagari and Perso-Arabic

1. Poor scan quality, including ink smears, skewed pages, and background noise

• Complex Sanskrit words and ligatures requiring Sandhi resolution

1. OCR noise caused by inconsistent spacing and font styles

• Domain-specific terminology absent from standard NLP dictionaries

To address these challenges, a script-aware OCR pipeline, adaptive image preprocessing, and domain-specific normaliza- tion rules were applied prior to downstream NLP processing.

D. *Data Collection Challenges*

The data collection process faced several challenges, includ- ing:

Degraded pages with faded ink and physical creases

Highly complex Sanskrit compounds and ligatures

1. OCR errors arising from inconsistent spacing and format- ting

VII. MATHEMATICAL MODEL

The proposed system can be mathematically expressed as a series of transformations that convert scanned AYUSH manuscripts into structured and intelligent outputs.

Let:

$S = \{I, P, O, f\}$

where:

I: Input set (scanned AYUSH texts)

P : Set of processing functions

1. O: Output set (Knowledge Graph and AI-based recommendations)

2. f : Overall transformation function , relationship can be expressed as follow

$$O = fAI\ fKG\ fNLP\ fpre\ fOCR(I)$$

Here:

1. fOCR: Extracts text from scanned images using Tesseract OCR

• fpre: Cleans and normalizes text (IndicNLP, NLTK)

1. fNLP : Identifies entities and relationships using hybrid NLP models

• fKG: Constructs the Knowledge Graph using Neo4j

• fAI : Generates AI-based recommendations using trained models

The system's learning objective is to minimize the prediction error between the actual and predicted relationship scores, defined as:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i and $\hat{y}_i$ represent the actual and predicted relationship scores, respectively.

The system is considered successful when:

Entity Extraction Accuracy ≥ 90% and Graph Completeness ≥ 85%

Thus, the mathematical model represents a well-defined mapping from raw scanned data to structured, explainable, and AI-driven AYUSH knowledge.

VIII. ALGORITHM DESIGN

The overall design of the proposed AYUSH Knowledge Extraction and Recommendation Framework follows a systematic sequence of algorithmic steps. The steps consists of gradual transformation of unstructured text data into structured, machine-interpretable knowledge suitable for intelligent reasoning.

Algorithm 1: AYUSH Knowledge Extraction and Recommendation Process:

1) **Start**
2) **Collect Dataset:** Collect dataset D consisting of scanned AYUSH manuscripts and digital documents.
3) **For each document** $d_i \in D$:
   a) Apply OCR (Tesseract) to extract textual content
   → T

   b) Perform image preprocessing using OpenCV (bi-
   narization, de-noising, skew correction)
4) **Text Preprocessing:**
   a) Tokenize sentences and words

   using NLTK/IndicNLP
   b) Remove stopwords and special symbols
   c) Perform stemming and lemmatization
   d) Normalize text to Devanagari Unicode format
5) **Language Identification:** Detect and separate Sanskrit, Hindi, and English text segments for context-specific processing.
6) **Named Entity Recognition (NER):** Apply IndicBERT and SpaCy models to identify key entities:
   - Herbs (e.g., Tulsi, Ashwagandha)
   - Diseases (e.g., Jwara, Cold)
   - Treatments / Formulations (e.g., Chyawanprash)
7) **Semantic Relationship Extraction:** Use dependency parsing to identify and extract relations between entities. Example: *"Tulsi treats cough"* → (Tulsi —[treats]→ Cough)
8) **Knowledge Graph Construction:** Construct Knowl- edge Graph G using Neo4j, where:
   - Nodes represent entities (V)
   - Edges represent relationships (E)
   - $G = (Nodes, Edges)$
9) **AI Model Training:** Train machine learning and deep learning models using extracted graph data:
   - Random Forest and Logistic Regression for statisti- cal learning
   - BERT for semantic relationship understanding
10) **Query Processing:** When a user query Q is

received:
   a) Analyze query intent using BERT-based language understanding
   b) Retrieve relevant nodes and edges from Knowledge Graph *G*
   c) Rank results based on confidence scores and con- textual similarity
11) **Result Visualization:** Display the final results *R* through a web interface (Flask + React.js):
   - Show text-based output and ranked recommenda- tions
   - Provide interactive graph visualization using PyVis or Neo4j Browser
12) **End**

Unicode Format

IX. CONCLUSION AND FUTURE WORK

The advancement of artificial intelligence has introduced innovative methods for the preservation and interpretation of historical medical knowledge. Techniques such as Optical Character Recognition (OCR), Natural Language Processing (NLP), Knowledge Graphs, and machine learning facilitate the digitization of historical manuscripts, transforming them into structured and accessible information. This integration bridges ancient wisdom with contemporary computing, thereby en- hancing accessibility and research opportunities within tradi- tional healthcare. Future initiatives may aim to enhance the ac- curacy of OCR, develop larger annotated datasets, and employ Graph Neural Networks (GNNs) for more profound semantic comprehension. Furthermore, expanding support for multiple languages and mobile platforms, in addition to integrating traditional and biomedical databases, can significantly bolster data-driven, comprehensive research in the field of healthcare.

REFERENCES

1. M. Ranade, “Artificial intelligence in Ayurveda: Current concepts and prospects,” Journal of Indian System of Medicine, vol. 12, no. 1, pp. 53–59, Jan.–Mar. 2024, doi: 10.4103/jism.jism 60 23.
2. H. T. Jayaraj, K. Sourav, and B. Resmi, “Artificial intelligence powered OCR models for digitizing Ayurveda manuscripts,” Journal of Ayurveda and Holistic Medicine (JAHM), vol. 12, no. 9, 2024.
3. R. Acharya, “Integrating artificial intelligence into Ayurveda: Path- ways, potentials, and challenges,” Journal of Drug Research in Ayurvedic Sciences, vol. 10, no. 3, pp. 177–180, May–Jun. 2025, doi: 10.4103/jdras.jdras 176 25.
4. “AI Driven Ayurvedic Chatbot: Revolutionizing Formulation Selection and Knowledge Accessibility in Traditional Medicine Through the Integration of LENET and Stacked Autoencoders,” 2024. bibitemb5 C. K. Katiyar and S. K. Dubey, ”Opportunities and challenges for Ayurvedic industry,” International Journal of Ayurveda Research, vol. 4, no. 3, pp. 123–131, Jul.–Sep. 2023, doi: 10.4103/ijar.ijar 114 23.
5. R. Naik, G. Patheja, and A. R. Unnikrishnan, ”Standardization of Ayurveda medical devices: A new initiative by Bureau of Indian Stan- dards,” International Journal of Ayurveda Research, vol. 6, no. 1, pp. 46–51, Jan.–Mar. 2025, doi: 10.4103/ijar.ijar 258 24.
6. S. Rastogi and P. Tiwari, ”Implementation of Machine Learning Tech- niques in Ayurveda for Diagnosis and Treatment,” in Proc. Int. Conf. Recent Trends Comput. Sci. Electron., Singapore: Springer, 2018, pp. 131–137.
7. A. Shukla et al., ”Machine Learning Based Ayurvedic Treatment Rec- ommendation System,” in Proc. Int. Conf. Comput. Intell. Data Sci., Singapore: Springer, 2020, pp. 113–121.
8. M. M. Ahsan, S. A. Luna, and Z. Siddique, ”Machine-Learning-Based Disease Diagnosis: A Comprehensive Review,” Healthcare (Basel), vol. 10, no. 3, p. 541, Mar. 2022, doi: 10.3390/healthcare10030541.
9. J. Al-Salman, J. Mustafina, and G. Shahoodh, ”A Systematic Review of Artificial Neural Networks in Medical Science and Applications,” in Proc. 13th Int. Conf. Developments Syst. Eng. (DeSE), Liverpool, U.K., 2020, pp. 279–282, doi: 10.1109/DeSE51703.2020.9450245.
10. C. Rathor, V. S. Mandloi, I. Sachan, and V. Sahu, ”Artificial Intelligence in Ayurveda: A Simple

Overview," J. Ayu Int. Med. Sci., vol. 9, no. 11, pp. 271–275, 2024.