Integrated Diagnostic System with Ensemble ML, Computer Vision, and LLM Agent

February 8, 2025

Abstract

This document presents a detailed technical specification for an integrated diagnostic system that processes laboratory data and patient-supplied images to produce probabilistic illness predictions. It employs two primary pipelines: an ensemble machine learning (ML) model for lab reports and a computer vision (CV) pipeline for image-based analysis. Outputs from these pipelines are fused and fed into a large language model (LLM), which engages in a structured conversation, asks targeted follow-up questions, and generates a final diagnosis. A robust feedback loop compares the LLM's diagnostic output to the ground-truth prescription from a clinician, logs discrepancies, and updates the system components as necessary.

Contents

1 Introduction

The goal of this diagnostic platform is to streamline medical decision-making by automatically:

i) Extracting probability distributions of potential illnesses based on laboratory reports through an ensemble ML model.

ii) Analyzing patient-provided images (e.g., potential fungal infections) via a dedicated CV pipeline.

iii) Integrating both outputs into a trained LLM that mimics a physician's approach, asking relevant questions and refining diagnostic hypotheses.

iv) Validating and improving performance through a feedback loop that incorporates real clinician prescriptions.

This document covers the end-to-end pipeline, data processing strategies, feedback loop architecture, and technical considerations for deployment and continuous improvement.

2 System Architecture

2.1 High-Level Overview

Figure 1 conceptually illustrates the major components of the system:

a) **Lab-Based Ensemble ML:** Processes cleaned lab data to produce a probabilistic illness distribution.

b) **Computer Vision Pipeline (CV):** Utilizes a vision-language model (VLM) for patient image analysis.

c) **LLM Integration:** Aggregates information from both pipelines to conduct an interactive diagnostic session with the patient and output a final report.

d) **Feedback Loop:** Compares the LLM's final diagnosis to the clinician's prescription and triggers updates to all models when discrepancies exceed a threshold.

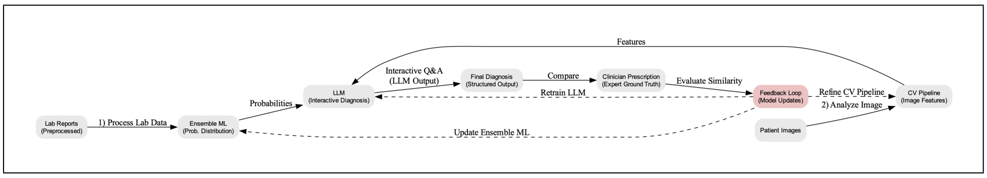

Figure 1: Conceptual System Architecture

3 Data Processing and Preparation

3.1 Laboratory Reports

- **Data Ingestion:** Parse patient lab results (e.g., CBC, metabolic panel). Records with >50% missing values are discarded to maintain data reliability.

- **Missing Data Handling:** Mean imputation is used for tests with ≤50% missingness; potential outliers are capped or removed based on clinical thresholds.

- **Feature Vectors:** Each patient's lab data is transformed into a numerical feature vector suitable for the ensemble ML model.
- **Label Definition:** Instead of using specific ICD codes, illness categories are grouped into broader umbrella terms (e.g., "Respiratory," "Metabolic," "Infectious"), which serve as the classification targets.

3.2 Image Data and CV Pipeline

- **Input Modalities:** Patient-supplied images indicating possible infections or visible symptoms.
- **Computer Vision Model:**
  i) Pre-trained on diverse medical and non-medical image sets for robust feature extraction.
  ii) Fine-tuned on condition-specific data (e.g., fungal infection datasets) if available.
  iii) Outputs a natural language output indicating the extracted relevant information from the image, which can be passed onto the LLM at later stages.

4 Modeling Components

4.1 Ensemble ML for Lab Reports

- **Architecture:** Weighted ensemble of:
  a) **LightGBM:** Fast gradient boosting, effective for structured data.
  b) **XGBoost:** Handles diverse features robustly.
  c) **Deep Neural Network (DNN):** Captures non-linear relationships.
- **Training Procedure:**
  i) **Cross-Validation:** Stratifed k-fold cross-validation to handle class imbalance.
  ii) **Ensemble Fusion:** Weighted averaging or meta-learner (stacked ensemble) to combine model outputs.
- **Output:** Probability distribution over broad disease categories, plus top-k (commonly $k = 3$) predictions with confidence scores.

4.2 LLM Integration for Interactive Diagnosis

- **Input Fusion:**
  a) Probabilistic output from the ensemble ML model (lab-based).
  b) Detected features from the CV pipeline.
- **Dialogue Management:**
  i) The LLM (fine-tuned on medical dialogue data) uses the combined context to ask targeted follow-up questions.
  ii) Ensures relevant topics (e.g., timeline, medication history) are explored.
- **Final Diagnosis Generation:**
  i) Outputs a structured diagnostic summary: timeline, symptoms, probable illness, recommended interventions, and emergency status.
  ii) This summary is stored for subsequent evaluation and feedback.

5 Inference Workflow

1. **Patient Submission:** User uploads lab report data and an image (if available).
2. **ML + CV Processing:**
   a) Ensemble ML model infers illness probabilities from lab data.
   b) CV pipeline analyzes images, generating a natural language output.
3. **LLM Interaction:**
   a) Aggregated results are fed into the LLM.
   b) LLM queries the user for additional clarifications, if needed.
   c) LLM produces a final diagnostic statement or set of recommendations.
4. **Outcome:** LLM's structured output is recorded, along with any confidence measures.

6 Feedback Loop and Continuous Improvement

6.1 Clinician Verification and Similarity Scoring

- **Expert Prescription:** A qualified clinician reviews the patient's data and provides a final prescription or official diagnosis.
- **Similarity Metric:** Compare the LLM's final diagnostic output to the clinician's prescription using a textual similarity or domain-specific metric (e.g., cosine similarity on embeddings, or a custom medical concept overlap score).
- **Thresholding:** If similarity < 0.8 (configurable), the case is flagged for review.

6.2 Flagged Case Analysis

- **Discrepancy Logging:** Store relevant information, including:
  i) ML ensemble outputs (top-k predictions, confidence scores).
  ii) CV pipeline results.
  iii) LLM's final diagnostic text and the clinician's prescription.
- **Root-Cause Examination:** Domain experts or medical data scientists analyze which features or pipeline steps led to the misalignment.

6.3 Retraining and Model Updates

a) **Data Augmentation:** Incorporate flagged cases into the training dataset to address systematic biases or coverage gaps.

b) **Incremental Training:** Apply partial (incremental) retraining for:
   i) Ensemble ML model: LightGBM/XGBoost can be updated with additional training data.
   ii) DNN and CV modules: Fine-tune if repeated misclassifications appear in image-based analysis.
   iii) LLM: Fine-tune the dialogue/response generation portion on newly flagged medical dialogues.

c) **Versioning and Monitoring:** Maintain model version control, track performance metrics before and after retraining, and revert to earlier stable versions if performance degrades.

7 Implementation Considerations

7.1 Data Privacy and Compliance

- Ensure all patient data is de-identified where possible.
- Comply with relevant local data protection laws through secure storage, encrypted transfer, and restricted access.

7.2 Scalability

- **Microservices Architecture:** Deploy each pipeline component (ML ensemble, CV module, LLM) as a separate service, allowing horizontal scaling.
- **Caching and Load Balancing:** Cache intermediate results (e.g., image embeddings) to reduce redundant computation; distribute load across nodes for concurrency.

7.3 Monitoring and Logging

- Use real-time monitoring (e.g., Grafana, MLflow, or Prometheus) to track inference latency, resource usage, and error rates.
- Log each inference request with pipeline outputs, similarity scores, and final decisions for traceability.

8 Conclusion

This technical specification outlines a robust, multi-pipeline diagnostic system that fuses lab-based ensemble ML outputs and CV-generated image features within an LLM-driven interactive diagnostic workflow. By integrating a feedback loop where clinician prescriptions are compared against system outputs, the platform ensures continuous refinement. Future improvements include domain-adaptive training for the LLM, more sophisticated imputation strategies for sparse lab data, and expanded image recognition capabilities tailored to region-specific pathologies.