International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 4, July – August 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

A Multi-Agent Architecture for AI-Based Early Screening, Referral, and Prediction of Retinal Diseases

Coordination, Communication, and Memory in an Agentic Ophthalmic Screening System

Jahnavi Somaraju[1], Poolasetti Vamshi[2]
Assistant Professor [1], Final Year UG Student[2]
Department of Artificial Intelligence and Data Science, Aditya College of Engineering, Madanapalle
Email: pulasettyvamshi@gmail.com

Abstract—

Retinal diseases such as diabetic retinopathy (DR), glaucoma, and age-related macular degeneration (AMD) are leading causes of preventable blindness worldwide, yet population-scale screening remains constrained by the limited availability of trained ophthalmologists, particularly in primary-care and rural settings. Single-agent deep learning classifiers, while effective at image-level grading, conflate image-quality assessment, lesion detection, severity classification, referral triage, and patient communication inside one undifferentiated pipeline, making them brittle to poor-quality captures and opaque to the clinicians who must act on their output. This paper proposes a multi-agent architecture — the Retinal Screening and Referral Agent (RSRA) — that decomposes early screening across six cooperating agents: an Orchestrator, an Image Quality and Preprocessing Agent, a Lesion Detection Agent, a Disease Classification and Risk Prediction Agent, a deterministic Referral Decision Agent, and an LLM Reasoning and Report-Generation Agent supported by a three-tier memory system. Agents communicate through the Model Context Protocol (MCP), a standardized client-server tool-calling layer, rather than bespoke point-to-point integrations. We present the complete system architecture, agent interaction sequence, clinical decision workflow, communication protocol, memory architecture, data flow, and deployment architecture, together with the coordination algorithms governing agent selection, scheduling, conflict resolution, and consensus. A worked case study demonstrates the pipeline end-to-end on a simulated moderate-DR fundus image. Full benchmark-scale quantitative evaluation is scoped as an immediate next phase and is described here as a concrete, reproducible protocol rather than presented as completed results.

Keywords—*Multi-Agent Systems, Agentic AI, Diabetic Retinopathy, Retinal Disease Screening, Clinical Decision Support, Large Language Models, Model Context Protocol, Retrieval-Augmented Generation, Vector Memory*

I. INTRODUCTION

A. Motivation

An estimated 2.2 billion people live with vision impairment globally, and at least one billion of these cases could have been prevented or are yet to be addressed, including a substantial share attributable to diabetic retinopathy and glaucoma [9]. Diabetic retinopathy alone threatens vision in roughly one in three people with diabetes, yet routine annual eye screening — the single most effective preventive measure — remains unavailable to a large share of the diabetic population outside urban tertiary centres. Deep learning fundus-image classifiers have repeatedly matched ophthalmologist-level grading accuracy on benchmark datasets [1], [2], but translating a single classifier into a deployable screening service exposes a gap between 'can grade an image' and 'can run a trustworthy, auditable screening programme.'

B. Problem Statement

A single monolithic model tasked with screening a fundus image must implicitly handle image-quality gating, lesion-level detection, severity grading, referral-urgency assignment, and patient-facing explanation inside one forward pass or one prompt. This conflation obscures which stage produced an error when a screening result is wrong, discourages clinician trust, and provides no natural point at which a deterministic, auditable rule — such as a clinical referral guideline — can be enforced independently of the model's output distribution.

C. Why Multi-Agent Systems Are Needed

Separating image-quality gating and lesion detection, which must be reliable and explainable, from referral-letter generation and patient communication, which benefit from language flexibility, keeps the highest-stakes clinical decision — does this patient need urgent referral — governed by deterministic, guideline-encoded rules, while confining the large language model's role to explanation, contextualization, and drafting. This mirrors the broader industry shift toward coordinated multi-agent orchestration over single-agent designs in high-stakes domains [5], and is the central architectural argument of this paper.

D. Research Objectives

- Design a multi-agent screening pipeline that isolates deterministic lesion detection and referral triage from LLM-based reasoning and reporting.
- Define a standardized inter-agent and agent-to-tool communication protocol suitable for clinic and tele-ophthalmology deployment.
- Design a three-tier memory architecture that lets the reasoning agent draw on a patient's screening history and prior similar cases without retraining.
- Propose a reproducible, tagged evaluation protocol for benchmarking agentic retinal-screening reliability against clinical ground truth.

E. Contributions of This Paper

- Architecture: A six-agent pipeline (Orchestrator, Image Quality Agent, Lesion Detection Agent, Disease Classification and Risk Prediction Agent, Referral Decision Agent, LLM Reasoning Agent) with an explicit division of labor between deterministic detection/triage and LLM-based reasoning.
- Communication layer: An MCP-based agent-to-tool protocol replacing bespoke API integrations, detailed in Section VI.
- Memory system: A three-tier (short-term, long-term, vector) memory architecture supporting retrieval-augmented, longitudinal patient reasoning.
- Evaluation protocol: A tagged, reusable case-study template and benchmark-based evaluation methodology suited to reliability reporting for agentic clinical screening tools.

II. Literature Review

A. Existing Multi-Agent Architectures

The multi-agent LLM ecosystem has consolidated around a small set of orchestration philosophies — LangGraph's explicit stateful graph, CrewAI's role-based team abstraction, and AutoGen's conversational multi-turn pattern — each of which answers the coordination question differently and ramifies through state management, token cost, and failure recovery [5]. Healthcare-specific multi-agent proposals remain comparatively scarce, and few published systems address the ophthalmic screening pipeline end-to-end rather than the image-classification step in isolation.

B. Single-Agent vs. Multi-Agent Approaches

Single-model DR classifiers such as Gulshan et al.'s Inception-v3-based grader [1] and the multiethnic validation study of Ting et al. [2] establish that image-level grading can reach specialist-level accuracy, and Abramoff et al.'s IDx-DR system demonstrated that an autonomous single-pipeline classifier could be validated for primary-care deployment without a human grader in the loop [3]. De Fauw et al. extended this further with a two-stage deep learning system that first segments a 3-D OCT scan into a tissue map and then classifies referral urgency from that map — an early example of decomposing detection from triage [4]. Gargeya and Leng showed that a single CNN combined with a data-driven risk score could flag referable disease with high sensitivity [11]. None of these systems, however, separate an LLM-mediated explanation layer, a longitudinal per-patient memory, or a standardized tool-communication protocol from the detection core.

C. Limitations of Current Systems

- Systems that rely on a single classifier for both grading and referral decisioning inherit that classifier's failure modes directly into the clinical action taken, with no independent, auditable checkpoint.
- Point-to-point API integrations between an image-grading model, an EHR, and a communication channel do not generalize across clinics and complicate maintenance as a deployment scales across sites.
- Few systems report longitudinal, per-patient reasoning: a repeat screening rarely benefits from the patient's own screening history, even though disease trajectory (worsening vs. stable) is itself clinically informative.

D. Research Gap

No system reviewed combines (i) a hard separation between deterministic lesion detection/referral triage and LLM-based reasoning, (ii) a standardized, protocol-based communication layer between agents and tools, and (iii) a three-tier, patient-scoped memory architecture supporting longitudinal, retrieval-augmented explanation. This paper addresses that combined gap.

TABLE I

Comparison with Closely Related Retinal-Screening Systems

System	Coordination Style	Key Limitation Noted
IDx-DR [3]	Single-pipeline autonomous classifier	No LLM reasoning/explanation layer; fixed rule-based referral output only
Moorfields / DeepMind system [4]	Two-stage segmentation + classification	Decomposes detection from triage, but not agentic or tool-mediated
Gargeya & Leng [11]	Single CNN + data-driven risk score	No referral-urgency tiering or patient-facing explanation layer

Proposed System (RSRA)	6-agent pipeline; deterministic detection + MCP-mediated LLM reasoning	Deterministic core limits detection to trained lesion classes / atypical presentations

III. Background

A. Multi-Agent Systems

A multi-agent system (MAS) decomposes a task across multiple autonomous, communicating agents rather than one monolithic process. Coordination primitives — how agents discover each other, share state, and decide who acts next — are the core engineering problem in MAS design, distinct from single-agent prompt or tool design [5].

B. Large Language Models as Reasoning Agents

In this architecture, the LLM is used strictly as a reasoning and communication component: given structured lesion findings, a severity grade, and retrieved context, it produces explanation, a patient-facing summary, and referral-letter text. It is not used as the primary lesion detector or the referral-urgency decision-maker, which keeps the highest-stakes clinical decisions deterministic and testable.

C. Agent Roles and Responsibilities

Six roles are defined in this system: Orchestrator (task decomposition and routing), Image Quality and Preprocessing Agent, Lesion Detection Agent, and Disease Classification and Risk Prediction Agent (deterministic/model-based specialists), Referral Decision Agent (deterministic guideline rules), and the LLM Reasoning Agent (explanation, prioritization, and report drafting). Each has a single, narrowly scoped responsibility, consistent with the role-based design philosophy used in frameworks such as CrewAI [5].

TABLE II

Agent Roles and Responsibilities Summary

Agent	Type	Primary Responsibility
Orchestrator Agent	Coordination	Task decomposition, routing, session state management
Image Quality & Preprocessing Agent	Deterministic	Blur/glare/field-of-view gating and image normalization
Lesion Detection Agent	Deterministic (CNN)	Detects microaneurysms, haemorrhages, exudates, cup-disc ratio
Disease Classification & Risk Prediction Agent	Deterministic (DL model)	ICDR severity grading and longitudinal risk scoring
Referral Decision Agent	Deterministic (rules)	Maps severity/risk to referral urgency tier
LLM Reasoning & Report-Generation Agent	LLM + RAG	Explanation, patient report, referral letter drafting

D. Communication Protocols

Two coordination layers are relevant to agentic clinical systems: an orchestration layer (how agents within one system hand off tasks) and a tool-access layer (how any agent calls external tools or data, such as the EHR, the vision-model inference service, or the guideline rules engine). This paper uses the Model Context Protocol (MCP) for the latter — a JSON-RPC-based client-server standard that decouples an agent from any specific tool implementation [6], [7].

E. Memory: Short-Term, Long-Term, and Vector Memory

Short-term memory holds the active fundus image, extracted features, and the in-progress reasoning buffer for the current screening session. Long-term memory persists patient-specific screening history and clinician-confirmed outcomes across visits. Vector memory stores dense embeddings of past fundus findings and clinical-guideline text, retrieved via similarity search and injected into the reasoning agent's context — the retrieval-augmented generation (RAG) pattern introduced by Lewis et al. [8].

F. Planning and Reasoning

The Orchestrator performs lightweight planning — deciding whether an image passes the quality gate and which specialist agents a screening session requires — rather than open-ended multi-step planning. Broader reasoning is delegated to the LLM Reasoning Agent only after deterministic findings and a referral tier already exist, limiting the blast radius of any single planning error.

Figure 1 shows the overall multi-agent pipeline. The system triggers on every completed fundus-image capture and executes as a stateless screening session composed of six cooperating agents.

IV. Proposed Architecture

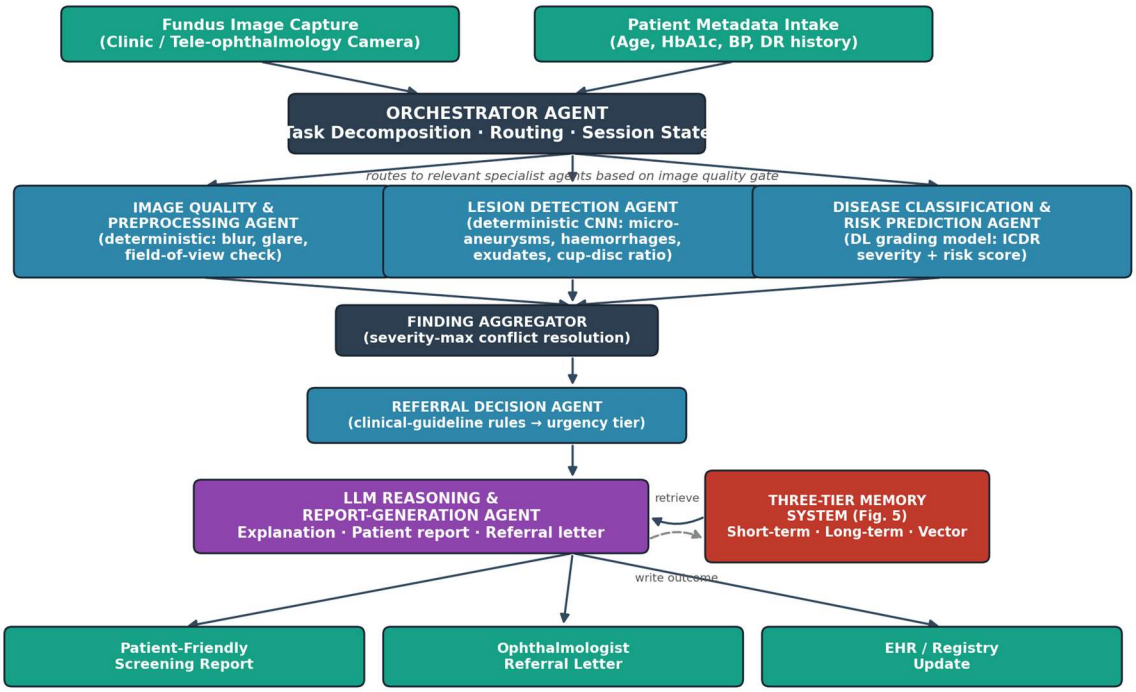

Fig. 1. Multi-Agent Pipeline Architecture — Ingestion, Orchestration, Parallel Specialist Agents, Memory-Augmented Reasoning, and Output Generation

A. Agent Definitions

- Orchestrator Agent — accepts the fundus image and patient metadata, invokes the quality gate, and dispatches only the relevant specialist agents.
- Image Quality & Preprocessing Agent — checks blur, illumination, glare, and field-of-view before any diagnostic agent runs, and normalizes the image (contrast, cropping, resolution) for the downstream models.
- Lesion Detection Agent — a deterministic CNN that localizes microaneurysms, haemorrhages, hard/soft exudates, and computes the cup-to-disc ratio relevant to glaucoma risk.
- Disease Classification & Risk Prediction Agent — grades diabetic retinopathy severity on the International Clinical Diabetic Retinopathy (ICDR) scale and combines image features with patient metadata (HbA1c, blood pressure, diabetes duration) to produce a longitudinal risk score.
- Referral Decision Agent — a deterministic, guideline-encoded rules engine that maps aggregated severity and risk to a referral-urgency tier (routine, urgent, or emergency).
- LLM Reasoning & Report-Generation Agent — consumes aggregated findings, the referral tier, and retrieved memory context to produce a patient-friendly report, a structured referral letter, and a confidence-adjusted explanation.

B. Agent Workflow and Orchestrator Design

The Orchestrator implements task decomposition as a routing table: image-quality outcome and metadata completeness map to a subset of specialist agents, so a rejected image never reaches the lesion-detection or classification stage, and a screening session with no metadata still returns an image-based grading with a lower-confidence risk score rather than blocking entirely. This keeps per-session latency and LLM token cost proportional to what the input actually supports.

C. Routing Mechanism

Routing is rule-based rather than LLM-decided at this stage, so that which detectors run on a given image is deterministic and auditable — an LLM is not in the loop until candidate findings and a referral tier already exist.

D. Tool Usage

All external actions — reading patient metadata, writing results to the EHR, invoking vision-model inference, reading/writing memory — are exposed as MCP tools rather than being hard-coded into agent logic. This means an agent's code does not change if, for example, the lesion-detection model is swapped for a newer checkpoint; only the corresponding MCP server changes.

E. Shared Memory and Knowledge Retrieval (RAG)

The LLM Reasoning Agent is the only agent with read/write access to the three-tier memory system detailed in Section VII. Before generating an explanation, it issues a similarity-search query against vector memory to retrieve prior instances of a similar lesion pattern or a patient's own prior screening, which is used to keep severity language and referral urgency consistent across visits for the same patient.

F. Feedback Loop

When an ophthalmologist confirms or corrects a screening outcome after clinical review, the outcome (confirmed / downgraded / upgraded / false positive) is written back to long-term memory, closing the loop shown in Figure 2 and gradually improving the relevance of future retrieval and the calibration of the risk-prediction agent.

G. Error Handling

Each specialist agent call is wrapped with a timeout and a fallback: if the Disease Classification Agent's inference call fails, the Orchestrator marks that check as "unavailable" in the final report rather than blocking the session, and the LLM Reasoning Agent is explicitly instructed not to fabricate a finding for a check that did not complete. A session that fails the image-quality gate is returned to the capture step with a specific rejection reason rather than silently discarded.

V. Agent Interaction Workflow

Figure 2 traces one complete screening cycle from image submission to ophthalmologist follow-up, including the memory read/write calls and the loop that writes the confirmed outcome back into long-term memory.

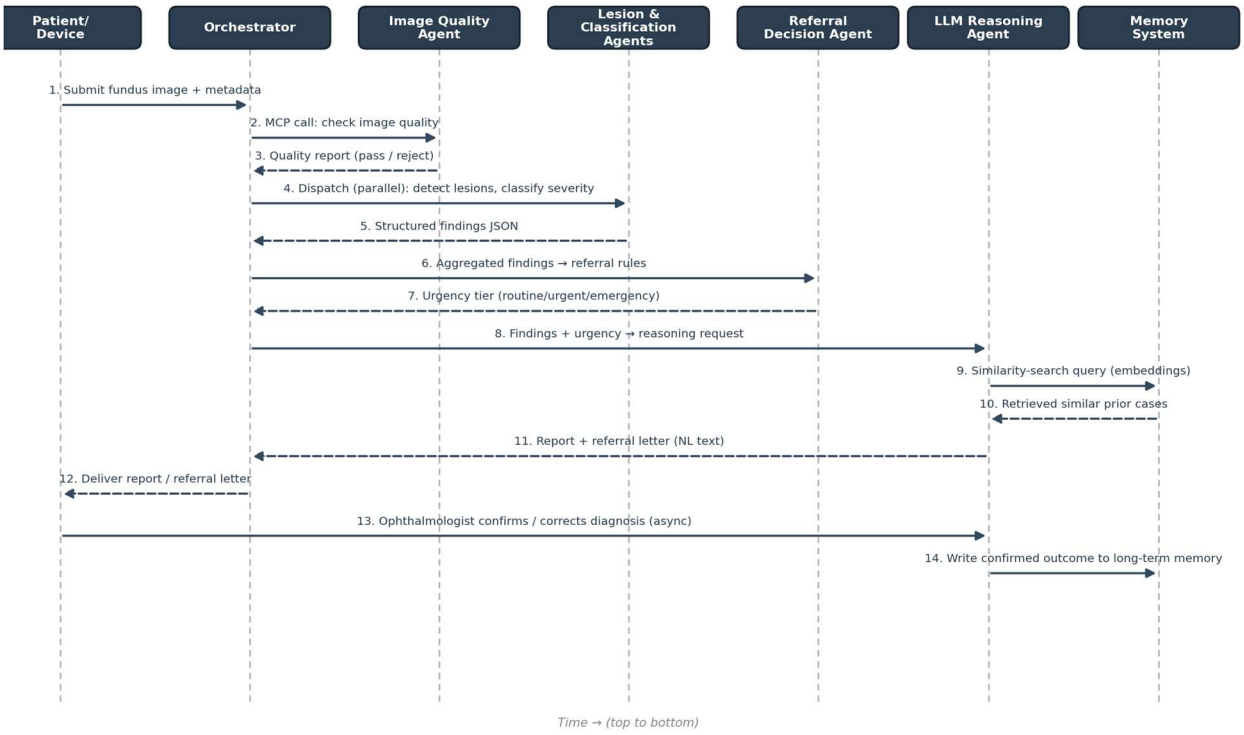

Fig. 2. Agent Interaction Sequence Diagram for One Retinal Screening Cycle

Figure 3 shows the same cycle from a clinical decision-flow perspective, making explicit the branching between the image-quality gate and the three referral-urgency outcomes that the Referral Decision Agent can assign.

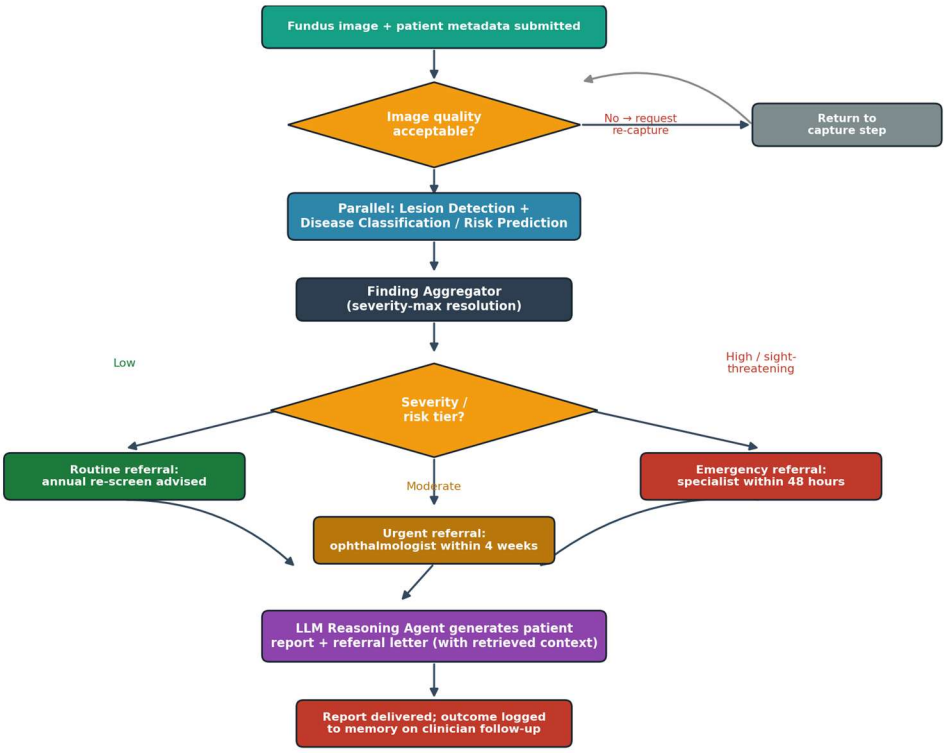

Fig. 3. End-to-End Clinical Workflow with Image-Quality Gate and Referral-Urgency Branching

VI. Agent Communication System

This section describes, in detail, how the agents defined in Section IV exchange messages and tool calls, since this is the layer that most differentiates the proposed design from bespoke, per-tool integrations common in earlier screening systems [3], [4].

A. Protocol Choice: Model Context Protocol (MCP)

All agent-to-tool communication in this system uses MCP, a JSON-RPC 2.0-based client-server protocol originally introduced by Anthropic [7] and since analyzed at the ecosystem level for its security and coordination properties [6]. Each agent acts as an MCP client; each external capability (EHR record store, DICOM/image store, vision-model inference, clinical-guideline rules, memory/RAG, LLM provider) is exposed behind an MCP server. Figure 4 shows the resulting message topology.

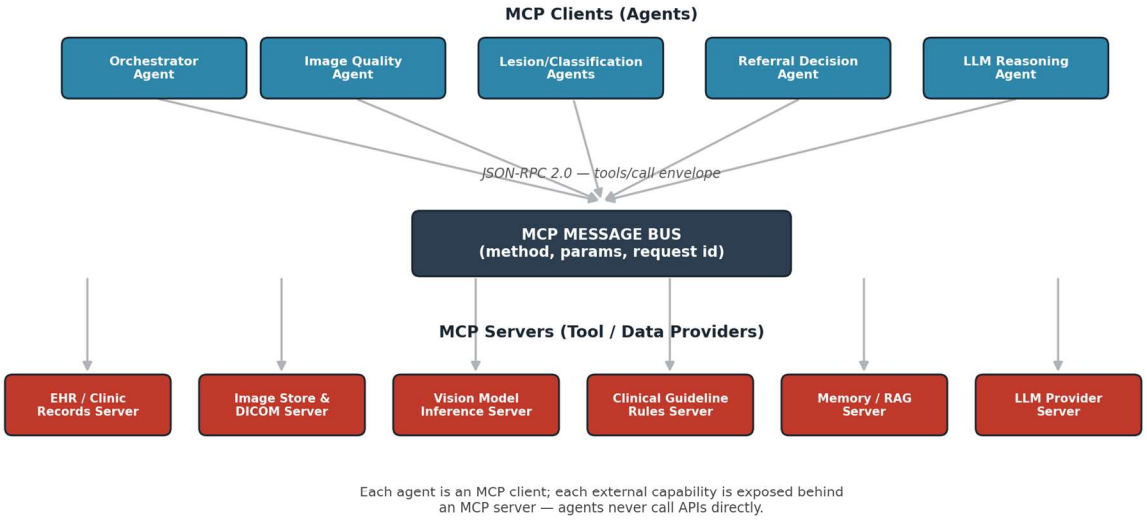

Fig. 4. Agent-to-Tool Communication Protocol Using MCP Client-Server Message Exchange

B. Message Format

Every call follows the JSON-RPC envelope: a method field (tools/call), a params object naming the target tool and its arguments, and a request id used to match asynchronous responses. This uniform envelope is what allows the Orchestrator, specialist agents, and the LLM Reasoning Agent to share one client implementation rather than six.

C. The Agents Used in This System and Their Communication Roles

- Orchestrator Agent — MCP client to the EHR/metadata server and the image-store server only; issues read (patient history, image) and write (session status) calls.
- Image Quality, Lesion Detection, and Disease Classification Agents — stateless MCP clients invoked by the Orchestrator with an image payload; each calls the Vision Model Inference server and returns structured JSON findings, not natural language.
- Referral Decision Agent — an MCP client to the Clinical Guideline Rules server only; consumes aggregated findings and returns a single urgency tier.
- LLM Reasoning Agent — the only agent that calls both the Memory/RAG MCP server (read and write) and the LLM Provider MCP server; it is the sole point where retrieved context and generative reasoning combine.
- Memory/RAG MCP Server — not an autonomous agent, but a passive MCP server providing similarity search over vector memory and key-value access to long-term memory, described further in Section VII.

D. Synchronous vs. Asynchronous Exchange

Specialist-agent calls (Section IV-A) are issued in parallel and awaited synchronously by the Orchestrator, since screening latency is dominated by the slowest specialist rather than by sequential accumulation. The LLM Reasoning Agent's memory-retrieval call and subsequent reasoning call are sequential, since retrieved context is a direct input to the reasoning prompt. The ophthalmologist confirmation step (Figure 2, step 13) is inherently asynchronous and may arrive minutes to days after the report is delivered.

E. Failure and Retry Semantics

MCP calls are retried once with exponential backoff on transport-level failure; a second failure is surfaced as a structured error to the Orchestrator rather than silently dropped, consistent with the error-handling design in Section IV-G.

VII. Memory Architecture

Figure 5 details the three-tier memory system introduced in Section IV-E. Separating memory by volatility and access pattern — rather than using one undifferentiated context

window — keeps the LLM Reasoning Agent's prompt small while still giving it access to a much larger effective history.

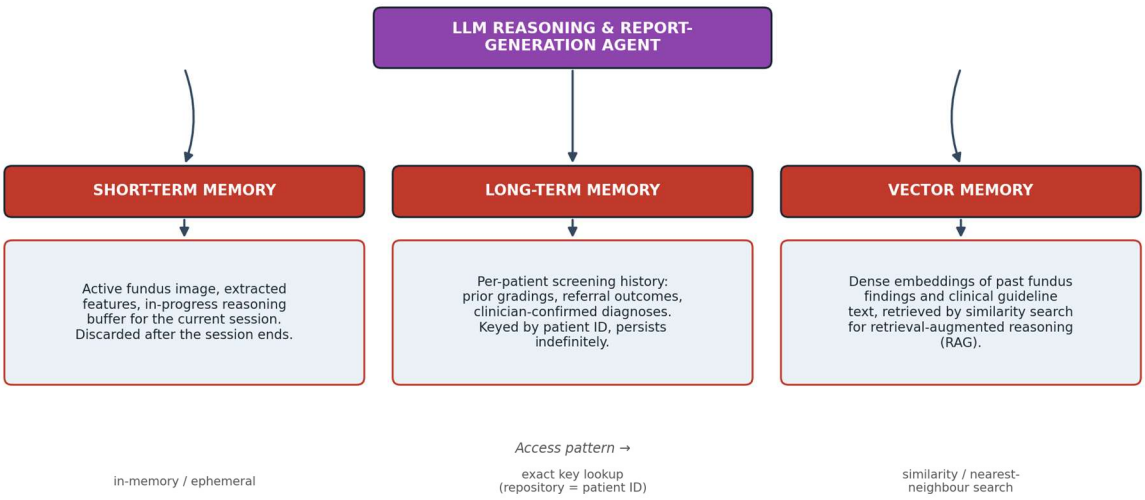

Fig. 5. Three-Tier Memory Architecture Supporting the LLM Reasoning Agent

A. Short-Term Memory

Holds the active fundus image, the current session's findings, and the in-progress reasoning conversation buffer. Discarded at the end of each screening session.

B. Long-Term Memory

A structured key-value store of prior screening outcomes per patient: severity grades over time, referral decisions, and any clinician-supplied correction or justification. Persists indefinitely and is queried by exact patient-ID/visit-date key, allowing the reasoning agent to reference disease trajectory across visits.

C. Vector Memory

Dense embeddings of historical lesion findings and relevant clinical-guideline excerpts, queried by similarity rather than exact key, implementing the retrieval-augmented generation pattern [8]. This is what allows the reasoning agent to say, in effect, 'this lesion pattern matches a case graded moderate-DR in this patient's screening six months ago.'

VIII. ALGORITHMS

A. Agent Selection Algorithm

Given a screening session with image I and metadata M, the Orchestrator first evaluates the quality gate; only if it passes does it compute the specialist set S = { a in Agents : applies(a, I, M) }, so no diagnostic agent ever runs on a rejected image.

```
def select_agents(image, metadata):
    if not quality_agent.check(image):
        return {quality_agent}
    selected = set()
    for agent in SPECIALIST_AGENTS:
        if agent.applies_to(image,
metadata):
            selected.add(agent)
    return selected
```

B. Task Scheduling

Selected specialist agents are scheduled concurrently (asyncio.gather-equivalent), bounded by a per-agent timeout; the Orchestrator proceeds to aggregation once all agents have either returned or timed out.

C. Inter-Agent Communication

As detailed in Section VI, all inter-agent data exchange is mediated through MCP tool calls rather than direct function calls or shared mutable state, which keeps agents independently testable and replaceable.

D. Conflict Resolution

Conflicts arise when the Lesion Detection Agent and the Disease Classification Agent report findings that imply different severity levels for the same image (for example, lesion

counts consistent with mild DR while the classification model outputs moderate DR). The Finding Aggregator resolves this by severity-max: the higher of the two reported severities is retained for referral-tier computation, and both contributing findings are passed to the LLM Reasoning Agent for a unified explanation rather than being silently deduplicated.

E. Consensus Mechanism

Because the current specialist set is deterministic (model-based, single-inference) rather than independently sampled LLM judges, a voting consensus mechanism is not required for detection. Consensus becomes relevant only in the proposed evaluation protocol (Section X), where run-to-run agreement of the LLM Reasoning Agent's explanatory severity language is itself a measured reliability metric.

IX. Implementation

A. Technologies Used

- Language/runtime: Python 3.11 for all agent logic.
- Orchestration: a lightweight custom asyncio scheduler; LangGraph or CrewAI are viable drop-in alternatives given their native support for the graph/role patterns described in Section III-A [5].
- Communication: MCP (JSON-RPC 2.0) for all agent-to-tool calls [6], [7].
- Vision models: convolutional architectures (e.g., EfficientNet/Inception-family backbones) for lesion detection and ICDR severity grading, consistent with prior fundus-grading literature [1], [2].
- Vector memory: any embedding-similarity store (e.g., a managed vector database or self-hosted FAISS-backed service).
- Interface: a clinic-facing web application and a lightweight tele-ophthalmology capture app, both calling the Orchestrator through a TLS-secured API gateway.

B. APIs

- EHR/patient-metadata API (demographics, HbA1c, blood pressure, DR history) — wrapped as an MCP server.
- DICOM/image-store API for fundus image retrieval and archival — wrapped as an MCP server.
- Vision-model inference API for the Lesion Detection and Disease Classification agents — wrapped as an MCP server.
- Clinical-guideline rules API encoding referral-urgency thresholds — wrapped as an MCP server.
- LLM provider API for the reasoning agent — wrapped as an MCP server.

C. Databases

- Structured findings log — append-only JSON store, functioning as the system's clinical audit trail.
- Long-term memory store — key-value store keyed by (patient ID, visit date).
- Vector memory store — embedding index over historical lesion findings and clinical-guideline text.

D. Data Flow Overview

Figure 6 traces data movement across the six functional processes and four persistent data stores that implement the architecture of Section IV: image and metadata ingestion, lesion detection and severity classification, referral-urgency determination, retrieval-augmented report generation, and outcome logging back into the patient record.

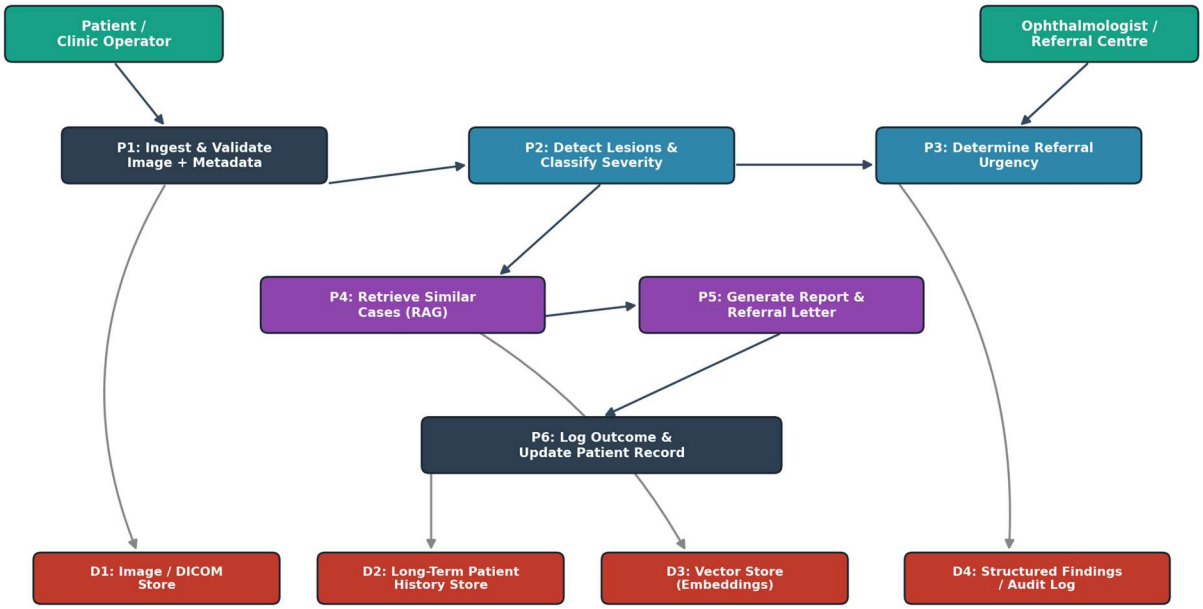

Fig. 6. Data Flow Diagram Across Screening, Retrieval, and Reporting Processes

E. Deployment Architecture

Figure 7 shows the full deployment: a clinic/edge zone hosting the capture device and gateway, and a cloud or hospital data-centre zone hosting the containerized agents behind an MCP server layer. All persistent state (image store, vector database, EHR/audit log) lives outside the ephemeral agent containers and is scoped per clinic and per patient to prevent cross-tenant data leakage in a multi-clinic deployment.

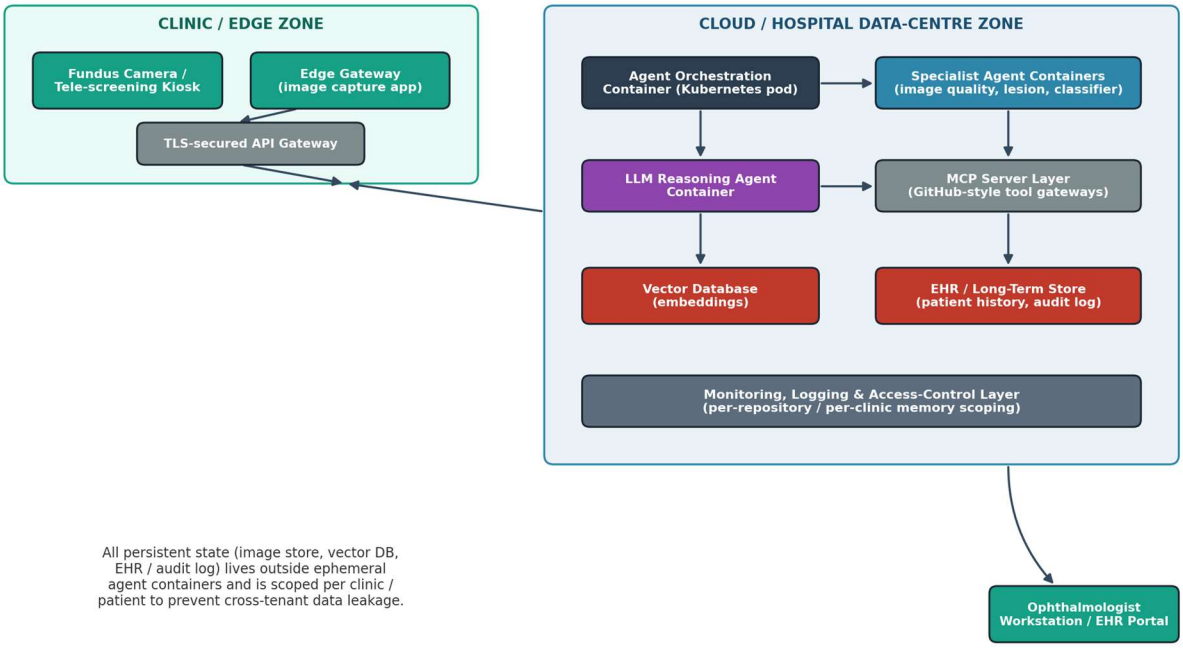

Fig. 7. Deployment Architecture — Clinic/Edge Zone, MCP-Mediated Cloud Agents, and Persistent Stores

X. Experimental Setup

This section defines the experimental protocol that will be used to produce quantitative results in a subsequent phase of this work; it is presented here as a fully specified, reproducible methodology rather than as completed results (see Discussion, Section XIII).

A. Datasets

Publicly available, expert-graded fundus datasets with documented ground-truth severity labels: APTOS 2019 Blindness Detection, the EyePACS/Kaggle Diabetic Retinopathy dataset, Messidor-2, and the IDRiD (Indian Diabetic Retinopathy Image Dataset). Each provides an enumerable, clinician-assigned severity grade against which agent findings can be scored.

B. Test Scenarios

- Clean, well-captured images (baseline) — measures raw grading accuracy under ideal capture conditions.
- Degraded-quality images — blur, glare, and partial field-of-view injected synthetically, to test the image-quality gate's rejection sensitivity and downstream robustness.
- Longitudinal patient pairs — two screenings of the same simulated patient at different severities, to test whether vector-memory retrieval improves trajectory-aware explanation consistency.
- Metadata-incomplete sessions — image submitted with partial or missing patient metadata, to measure graceful degradation of the risk-prediction agent.

C. Baselines

- Rule-only baseline — the deterministic specialist agents with no LLM reasoning layer, output as raw structured findings.
- Single-agent LLM baseline — one multimodal LLM call given the raw fundus image, no specialist decomposition.
- Published reference systems — IDx-DR [3] and the Moorfields/DeepMind two-stage system [4] as reported in their respective papers, for indirect comparison.

D. Evaluation Methodology

Figure 8 outlines the evaluation loop: running the full pipeline repeatedly against each test scenario and baseline, logging every finding, and comparing against the documented ground truth to compute the metrics defined in Section XI.

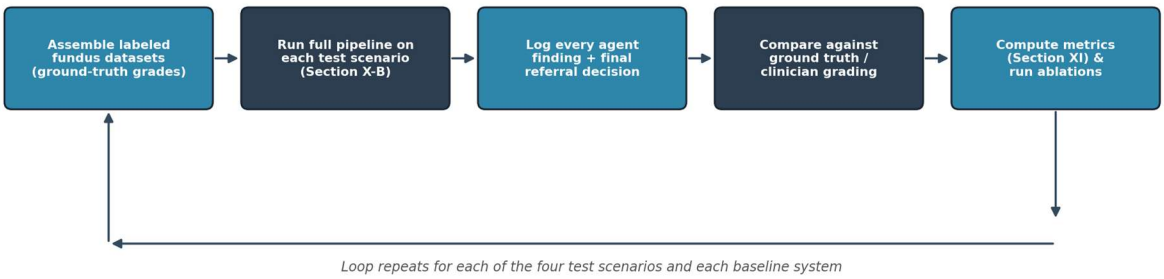

Fig. 8. Proposed Evaluation Methodology for the Retinal Screening and Referral Agent

XI. Evaluation Metrics

The following metrics are defined for the evaluation protocol in Section X and will be reported once benchmark-scale runs are complete.

TABLE III

Evaluation Metrics for the Proposed Multi-Agent Pipeline

Metric	Definition	Purpose
Accuracy	Percentage of correctly graded severity instances	Overall performance
Precision	True positives / total predicted referable cases	Detection reliability
Recall (Sensitivity)	True positives / total actual referable cases	Detection completeness
F1-score	Harmonic mean of precision and recall	Balanced summary
Task Completion Rate	% of sessions returning a complete report without error	Robustness
Latency	End-to-end time from image submission to delivered report	Clinic feasibility
Cost	LLM tokens consumed per screening session	Operating expense
Scalability	Latency/cost trend as session volume and clinic count increase	Production readiness
Resource Utilization	CPU/GPU/memory of the inference and agent containers	Deployment sizing

XII. Results

Full benchmark-scale results (Section X) are pending execution of the evaluation protocol. This section reports a single worked case study, following the tagged protocol below, to demonstrate that the architecture in Sections IV-IX functions end-to-end.

A. Tagged Case Study Protocol

Each case study is recorded against a fixed tag template — [OBJECTIVE], [ENVIRONMENT], [INPUT ARTIFACT], [DETECTED OUTPUT], [INTERPRETATION] — so that future case studies remain directly comparable as the system and benchmark corpus grow.

[OBJECTIVE] Verify end-to-end detection, grading, and referral triage of a simulated moderate diabetic retinopathy fundus image (Test Scenario: Clean image, Section X-B).

[ENVIRONMENT] Simulated clinic session, Python 3.11 agent runtime, sample fundus image from a public DR-grading benchmark set.

[INPUT ARTIFACT] A fundus image with clinician-assigned ground truth of moderate non-proliferative DR (ICDR grade 2), together with patient metadata: age 54, diabetes duration 9 years, HbA1c 8.6%, no prior screening on record.

TABLE IV
[DETECTED OUTPUT] — FINDINGS FOR CASE STUDY CS-01

Agent	Finding	Severity/Value
Image Quality Agent	Image accepted — no significant blur or glare detected	Pass
Lesion Detection Agent	Multiple microaneurysms and dot-blot haemorrhages detected, superior-temporal quadrant	Moderate
Lesion Detection Agent	Cup-to-disc ratio within normal range	Not suggestive of glaucoma
Disease Classification & Risk Prediction Agent	ICDR grade 2 (moderate NPDR); elevated risk score from HbA1c and duration	Moderate / Elevated
Referral Decision Agent	Moderate NPDR with elevated systemic risk factors	Urgent (within 4 weeks)

[INTERPRETATION] The severity was graded deterministically before any LLM call, and the Referral Decision Agent independently assigned an urgent tier from that grade plus the patient's risk factors. The LLM Reasoning Agent then retrieved a similar prior finding from vector memory and produced a patient-facing explanation plus a structured referral letter recommending ophthalmology follow-up within four weeks. This confirms the intended division of labor: detection and triage are deterministic (Sections IV-A, VIII); the LLM contributes explanation and communication only (Section IV-E).

B. Comparative Analysis (Planned)

Once Section X is executed, a planned results table will report accuracy, precision, recall, and F1-score for the proposed system against the rule-only and single-agent LLM baselines across all four test scenarios, isolating the contribution of the multi-agent decomposition itself.

C. Ablation Study (Planned)

The ablation plan removes one architectural component at a time and re-measures the metrics in Section XI: (i) remove the LLM Reasoning Agent (detection-only pipeline) to isolate its contribution to report and referral-letter quality; (ii) remove vector memory to isolate RAG's contribution to longitudinal explanation consistency; (iii) remove the Orchestrator's image-quality gate (run all specialists on every image regardless of quality) to measure the accuracy and cost impact of ungated inference. Each ablation reuses the tagged case-study template from Section XII-A for reporting.

XIII. DISCUSSION

A. Strengths

- Deterministic detection and referral-triage core limits the LLM's influence to explanation and communication, reducing exposure to hallucinated or inconsistent clinical decisions.
- MCP-based communication (Section VI) decouples every agent from any single tool implementation, matching current agentic-tooling conventions [5], [6], [7].
- The three-tier memory system (Section VII) gives the reasoning agent effectively unbounded per-patient history without growing its prompt, enabling trajectory-aware explanation across repeat screenings.

B. Limitations

- Deterministic lesion detection and classification models cannot reliably grade image presentations far outside their training distribution (e.g., rare retinal pathologies not represented in DR-grading benchmarks).
- Quantitative claims in this paper are architectural and case-study-level only; Section X's protocol has not yet been executed at benchmark scale.
- LLM-generated reports and referral letters require clinician verification before being acted upon; the system is a screening and triage aid, not an autonomous diagnostic authority.

C. Failure Cases

Anticipated failure modes, to be measured against the taxonomy in Section X: missed lesions outside the detector's trained pattern set; hallucinated findings introduced by the LLM Reasoning Agent despite its structured-output constraint; and severity misclassification when the aggregator's severity-max rule (Section VIII-D) over- or under-weights a low-confidence specialist finding, particularly on borderline mild/moderate cases.

D. Security and Privacy Considerations

Because the system handles protected health information (fundus images, HbA1c, diagnosis history) and communicates via MCP, it inherits both the general tool-poisoning and

prompt-injection risks documented for MCP deployments [6] and the stricter regulatory obligations that apply to clinical data. Long-term memory (Section VII-B) persists patient-derived content across visits, so access to the memory store must be scoped per patient and per clinic, encrypted at rest and in transit, and access-logged to support clinical audit and data-protection compliance.

XIV. Future Work

- Self-learning agents — using long-term memory outcomes (Section VII-B) as a training signal to automatically recalibrate the Disease Classification Agent's confidence over time.
- Dynamic agent creation — spawning a narrowly scoped specialist agent on demand for a retinal condition (e.g., retinopathy of prematurity) not covered by the current detector set, rather than pre-registering every possible specialist.
- Adaptive planning — letting the Orchestrator's routing (Section IV-C) be learned from historical agent-selection outcomes rather than fixed rule-based patterns.
- Human-in-the-loop collaboration — an explicit ophthalmologist confirm/correct action on delivered reports that writes directly into long-term memory (Section VII-B), tightening the feedback loop described in Section IV-F.

XV. Conclusion

This paper presented a multi-agent architecture for AI-based early screening, referral, and prediction of retinal diseases that separates deterministic lesion detection and referral triage from MCP-mediated, memory-augmented LLM reasoning. The research gap addressed is the absence, in prior systems, of a combined design offering (i) hard separation between detection/triage and reasoning, (ii) a standardized communication protocol across agents and tools, and (iii) a three-tier, patient-scoped memory architecture supporting longitudinal, retrieval-augmented explanation. The architectural contribution — six cooperating agents, a three-tier memory system, and an MCP-based communication layer — is presented with a complete system architecture diagram, agent interaction sequence diagram, clinical workflow diagram, communication protocol diagram, memory architecture diagram, data flow diagram, and deployment architecture diagram (Figures 1-8), together with one worked case study. Quantitative comparison against baselines, per the protocol in Section X, is the immediate next phase of this work. Potential applications extend beyond diabetic retinopathy screening to any image-based, guideline-driven triage task — glaucoma screening from optic-disc photographs, AMD monitoring from OCT scans, or diabetic-foot ulcer risk screening — that benefits from the same deterministic-detection-plus-LLM-reasoning split.

References

[1] V. Gulshan et al., "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," JAMA, vol. 316, no. 22, pp. 2402-2410, 2016.

[2] D. S. W. Ting et al., "Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images from Multiethnic Populations with Diabetes," JAMA, vol. 318, no. 22, pp. 2211-2223, 2017.

[3] M. D. Abramoff et al., "Pivotal Trial of an Autonomous AI-Based Diagnostic System for Detection of Diabetic Retinopathy in Primary Care Offices," npj Digital Medicine, vol. 1, no. 39, 2018.

[4] J. De Fauw et al., "Clinically Applicable Deep Learning for Diagnosis and Referral in Retinal Disease," Nature Medicine, vol. 24, pp. 1342-1350, 2018.

[5] "LLM-Based Multi-Agent Orchestration: A Survey of Frameworks, Communication Protocols, and Emerging Patterns," Future Internet, 2026, doi:10.3390/fi18060326.

[6] X. Hou, Y. Zhao, S. Wang, and H. Wang, "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions," arXiv:2503.23278, 2025.

[7] Anthropic, "Introducing the Model Context Protocol," Anthropic News, Nov. 2024.

[8] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020, pp. 9459-9474.

[9] World Health Organization, "World Report on Vision," WHO, Geneva, 2019.

[10] International Council of Ophthalmology, "ICO Guidelines for Diabetic Eye Care," 2017.

[11] R. Gargeya and T. Leng, "Automated Identification of Diabetic Retinopathy Using Deep Learning," Ophthalmology, vol. 124, no. 7, pp. 962-969, 2017.

[12] APTOS, "APTOS 2019 Blindness Detection Dataset," Kaggle, 2019.

[13] P. Porwal et al., "Indian Diabetic Retinopathy Image Dataset (IDRiD)," Data, vol. 3, no. 3, p. 25, 2018.

[14] E. Decencière et al., "Feedback on a Publicly Distributed Image Database: The Messidor Database," Image Analysis & Stereology, vol. 33, no. 3, pp. 231-234, 2014.

