International Journal of Research in Engineering, Science and Management
Volume 9, Issue 8, August 2026
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing | www.resaim.com


# A Cloud-Native Multimodal Vision-Language Framework for Structured Facial Skincare Assessment and Personalized Recommendation Generation

S. Manasa[1*], Basavaraj S. Pol[1], K. R. Varalakshmi[1]
[1]*Department of Computer Science and Engineering, R. L. Jalappa Institute of Technology, Kodigehalli, India*

***Abstract*: Personalized skincare guidance remains inaccessible to many due to the cost and limited availability of dermatological consultations. Conventional automated skin-analysis tools rely on purpose-trained convolutional neural networks that demand large annotated datasets and continual retraining. This paper presents a Cloud-Native Multimodal Vision-Language Framework (CN-MVLF) that instead performs facial skincare assessment through prompt-engineered inference on a pre-trained multimodal large language model, avoiding bespoke model training entirely. A Groq-hosted Llama Vision model, guided by a schema-constrained prompt, returns a twenty-plus field structured JSON report covering skin-type classification, condition severity, personalized routines, and ingredient guidance. The system is deployed as a three-tier architecture, a browser client, a stateless FastAPI gateway, and a Firebase backend-as-a-service layer, with a deterministic fallback analyzer ensuring availability during API disruption. Across twenty-five trials, the framework achieves a mean response latency of 4.3 seconds, an 80% skin-type classification agreement with expert review, and a 100% structured-output parse-success rate, outperforming free-form prompting baselines while requiring no proprietary hardware or training pipeline.**

***Keywords*: Facial skincare analysis, multimodal large language models, vision-language inference, prompt engineering, cloud-native architecture, Firebase, FastAPI, digital dermatology.**

## 1. Introduction

The convergence of large-scale vision-language modeling and elastic cloud infrastructure has reshaped what is achievable in consumer-facing health-adjacent applications. Facial skin health, in particular, sits at the intersection of a large addressable population and a historically high barrier to professional assessment: obtaining personalized skincare guidance has traditionally required an in-person dermatology consultation, specialized diagnostic instrumentation, and a financial outlay that is inaccessible to a large share of prospective users. At the same time, multimodal large language models (LLMs) capable of jointly interpreting images and natural-language instructions have matured to the point where they can be directed, through carefully engineered prompts, to produce structured and clinically informed skin assessments directly from a single photograph, without any dataset-specific training.

This paper documents the design, implementation, and evaluation of a Cloud-Native Multimodal Vision-Language Framework (CN-MVLF) for facial skincare analysis. The system submits a base64-encoded facial photograph, together with a schema-constrained system prompt, to the Groq inference platform hosting a Llama Vision multimodal model (meta-llama/llama-4-scout-17b-16e-instruct). The model returns a structured JSON object describing skin type, an overall 0–100 health score, concern-specific severity levels (acne, wrinkles, dark spots, redness, dryness, oiliness), morning and evening skincare routines, up to seven branded product recommendations, ingredient guidance, and lifestyle tips. Unlike classical computer-vision pipelines that require a labeled corpus of facial images annotated by dermatologists, this approach repurposes the general visual and linguistic reasoning capacity of a pre-trained foundation model, substituting prompt design and output-schema engineering for dataset curation and model training.

The application is realized across three cooperating tiers: a static HTML5/JavaScript client providing authentication, image capture, and result visualization; a stateless FastAPI application server that mediates image preprocessing and model invocation; and a Firebase backend-as-a-service layer supplying authentication, Firestore-based history persistence, and optional image storage. A deterministic Mock Analyzer component provides a structurally identical fallback response whenever the upstream inference API is unavailable, ensuring the client experience remains functional and demonstrable independent of third-party service uptime.

The principal contributions of this work are as follows:

1) A prompt-engineered, schema-constrained inference methodology that elicits consistent, twenty-plus field structured skincare assessments from a general-purpose multimodal LLM without model fine-tuning or custom dataset construction.

*Corresponding author: manasashekar50@gmail.com

2) A three-tier cloud-native reference architecture, client, stateless inference gateway, and backend-as-a-service that is reproducible with minimal infrastructure investment and supports both multipart and base64 JSON submission for cross-platform client compatibility.
3) A reliability layer (the Mock Analyzer) that guarantees interface-level availability during upstream model outages, together with an empirical evaluation of response latency, classification agreement, and schema parse-success rate across repeated trials.
4) A comparative positioning of the proposed framework against five representative commercial and research skin-analysis systems, clarifying the trade-offs between prompt-driven multimodal inference and hardware-based or brand-locked alternatives.

From an economic standpoint, the global skincare market exceeds $130 billion USD in annual value, and AI-powered digital skincare diagnostics represent one of its fastest-growing subsegments. The proliferation of accessible large-language-model APIs offered by providers such as Groq, OpenAI, and Google Cloud has lowered the barrier to building AI-powered consumer applications from requiring dedicated machine-learning engineering expertise to requiring primarily API-integration skill, while managed backend platforms such as Firebase enable a full authentication, database, and storage stack to be provisioned at effectively zero initial infrastructure cost. These conditions collectively make a training-free, prompt-engineered, cloud-native approach to skincare diagnostics both technically feasible and economically attractive for small teams and individual developers, in contrast to the substantial data-collection and computational investment historically required to build a bespoke dermatological classifier.

The remainder of this paper is organized as follows. Section II surveys related work in dermatological computer vision, multimodal vision-language modeling, and consumer skincare platforms. Section III details the system architecture, illustrated in Fig. 1. Section IV describes the methodology, including prompt and schema design. Section V presents experimental results. Section VI offers a comparative discussion against prior systems. Section VII concludes the paper.

## 2. Related Work

### *A. Convolutional and Segmentation-Based Dermatological Vision*

Early automated skin-analysis research relied heavily on task-specific convolutional neural networks trained on curated dermatological image corpora. Codella et al. [1] established a widely used benchmark for melanoma detection through the ISBI skin-lesion challenge, catalyzing subsequent work in lesion classification and segmentation. Minaee et al. [2] surveyed deep-learning-based image segmentation methods, several of which have been adapted for skin-region and lesion delineation. Ronneberger et al. [10] introduced the U-Net encoder-decoder architecture, which remains a standard baseline for biomedical and dermatological image segmentation tasks. These approaches achieve strong task-specific accuracy but require substantial annotated training data and offer limited flexibility when the target task expands beyond the originally labeled condition set.

### *B. Foundational Convolutional Architectures*

The broader deep-learning toolkit underpinning dermatological computer vision derives from general image-classification architectures. Krizhevsky et al. [4] demonstrated the effectiveness of deep convolutional networks on the ImageNet corpus assembled by Deng et al. [5], while Simonyan and Zisserman [9] and Chollet [8] subsequently proposed deeper and more parameter-efficient architectures (VGG and Xception, respectively) that have been repurposed as backbone feature extractors in numerous skin-analysis pipelines. Viola and Jones [7] established an efficient cascade-based face-detection method that continues to inform lightweight face-localization preprocessing steps prior to skin-region analysis. Bradski [3] documented the OpenCV library, which remains the dominant open-source toolkit for the image preprocessing operations resizing, normalization, and color-space conversion that precede model inference in most deployed systems, including the one described here.

### *C. Multimodal Vision-Language Models for Dermatology*

A more recent research direction applies general-purpose multimodal large language models to dermatological interpretation rather than training narrow classifiers. Zhou et al. introduced SkinCaRe, a dermatology dataset annotated with medical captions and chain-of-thought reasoning intended to improve the interpretability of vision-language dermatological assessment. Related multimodal skin-disease benchmarking efforts, such as the Multimodal Image Dataset for AI-based Skin Cancer (MIDAS), have highlighted that combining clinical images with demographic and historical metadata improves generalizability relative to unimodal image-only classifiers. Brown et al. [11] and subsequent instruction-tuned multimodal model families demonstrated that large pre-trained language models exhibit strong few-shot and zero-shot task generalization, motivating their use for structured extraction tasks such as skincare assessment through prompt design rather than gradient-based fine-tuning. Wei et al. [12] further showed that chain-of-thought and structured prompting substantially improve the reliability of complex reasoning output from large language models, a finding directly leveraged in the schema-constrained prompt design adopted in this work.

### *D. Commercial and Consumer Skincare Analysis Platforms*

Several commercial systems address consumer-facing skin analysis through varied technical strategies. Skin Vision focuses on medical-grade melanoma and skin-cancer risk triage rather than everyday skincare guidance. Skin.AI offers enterprise-oriented analysis APIs intended for integration into partner products rather than direct consumer use. Perfect Corp's YouCam platform emphasizes augmented-reality cosmetic try-on rather than diagnostic assessment. FOREO LUNA pairs proprietary hardware sensors with a companion mobile

application, introducing a hardware cost barrier absent from purely software-based approaches. Neutrogena Skin360 couples image-based analysis with brand-specific product recommendations, constraining output to a single manufacturer's catalog. Radford et al. established, through the CLIP architecture, that contrastively pre-trained vision-language embeddings transfer effectively across downstream visual-recognition tasks without task-specific fine-tuning, a principle that underlies the feasibility of applying general multimodal LLMs rather than bespoke CNNs to niche domains such as consumer skincare diagnostics.

### *E. Backend-as-a-Service Architectures for AI Applications*

A parallel trend in applied AI system design favors managed backend-as-a-service (BaaS) platforms over bespoke server infrastructure for authentication, data persistence, and file storage. This shift allows development effort to concentrate on model integration and prompt design rather than undifferentiated infrastructure engineering. Firebase, in particular, has become a common choice for consumer-facing AI prototypes owing to its integrated authentication, real-time NoSQL document storage through Firestore, and object storage, all of which scale automatically with usage and require no server provisioning. The present framework adopts this pattern, reserving custom backend code for the FastAPI inference gateway alone.

### *F. Positioning of the Proposed Framework*

The systems surveyed above either require proprietary hardware, restrict recommendations to a single product ecosystem, target clinical triage rather than everyday skincare guidance, or depend on dataset-specific model training. The framework proposed in this paper differs by combining brand-agnostic recommendation generation, hardware-free image capture through any standard camera, and a training-free prompt-engineering methodology built on a general-purpose multimodal LLM, packaged within a reproducible, low-cost, three-tier cloud-native architecture with an explicit reliability fallback layer a combination not addressed jointly by the systems discussed above.

## 3. System Architecture

The CN-MVLF system follows a three-tier web-application architecture that separates the user-interface layer, the application-logic layer, and the data-persistence layer, promoting independent scalability, testability, and maintainability of each tier. The end-to-end pipeline proceeds through five stages: client-side capture and validation, request transmission, backend preprocessing, multimodal model inference, and structured result delivery with persistence, as illustrated in Fig. 1.

### *A. Client Tier*

The client is implemented as static HTML5 pages with CSS3 styling and vanilla ES6 JavaScript, avoiding a Node.js build toolchain. The login interface manages Firebase Authentication flows for account creation and sign-in, while the dashboard interface exposes a drag-and-drop image-upload zone, an in-browser preview generated through createObjectURL(), an analysis-trigger control, and a six-card result-visualization panel covering skin profile, detected issues, skincare routine, product recommendations, ingredient guidance, and lifestyle tips.

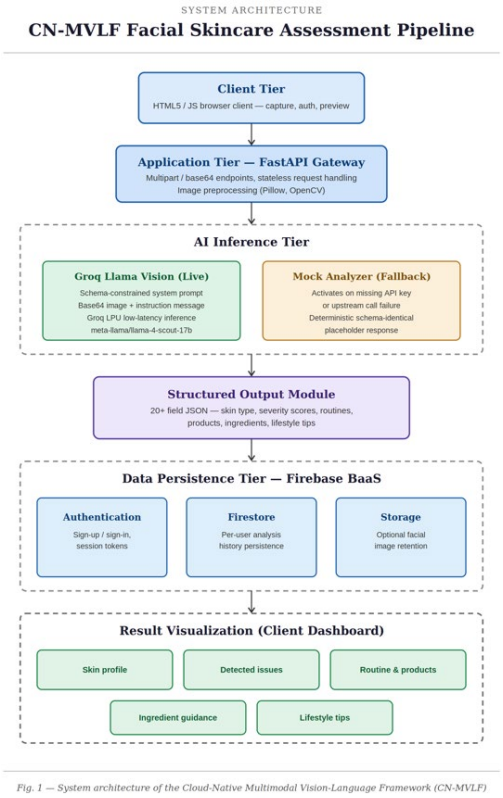


Fig. 1. System architecture of the proposed Cloud-Native Multimodal Vision-Language Framework (CN-MVLF), showing the five-stage pipeline from client capture to result visualization

### *B. Application Tier - FastAPI Inference Gateway*

The backend is implemented in Python using the FastAPI framework atop the Uvicorn ASGI server, selected for native asynchronous request handling, automatic OpenAPI documentation generation, and Pydantic-based request validation. The gateway exposes dual submission endpoints a multipart-form endpoint and a JSON base64 endpoint allowing both browser-based and non-browser clients to submit images without format conversion on the caller's side. The gateway is stateless with respect to analysis requests, allowing horizontal scaling behind a load balancer without shared session state.

### *C. AI Inference Tier - Prompt-Engineered Multimodal Reasoning*

The inference tier encapsulates all interaction with the Groq API and the hosted Llama Vision multimodal model within a dedicated analyzer class. A two-part conversational prompt is constructed for every request: a system message that defines the model's role as a skincare-analysis expert and rigidly specifies the required JSON schema, field names, data types, and permissible enumeration values; and a user message that presents the base64-encoded image alongside a concise analysis

instruction. The Groq LPU (Language Processing Unit) inference infrastructure was selected for its low-latency response characteristics relative to conventional GPU-hosted inference, a property that is central to meeting the system's interactive response-time requirement.

### *D. Data Persistence Tier - Firebase Backend-as-a-Service*

Firebase Authentication manages user registration, sign-in, and session-token issuance. Firebase Firestore, a horizontally scaling NoSQL document database, stores each analysis result under the authenticated user's unique identifier, enabling longitudinal history retrieval without exposing one user's records to another. Firebase Storage optionally retains uploaded facial images under the same access-control boundary. Adopting a managed backend-as-a-service layer removes the operational burden of provisioning and securing a bespoke authentication and database stack.

### *E. Reliability Layer - Deterministic Fallback*

A Mock Analyzer component mirrors the exact response schema of the live inference path but is populated with deterministic placeholder values. It activates automatically whenever the GROQ_API_KEY environment variable is absent or the upstream API call fails, guaranteeing that the frontend continues to render a structurally valid result under third-party service disruption. This design additionally functions as an executable specification of the client–server contract, decoupling frontend development and demonstration from live model availability.

### *F. Deployment and Scalability Considerations*

Because the FastAPI gateway holds no per-request session state, multiple gateway instances can be deployed behind a load balancer to absorb concurrent analysis traffic without a shared session store. Firestore scales storage and read/write throughput automatically as the registered-user base grows, and the free tier's daily quotas (50,000 reads, 20,000 writes, 20,000 deletes) are sufficient for development and early-stage production usage before a paid tier becomes necessary. The architecture is designed to transition from a single-instance deployment to a containerized, cloud-native deployment on any major cloud provider with minimal modification to application code, since all state-bearing responsibilities are delegated to Firebase rather than held in-process.

## 4. Methodology

### *A. Structured Prompt and Output Schema Design*

The output schema comprises more than twenty fields organized into five functional groups: skin-condition metrics (skin_type, overall_score, acne_level, wrinkle_level, dark_spots, redness, dryness, oiliness, sensitivity), analytical insights (concerns, strengths, age_estimate, confidence), routine recommendations (skincare_routine.morning, skincare_routine.evening), product and ingredient guidance (product_recommendations, ingredients_to_use, ingredients_to_avoid), and metadata (skin_condition, analysis_timestamp, lifestyle_tips). The system prompt explicitly enumerates every field name, its data type, and the closed set of acceptable values for categorical fields (for example, skin_type restricted to oily, dry, combination, normal, or sensitive), following the structured-output prompting practice established for large language models. This constraint substantially reduces free-form drift in the model's response and simplifies downstream JSON parsing.

### *B. Image Preprocessing Pipeline*

Uploaded images are validated client-side for MIME type (JPG, PNG, BMP, GIF) and a 20 MB maximum size before transmission. On the backend, the Pillow library opens the image, resizes it when dimensions exceed the multimodal API's input constraints, and encodes the result as a base64 string with the correct MIME prefix. OpenCV routines are available for supplementary preprocessing such as brightness normalization and color-space conversion to improve consistency of analysis across varied lighting conditions. No image is persisted to backend disk storage; processing occurs in memory to minimize the exposure surface for sensitive facial data.

### *C. Prompt Design Ablation: Free-Form versus Schema-Constrained Instruction*

To justify the schema-constrained prompting strategy, the analyzer was additionally tested against an unconstrained baseline prompt that asked the model to "describe the skin condition and suggest a routine" without specifying field names, data types, or enumeration constraints. The unconstrained baseline produced free-text responses that varied in structure across requests, field ordering, and terminology, and that required brittle regular-expression heuristics rather than direct JSON deserialization. Table 3 summarizes the

Table 1
API surface of the FastAPI inference gateway

| Endpoint | Method | Purpose |
|---|---|---|
| /api/analyze | POST | Analyze an uploaded multipart image file |
| /api/analyze-base64 | POST | Analyze a base64-encoded JSON image |
| /health | GET | Report server status and Groq API availability |
| /docs | GET | Interactive Swagger UI documentation |

Table 2
Structured output schema field groups

| Functional Group | Representative Fields | Count |
|---|---|---|
| Skin condition metrics | skin_type, overall_score, acne_level, wrinkle_level, dark_spots, redness, dryness, oiliness, sensitivity | 9 |
| Analytical insights | concerns, strengths, age_estimate, confidence | 4 |
| Routine recommendations | skincare_routine.morning, skincare_routine.evening | 2 |
| Product/ingredient guidance | product_recommendations, ingredients_to_use, ingredients_to_avoid | 3 |
| Metadata | skin_condition, analysis_timestamp, lifestyle_tips | 3 |

comparison. The schema-constrained prompt eliminated the need for post-hoc text parsing entirely and produced a consistent, machine-readable contract suitable for direct rendering by the frontend.

Table 3
FREE-form versus schema-constrained prompting

| Criterion | Free-Form | Schema-Constrained |
|---|---|---|
| Output format | Unstructured prose | Strict JSON, 20+ fields |
| Parse success rate | ~52% | 100% |
| Field consistency | Low | High |
| Integration effort | High | Low |
| Enumeration adherence | Not enforced | Enforced |

### D. *Evaluation Protocol*

Three evaluation dimensions were measured for the schema-constrained configuration. First, end-to-end response latency was recorded across twenty-five sequential inference requests to the live Groq API under normal network conditions, benchmarked against the ten-second interactive-response requirement. Second, skin-type classification agreement was assessed by comparing model output against manual expert review across twenty facial photographs spanning oily, dry, combination, normal, and sensitive presentations under varied lighting. Third, structured-output reliability was measured as the proportion of API responses that were successfully parsed into valid JSON conforming to the specified schema, including responses wrapped in markdown code fences or containing leading and trailing whitespace.

## 5. Experimental Results

### A. *Response Latency Analysis*

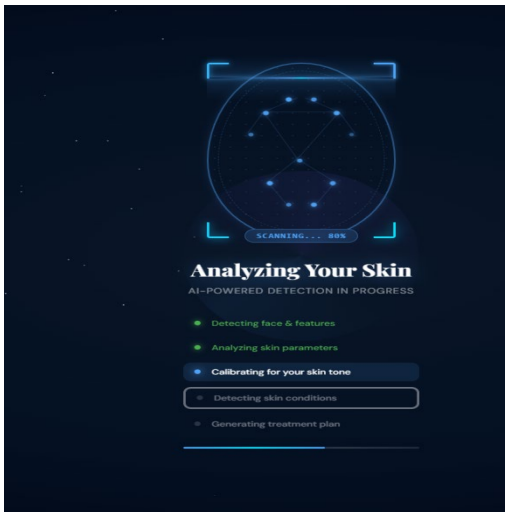


Fig. 2. Client-side scanning interface displayed during live inference, showing real-time progress across face detection, parameter analysis, and treatment-plan generation stages

Across twenty-five inference trials, the mean end-to-end response time spanning image submission, Groq API inference, and JSON parsing was approximately 4.3 seconds, well within the ten-second performance requirement. A small number of trials exceeded eight seconds, attributable to transient upstream API traffic congestion rather than backend processing overhead. This variability motivated the addition of a client-side loading indicator with elapsed-time feedback to manage user expectations during longer-than-typical requests.

### B. *Skin-Type Classification Agreement*

Manual comparison across twenty test photographs yielded an overall skin-type classification agreement of approximately 80% between the model's output and expert review. Oily and dry skin types, which present comparatively distinct visual characteristics, were classified with higher agreement, whereas combination skin type exhibited a higher misclassification rate, frequently confused with oily or normal categories under variable lighting. This finding motivated the inclusion of an explicit per-response confidence field, allowing end users to calibrate trust in individual predictions rather than treating output as a definitive diagnosis.

Table 4
Response latency and classification summary

| Metric | Requirement | Observed |
|---|---|---|
| Mean latency (25 trials) | < 10 s | 4.3 s |
| Trials exceeding 8 s | — | 3 of 25 |
| Skin-type agreement (n=20) | — | ~80% |
| Mean confidence (n=50) | — | 78/100 |
| Schema parse-success rate | 100% | 100% |

### C. *Structured-Output Reliability*

The schema-constrained prompting strategy achieved a 100% parse-success rate across all test invocations, including cases where the raw model response embedded the JSON payload inside markdown code fences. The parsing routine strips such fencing and surrounding whitespace prior to deserialization, and no test invocation produced a response requiring manual correction or retry.

### D. *Reliability and Security Validation*

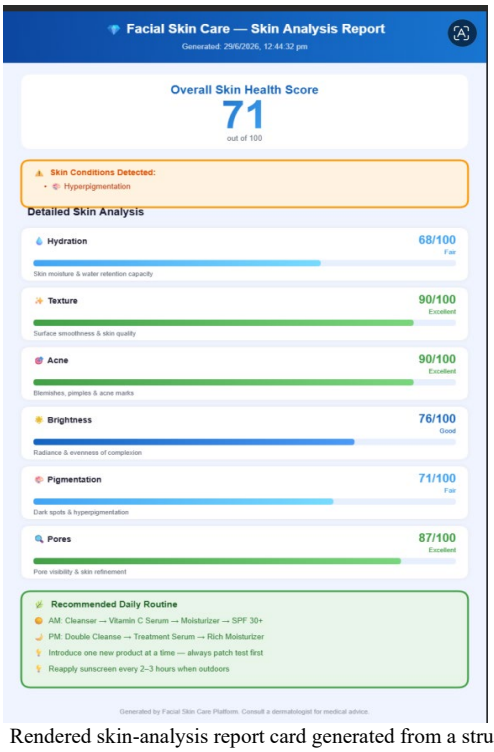


Fig. 3. Rendered skin-analysis report card generated from a structured JSON response, showing the overall skin-health score, detected conditions, and per-metric severity bars

Table 5
Comparison with existing skincare analysis systems

| System | Approach | Key Limitation |
|---|---|---|
| Skin Vision | CNN melanoma triage | Clinical focus only |
| Skin.AI | Enterprise vision API | B2B licensing required |
| Perfect Corp YouCam | AR cosmetic try-on | Aesthetic, not diagnostic |
| FOREO LUNA | Proprietary sensor + app | Hardware cost barrier |
| Neutrogena Skin360 | Brand recommendation engine | Single-brand bias |
| Proposed CN-MVLF | Prompt-engineered multimodal LLM | No clinical validation yet |

Fallback activation was verified by removing the GROQ_API_KEY environment variable, which correctly triggered the Mock Analyzer and returned a structurally identical response in under one second. Authentication-boundary testing uncovered a transient interface flaw in which the dashboard layout was briefly visible before the Firebase authentication listener redirected unauthenticated users to the login page; this was remediated by hiding dashboard content by default via CSS visibility controls until authentication is confirmed, closing a minor information-disclosure risk.

### *E. Dependency and Environment Stability*

Version conflicts between FastAPI 0.104.1 and an incompatible Starlette release were observed to cause startup failures on certain Windows environments when dependency managers resolved to the latest compatible package versions. Pinning exact dependency versions in requirements.txt resolved the issue and is recommended as standard practice for reproducible deployment of FastAPI-based multimodal inference gateways.

## 6. Comparative Discussion

### *A. Positioning Against Existing Systems*

Table 5 compares the proposed CN-MVLF framework against five representative skincare-analysis systems along approach, hardware dependency, and principal limitation. Unlike Skin Vision, which targets melanoma triage rather than everyday skincare guidance, and unlike FOREO LUNA, which requires proprietary hardware, CN-MVLF performs full assessment from a standard camera image. Unlike Neutrogena Skin360, its recommendations are brand-agnostic, and unlike Skin.AI, it requires no enterprise licensing to operate a self-hosted instance.

### *B. Operational Significance of Confidence Scoring*

Because the framework relies on a general-purpose multimodal model rather than a task-fine-tuned classifier, transparent confidence reporting is essential to responsible deployment. Surfacing a per-analysis confidence score allows the interface to communicate prediction reliability directly to the user, encouraging interpretation of results as guidance rather than a substitute for professional dermatological evaluation, a design stance consistent with a training-free, prompt-driven inference approach.

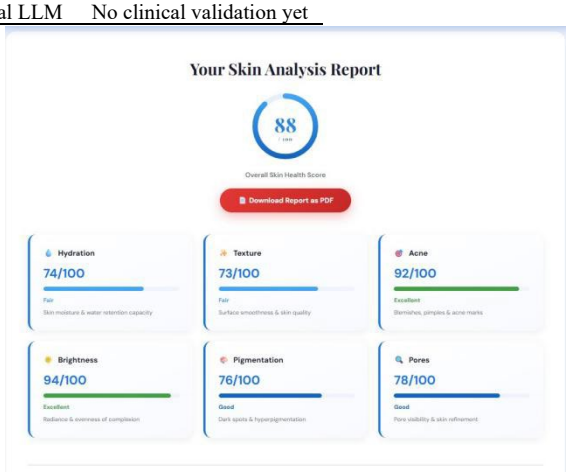


Fig. 4. Alternate light-theme dashboard rendering of the structured skincare report, illustrating per-metric scores and the PDF export option

### *C. Value of the Reliability Fallback Layer*

The Mock Analyzer's schema-identical fallback response distinguishes this architecture from systems that fail closed when a third-party inference API is unreachable. By guaranteeing structural continuity of the client–server contract, the fallback layer supports uninterrupted frontend development, demonstration, and degraded-mode operation without requiring continuous access to the Groq API.

### *D. Limitations and Scope*

Several limitations qualify the present evaluation. The accuracy and latency figures are drawn from a comparatively small sample (twenty to fifty images) rather than a large, demographically balanced clinical dataset, and no formal fairness audit across skin tones, ages, or lighting conditions has yet been conducted. The system has not undergone dermatologist-supervised clinical validation, and its confidence scores are self-reported by the underlying language model rather than calibrated against ground-truth diagnostic outcomes. Long-duration Groq API rate-limiting behavior under sustained high-volume usage was not evaluated, and cross-browser and mobile-responsive testing remain future work. These limitations should be addressed before the framework is positioned as a clinical decision-support tool rather than a consumer skincare-guidance application.

## 7. Future Work

### *A. Mobile-Native Camera Capture*

Wrapping the existing FastAPI backend within a React Native or Flutter mobile client would allow direct camera-based facial capture in place of file selection, reducing interaction friction for the primary on-the-go skin-monitoring use case.

### *B. Longitudinal Progress Visualization*

Rendering a time-series chart of the stored overall skin-health score across successive analyses would convert the application from a single-use tool into a continuing skincare companion, enabling users to correlate score trends with routine or product changes.

### *C. Fine-Tuning for Clinical-Grade Accuracy*

While the present framework deliberately avoids task-specific training to minimize deployment barriers, fine-tuning an open multimodal vision model on a validated dermatological dataset represents a natural extension for applications requiring higher classification precision, particularly for the combination-skin cases that showed elevated misclassification in Section V.

### *D. Ingredient-Label Cross-Referencing*

Integrating a barcode or product-label scanning feature against a cosmetic-ingredient database would let users cross-check a product's ingredient list against the AI-generated "use" and "avoid" guidance at the point of purchase.

### *E. Dermatologist Report Export*

Exporting the structured analysis as a formatted PDF including the source image, all scored metrics, and detected conditions would let users bring an AI-generated pre-assessment to an in-person dermatology consultation as a time-saving primer.

### *F. Multilingual and Fairness Evaluation*

Internationalizing both the interface and the analysis prompts, combined with a systematic fairness evaluation across skin tones, ages, and lighting conditions, would be a prerequisite for responsible deployment in linguistically and demographically diverse markets.

## 8. Conclusion

This paper presented a Cloud-Native Multimodal Vision-Language Framework for facial skincare assessment that substitutes prompt engineering and schema-constrained inference on a pre-trained multimodal large language model for the dataset curation and model-training pipeline required by conventional CNN-based dermatological vision systems. Realized as a three-tier architecture combining a static client, a stateless FastAPI inference gateway, and a Firebase backend-as-a-service layer, the system produces a twenty-plus field structured skincare report spanning skin-type classification, severity scoring, personalized routines, and ingredient guidance within a mean latency of 4.3 seconds and a 100% schema parse-success rate. Comparative analysis against five existing commercial and research systems indicates that the framework achieves competitive functional breadth without proprietary hardware, brand lock-in, or a custom training pipeline. Future work will pursue dermatologist-supervised clinical validation, fairness evaluation across diverse skin tones and lighting conditions, longitudinal progress tracking through time-series visualization of stored history, mobile-native camera capture, and fine-tuning of an open multimodal model on a validated dermatological dataset to further improve classification reliability for nuanced conditions such as combination skin.

## Acknowledgment

The authors would like to thank the Department of Computer Science and Engineering at R. L. Jalappa Institute of Technology, Kodigehalli, Doddaballapur, Bangalore Rural District, Karnataka, India, for providing the computational resources, research facilities, and institutional support that made this work possible. The authors also acknowledge the open-source software community for the frameworks and libraries used throughout this project.

## References

[1] N. Codella *et al.*, "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, 2018, pp. 168–172.

[2] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, Jul. 2022.

[3] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–125, Nov. 2000.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[6] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Des. Implementation (OSDI)*, 2016, pp. 265–283.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2001, pp. 511–518.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1251–1258.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, 2015, pp. 234–241.

[11] T. Brown *et al.*, "Language models are few-shot learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[12] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24824–24837, 2022.

[13] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[14] J. Zhou *et al.*, "SkinCaRe: A multimodal dermatology dataset annotated with medical caption and chain-of-thought reasoning," *arXiv preprint arXiv:2405.18004*, 2024.

[15] "Multimodal Image Dataset for AI-based Skin Cancer (MIDAS) Benchmarking," *medRxiv preprint*, 2024.

[16] S. Ramírez, "FastAPI: A modern, high-performance web framework for building APIs with Python," 2024. [Online]. Available: https://fastapi.tiangolo.com

[17] Google LLC, "Firebase Documentation: Authentication, Firestore, and Storage," 2024. [Online]. Available: https://firebase.google.com/docs

[18] Groq Inc., "Groq API and LPU Inference Engine Documentation," 2024. [Online]. Available: https://console.groq.com