WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

A Cloud-Based AI Skin Analyzer with Hybrid Cosmetic Recommendation Using Facial Image Analysis

Shrawani Sable [a], Sejal shende [a], Diya Tekale [a], Bhavika Gillurkar [a], Shrawani Ghadge [a], Priyanka Wasnik [a]

[a] *Information technology, Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra*
E-mails: shrawani.sable09@gmail.com, sejalshende269@gmail.com, diyatekale@gmail.com, bhavikagillurkar27@gmail.com, shrawanighadge30@gmail.com, pbcoepriyanka@gmail.com

A B S T R A C T

Professional skin assessment is limited by cost of the assessment, distance to the service and wait time and most people select cosmetic products without any knowledge of their skin. This paper introduces a system which is implemented in web-based platform that is capable of analyzing one facial photograph and returning a skin type, a list of skin conditions and a ranked list of cosmetic products. This system consists of three parts: a quality controlled image pre-processing stage, an image processing stage for face detection and regional segmentation based on MTCNN and an OpenCV fallback, and a convolutional neural network (CNN) classification stage for five skin types and four skin conditions. Following a hard filter which excludes products containing the declared allergens, a hybrid recommendation engine combines content score based on ingredients and profile-similarity score. The platform's front-end is built using React and TypeScript, while the back-end is coded with Flask. Supabase offers authentication, PostgreSQL storage, and image storage, and a large language model chatbot which answers questions about skincare using the user's own profile. We detail out the design, algorithms, and deployment, and verify system-level the components implemented and give a worked example of the scoring. The current prototype implements a mock inference path and has not yet been evaluated for classification accuracy, therefore we outline an evaluation protocol and openly discuss limitations. The proposed system demonstrates that an Open and Explainable service for the analysis of the skin can be constructed and deployed on a serverless infrastructure while respecting privacy.

Keywords: skin analysis, convolutional neural network, face detection, recommender system, cosmetic recommendation, cloud computing, web application

Introduction

Acne, hyper pigmentation, bigger pores, wrinkles and many other skin problems occur in every age group and often have an impact on the confidence and daily life of the person. Diagnosis is usually performed in a clinic, which comes with a high cost, a lot of time or simply is not available in many areas. This means that many consumers opt for cosmetic products which are advertised, promoted through social media or through trial and error. Substances not compatible with a person's skin tone can exacerbate the very issues they were designed to help alleviate, as can ingredients that cause allergies or other reactions.

The recent progress in deep learning has demonstrated that deep convolutional neural networks (CNNs) are able to perform a series of dermatological tasks at a level similar to that of a dermatologist [1, 2]. Together with this, there have been significant developments of lightweight network architectures [7] and mature face detectors [6] that have made it possible to execute image analysis within a standard web service. Now missing is an end-to-end consumer system that brings these three factors together: an objective assessment of the user's skin from a photo, a recommendation that considers the allergenies the user has declared, and a conversational interface that explains the result in a straightforward manner.

This paper presents an AI-based skin analysis and cosmetics recommendation system. The user signs up with a short skin profile, uploads or takes a photo of his face, and the facial image is segmented and analyzed, giving the user a skin type, a list of detected conditions with a level of confidence and a bounding box, and up to ten recommended products, ranked by the level of confidence that they are suitable for that skin type. The user can also download a PDF report, and pose additional questions to a chatbot that knows the profile and the latest analysis.

The contributions of this work are as follows.

1. End-to-end architecture for photograph-based skin analysis, dividing the image quality control, face localization, classification and recommendation stages into independent testable ones.

2. Region aware condition detector that calculates the score of four facial regions (forehead, left cheek, right cheek, and chin) and combines them with a max pooling method, thus attributing each reported condition to the region with the maximum score.
3. A hybrid recommender using a hard allergen filter and a weighted combination of scores based on ingredients and vectors of skin-profiles, calculated via cosine similarity.
4. An honest account of what has and hasn't been validated, alongside a serverless ready implementation with authentication, persistent history, PDF reporting, and a profile-aware chatbot.

The rest of this paper is organized as follows: In Section II, a structured table of ten papers is provided in which related work is reviewed. The system architecture is given in Section III. The methodology is outlined in Section IV. The implementation is described in Section V. The verification results and the evaluation protocol are reported in Section VI. Section VIII concludes; Section VII discusses limitations.

Literature Review

The literature related to the work done can be categorized into three parts: deep learning for dermatological image analysis, face detection and efficient network design in the image pipeline and recommender systems for matching products to users. A selection of 10 representative papers in these groups were reviewed. They are summarized in Table I, and gaps that motivate our design are extracted from the discussion below.

Table 1 - Literature Review of Ten Related Papers

#	Reference	Year	Task	Method and data	Main finding	Gap relevant to this work
1	Esteva et al. [1]	2017	Skin cancer classification	Deep CNN trained on about 129,000 clinical images covering more than 2,000 diseases	Classification of skin cancer at a level comparable with the dermatologists tested	Targets malignancy, not everyday cosmetic conditions or skin type; no product guidance
2	Liu et al. [2]	2020	Differential diagnosis of skin disease	Deep learning system for 26 common skin conditions using multiple images and patient metadata	Accuracy exceeded that of generalist clinicians and approached that of dermatologists	Clinical, multi-image, metadata-heavy workflow; not a consumer single-photo service
3	Tschandl et al. [3]	2018	Dataset for pigmented lesions	HAM10000, 10,015 dermatoscopic images in seven categories	Provided a public benchmark that accelerated skin lesion research	Dermatoscopic lesion images differ from smartphone photographs of the face
4	Groh et al. [4]	2021	Fairness in dermatology AI	Fitzpatrick 17k, 16,577 clinical images labeled with Fitzpatrick skin type	Models perform best on skin types similar to those seen in training	Shows that skin tone diversity in training data must be audited for any consumer tool
5	Shen et al. [5]	2018	Facial acne diagnosis	CNN-based classification of facial acne from photographs	Automated facial acne assessment is feasible from ordinary images	Single condition; no skin type, no regional aggregation, no recommendation step
6	Zhang et al. [6]	2016	Face detection and alignment	Three-stage multitask cascaded CNN (MTCNN)	Accurate and fast joint detection and landmark localization	Gives face boxes only; skin regions must be derived by the application
7	Sandler et al. [7]	2018	Efficient CNN backbone	MobileNetV2 with inverted residuals and linear bottlenecks	Strong accuracy with low compute and memory cost	General-purpose backbone; needs fine-tuning on skin data to be useful
8	Burke [8]	2002	Hybrid recommender systems	Survey and experiments on weighted, switching, and cascade hybrids	Hybrids can offset the weaknesses of single techniques, such as cold start	Not specific to cosmetics or to ingredient knowledge
9	Koren et al. [9]	2009	Collaborative filtering	Matrix factorization on large rating data	Latent factor models give accurate personalized predictions at scale	Requires a history of user-item interactions that a new product does not have

#	Reference	Year	Task	Method and data	Main finding	Gap relevant to this work
10	Lops et al. [10]	2011	Content-based recommendation	Review of item-feature profiling and user-profile matching	Content features give transparent, explainable recommendations	Needs domain features; ingredient-to-condition knowledge must be curated

Deep learning for dermatological images

A single Deep Network trained on a huge body of clinical images could correctly classify skin cancer at a level similar to that of the dermatologists who participated in the comparison carried out by Esteva et al. [1]. Liu et al. [2] expanded this concept to a larger differential diagnosis comprising 26 common conditions and demonstrated the benefit of integrating images with patient metadata for enhanced accuracy. The community had a standard public dataset with pigmented lesions available to them with the HAM10000 dataset from Tschandl et al. [3]. These works show that CNNs can be used for skin images, but they deal with clinical diagnosis primarily in clinical or dermatoscopic images, focusing mostly on the diagnosis of lesions. The proposed system is for a different and lower-risk question – the cosmetic skin type and common facial conditions from a smartphone quality photograph – and deliberately refrains from medical diagnosis.

Every consumer system is confronted with the question raised by Groh et al. [4]. In their Fitzpatrick 17k study, networks did the best on skin types that were similar to the ones they were trained on – an apparently accurate model can be unfair on under-represented skin types. This is a design requirement for the proposed system: The training and evaluation data for the skin-type and condition models should be stratified by skin tone and results reported by skin tone. In a specific application to facial conditions, Shen et al. [5] demonstrated the ability of CNNs to classify facial acne pictures, thus proving the possibility of our condition detector, but with one condition only and without any downstream recommendation.

Face detection and efficient networks

The skin analyzer first needs to locate the face, and remove the background. The multitask cascaded network of Zhang et al. [6] is accurate and fast, and is widely packaged for Python, hence it is used as the main detector in the system. Less accurate on challenging poses, the Haar cascade approach of Viola and Jones [11] is available in OpenCV without any additional dependency and is employed as a fall-back mechanism. MobileNetV2 [7] is chosen as the backbone family in our design as it provides a good accuracy / cost performance ratio while a web service running on a modest or serverless hardware cannot afford a heavy model.

Recommender systems

Content-based methods are able to match up features in an item with a user profile and they are easy to explain [10]. Collaborative filtering is able to learn from the behavior of many users, and can find non-obvious matches, but it requires interaction history and is inappropriate for new users or products. Burke [8] studied hybrid designs and demonstrated that weaknesses of individual components can be overcome by the hybridization of the design. The proposed system is a hybrid, as a new platform has no interaction history, the content component provides the knowledge of the ingredients from the first day and the similarity component provides a profile-matching signal that can be later replaced by a true collaborative filtering signal as ratings are accumulated.

Research gap

The reviewed papers are divided into four parts: clinical diagnosis, single-condition detection, building blocks for the image pipeline, and general recommendation theory. None of them has a single consumer facing pipeline that (i) reads skin type from one photo, (ii) identifies specific cosmetic conditions on a face, (iii) filters products for declared allergens, and (iv) provides an explanation of the results via conversation. The system is intended to fill this gap, while acknowledging the need for further validation.

System Architecture

The solution is a three-tiered architecture with a browser-based client, a stateless REST API and a set of managed services/artificial intelligence.

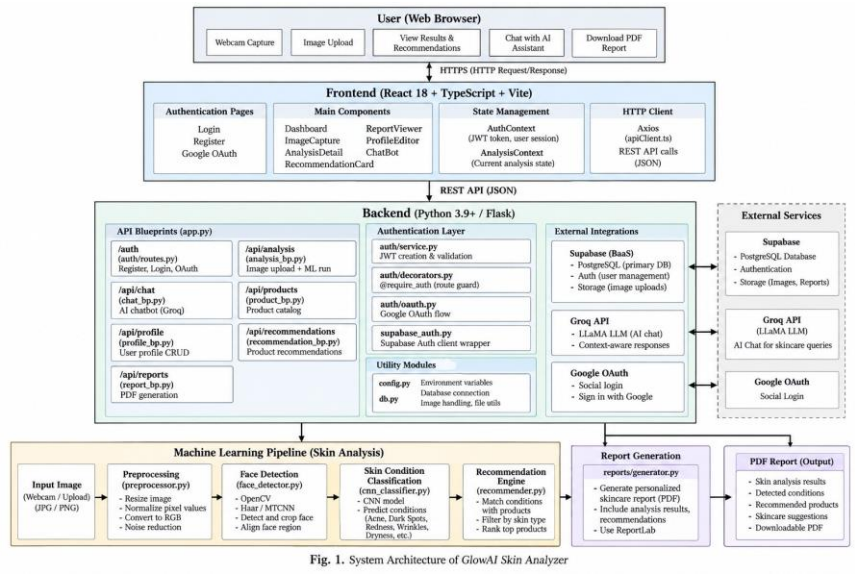


Fig. 1. System Architecture of *GlowAI Skin Analyzer*

Fig. 1 - High-level architecture of the proposed system.

Client tier

The front end is a single page application, implemented in React 18, TypeScript, and using Vite as the builder. It has three-step registration and login process, a dashboard with four tabs (analyze, history, profile, and recommendations), an image capture module (either upload a picture or take a photo with a webcam), a results view with a bounding box drawn around the conditions in the image, a recommendations card module, a report viewer, and a floating chatbot. The JSON Web Token (JWT) is attached to each request from an Axios client, while there are two React contexts to store the authentication state and history of the analysis.

Application tier

The Flask API is structured into blueprints for authentication, analysis, recommendations, profiles, products, reports and chat. Before any protected route is executed a decorator named require_auth validates the JWT and session that are stored in the session store. Tokens are not stored directly - there is a hash in the session table that contains an expiry time and an invalidation flag, so that the logout is effective immediately.

Data and service tier

Supabase offers user authentication, a PostgreSQL database featuring four tables (users, sessions, products, and analyses), and object storage for uploaded photos and created PDF reports. The AI pipeline is contained within the API process and loaded on-demand to ensure a rapid start-up. The chatbot calls the Groq inference API with the Llama 3.1 8B Instant model [13].

Request flow

Once the user uploads a photograph the API validates the file, saves it, and executes the four stages of the pipeline sequentially. Table II shows that each stage has a specific user-readable error that it can fail with. This is a type of error contract that the client can explicitly explain what it is that they need to be corrected on, such as a blurry photo that they need to retake.

Table 2 - Request Validation and Pipeline Error Contract

Stage	Check	Rejection condition	HTTP status
Upload	Format	Not JPEG, PNG, or WebP	400
Upload	Size	Larger than 100 MB	400
Upload	Resolution	Smaller than 224 × 224 pixels	400
Preprocess	Sharpness	Laplacian variance below 100	422
Detection	Face count	Zero faces	422
Detection	Face count	More than one face	422

Stage	Check	Rejection condition	HTTP status
Classification	Inference	Model raises an exception	500

Methodology

Image preprocessing and quality control

Unreliable skin analysis is the most common problem and the first stage will reject images that are not suitable for analysis rather than taking a guess. The image is then decoded to RGB, then converted to grayscale, and the Laplacian response variance is calculated:

$$S = Var\left(\nabla^2 I_{gray}\right) \quad (1)$$

If S is less than 100, then the image is rejected. Accepted images resize the images to 224 x 224 pixels using area interpolation and normalize the range from 0 to 1:

$$\hat{I}(x, y, c) = \frac{I(x,y,c)}{255} \quad (2)$$

This yields a float32 tensor with shape (224, 224, 3), corresponding to the shape of inputs fed to MobileNet-family backbones. The augmentation routine generates six augmentations of each image: original image, rotated images +15 and -15 degrees, horizontal flip, and brightness variations of ±20 percent. These changes are similar to the changes in head pose and lighting of consumer photos.

Face detection and regional segmentation

First the detector executes MTCNN [6]. If MTCNN does not detect any face, or is not available, the Haar cascade of Viola and Jones [11] in OpenCV is attempted, and then the request is denied. Only one face is needed as it would be no use if there were a skin profile of two people. Four distinct regions of the skin were obtained from the face box, defined by specifying fixed proportions of the origin (x, y), width (w), and height (h), as shown in Table III.

Table 3 - Proportional Facial Region Definitions

Region	Horizontal extent	Vertical extent
Forehead	Full face width	Top 25 percent of face height
Left cheek	Left 30 percent of width	30 to 65 percent of height
Right cheek	Right 30 percent of width	30 to 65 percent of height
Chin	Full face width	Bottom 20 percent of height

The boxes are easy to compute, and do not require any landmark model, but are only approximate for faces tilted around a strong angle or partially covered. A refinement based on landmarks is discussed in Section VII.

Skin type and condition classification

There are two heads on the classifier. The skin-type head is given the entire normalized face tensor and will produce a probability for each of five classes: oily, dry, combination, normal, and sensitive. The class with the greatest probability is chosen, and if this class is less than 0.60, it is tagged as low confidence; this will allow the interface to alert the user rather than give an ambiguous answer as if it were accurate.

The condition head identifies four conditions: acne, dark spots, enlarged pores and wrinkles. Regions of the four regions are cropped and resized independently to 224 × 224 and then scored. The aggregated score for condition c is the maximum value over regions r:

$$s_c = \max_{r \in R} p_c(r) \quad (3)$$

The probability of condition c in region r is denoted by p_c(r) and r ranges over the set R = {forehead, left cheek, right cheek, chin}.

A condition is reported if the s_c is equal or greater than 0.50 and the region with the highest value is displayed as the bounding box to the user. Here, max pooling would be suitable since if there is a problem with acne on the chin, the remaining parts of the face should not be affected. Conditions are reported according to confidence. The networks are loaded lazily from paths provided by environment variables, meaning that a model can easily be replaced without changing the code. The backbone family in the design is MobileNetV2 [7] backbone family, which is fine-tuned with skin-type and condition data.

The module additionally offers a mock inference path to generate correctly formatted output, which is determined by the image statistics and is returned every time trained weights are not set. This prevents the entire system from becoming untestable, but also implies that the outputs from the current prototype are not predictions (see Section VII).

Hybrid recommendation engine

The recommender is able to map the analysis and the user profile to the 10 best products among a catalog of products, each of which contains its ingredients, target skin types and target conditions. It goes in three stages.

Step 1: candidate selection and allergen filter. Products are chosen with reference to the detected product type, and if there are no products that match the detected product type, the entire catalog is used. Products with an allergen listed in the ingredients are withdrawn prior to scoring. This is a hard constraint, not a penalty, as recommending a known allergen is a safety failure, not just a bad match.

Step 2: content score. Based on its scientific links to each condition, an expert-defined map assigns the ingredient a score ranging from 0 to 1; salicylic acid gets a 1.0 for acne, while a 0.8 for enlarged pores, and a 1.0 for wrinkles for retinol. This content score is the confidence weighted average of the highest matching ingredient for each condition in the set D of detected conditions with confidence q_d:

$$C(p) = \frac{\sum_{d \in D} q_d \cdot \max_{i \in I(p)} m(i,d)}{\sum_{d \in D} q_d} \qquad (4)$$

The compatibility map m(i, d) and the ingredient set of p, where I(p) is the ingredient set of p, is used to determine compatibility. If no condition is found a neutral value of 0.5 is applied.

Step 3: profile-similarity score. Each product and the user is represented by a binary vector of length 9, 5 for skin type and 4 conditions. The similarity is the cosine of the angle between them:

$$F(p) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}_p}{\|\boldsymbol{u}\| \|\boldsymbol{v}_p\|} \qquad (5)$$

The final compatibility score is a weighted sum with a content weight of 0.7:

$$Score(p) = 0.7C(p) + 0.3F(p) \quad (6)$$

Products are sorted by this score and the top ten are returned with ranks and analyzed and stored with the result. The weights are biased towards the content score, since the latter should include the ingredient knowledge, while the similarity score is only based on labels. It is observed that F(p) is not the collaborative part in the strict sense of terms, that is learning from many users' ratings, but rather it is a profile-matching signal which will eventually be replaced by a true collaborative component [9] if the interaction data grows.

Profile-aware chatbot and report generation

The chatbot passes the latest ten messages to the Groq API, using the Llama 3.1 8B Instant model [13] following a system prompt that contains the user's skin type, main concern, known allergies, and conditions identified from the latest analysis. The prompt asks the assistant to remain succinct, provide advice relevant to the user's profile, suggest product categories rather than specific brands unless requested, never diagnose, and to recommend users to consult a dermatologist for more serious issues. The report is generated using ReportLab and a PDF is returned with the profile, analysis and recommendations on an A4 size, branded PDF and saved in object storage and made available for download.

Implementation

Technology stack

The technologies are listed in Table IV. All components are open source or free tiers, so the cost of this prototype is low.

Table 4 - Technology Stack

Layer	Technology
Front end	React 18, TypeScript, Vite, Axios, React Router
Back end	Python 3.12, Flask, Flask-CORS, Gunicorn
Authentication	Supabase Auth, PyJWT, bcrypt, Google OAuth 2.0
Database and storage	Supabase PostgreSQL and Storage
Computer vision	OpenCV, MTCNN, Pillow, NumPy
Deep learning	TensorFlow and Keras (model loaded from file)

Layer	Technology
Chatbot	Groq API, Llama 3.1 8B Instant
Reporting	ReportLab
Deployment	Vercel (front end and serverless Python API)

Data model and security

There are four tables in the database. The users table holds identity, contact, and skin-profile data such as skin type, main concern, skin tone, and a variety of known allergies as well as a soft-delete timestamp. The token hash, expiry and invalidation flag are stored in the session table. The products table contains name, brand, category, ingredient array, target skin types and target conditions; fifteen initial products from well known cosmetic brands are added to the table by a seed script. The analyses table is populated with the user who created the analysis, the image location, skin type, confidence, detected conditions and the ranked recommendations, which can then be displayed in the history view to track changes over time.

The security features include bcrypt password hashing, JWT (JSON Web Token) and a protected-route decorator on all data endpoints featuring invalidating sessions on the server, and a generic error message when the authentication is incorrect, without revealing which of the three items (email, password, or session) was incorrect. Images of a user's face are personal data and stored in access-controlled buckets that are linked to just the user.

Serverless deployment

The back end can be deployed as a Python serverless function on Vercel or as a Gunicorn process on a traditional server, and the front end as a static bundle. The notion of serverless execution relies on the database layer being connected to Supabase via a HTTP request rather than a persistent socket to the database. The deployment notes of the project show that server issues were debugged in the move to serverless hosting, and the repository has automated tests that check the deployment fixes as well as the recommendation logic.

Evaluation and Results

We keep some stuff that is now known apart from some that still needs to be measured, otherwise we would overestimate the contribution.

System-level verification

All the components implemented were tested against their specification. The upload endpoint returns the specified error code as listed in Table II if the image is not in a supported format, too large, or too small (less than 224 x 224), it does not have a face, or it has multiple faces.The upload endpoint rejects images that are not in a supported format, too large, too small (less than 224 x 224), do not contain a face, and contain multiple faces, returning the specified error code as listed in Table II. The recommender strips the allergen-containing products from the list before scoring, and returns at most of his 10 orders, which are sorted by the score in the interval [0, 1]. The mock path successfully generates the structural outputs of the classifier (five skin-type scores, up to four conditions with confidence values, and a low-confidence flag) which are successfully consumed by the recommender, database layer and report generator.

Worked example of the recommendation scoring

For example, if a user's analysis comes back with the conclusion that their skin is oily and they have acne (with a confidence of 0.82), their pores are enlarged (0.64), and they do not have an allergy, it is easy to see that their score is 0.82. We score two hypothetical products aimed at tackling oily skin. Product A treats oily or combination skin, acne and enlarged pores with salicylic acid. Product B includes niacinamide and addresses the problems of oily skin, dark spots and enlarged pores.

Table 5 - Worked Scoring Example (illustrative Products)

Quantity	Product A	Product B
Best ingredient match for acne	1.0	0.5
Best ingredient match for enlarged pores	0.8	0.8
Content score C by Eq. (4)	0.9123	0.6315
Similarity F by Eq. (5)	0.8660	0.6667
Final score by Eq. (6)	0.8984	0.6421
Rank	1	2

For Product A, the content score is (0.82 × 1.0 + 0.64 × 0.8) divided by (0.82 + 0.64), which is 1.332 divided by 1.46, or 0.9123. The user vector contains 3 active entries, oily, acne, pores and the product vector contains 4 active entries, oily, acne, pores, and one other, so that the cosine is 3 / (3*4^0.5) = 0.8660. The final score is 0.7 × 0.9123 + 0.3 × 0.8660 = 0.8984. The same steps for Product B yield 0.6421. It is user-friendly, because the salicylic acid product treats both pores and acne, and the results are easy to explain to the user based on matching ingredients.

Proposed evaluation protocol

The remaining task is mainly the evaluation of the trained models. We suggest the following protocol in order to provide a comparable and fair report of the results.

Data. A collection of public facial skin images and a locally gathered skin images set of which a consensus was reached, labeled with skin type and the four conditions, with Cohen's kappa as the measure of agreement. The data are partitioned per person (not per image), to avoid leakage.

Skin-type metrics. Confusion matrix (over five classes), macro F1, and calibration of the low-confidence threshold of 0.60.

Condition metrics. Precision, recall, F1, AUC at threshold 0.50 and localization accuracy of the reported region compared to reviewer annotation per condition.

Fairness. All measurements collected by Fitzpatrick skin type, as per Groh et al. [4] but the difference between the best and worst groups is reported.

Recommendation quality. We use a panel of reviewers to rate relevance for the top ten products for a set of test profiles, and report precision at 10 and the percentage of recommendations that contain an allergen (which should be 0%).

Robustness. Precision under controlled changes of lighting, pose and sharpness of images, to verify thresholds in Table II.

Latency. The end-to-end response time of the analysis endpoint when it is serverless and starts warm or cold.

Discussion and Limitations

No trained weights yet. The biggest drawback is the fact that the prototype currently follows the mock inference path. Architecture, interfaces, thresholds and downstream logic are complete and tested, but models not trained and loaded, so reported skin type and skin condition is not a true prediction. Any thing to say about the accuracy is premature, thus Section VI is not providing numbers, but rather a protocol.

Approximate regions. The fixed-proportion regions in Table III are inexpensive but are only approximate. A facial landmark model might generate regions in accordance with the actual anatomy of the face and be resistant to head turn, and might mask hair, eyebrow, and eyes which may be included in the crops.

Recommendation design. Similarity score F(p) is not a collaborative filtering technique, it is a measurement of similarity of category labels. The ingredient compatibility map is edited manually and applies to just fifteen ingredients, and the allergen filter is based on exact match on the ingredient name, a synonym or a compound ingredient name may not be matched. An ingredient vocabulary, e.g. INCI name, and fuzzy matching should be included. It is important to note that the catalog of 15 products is a demonstration catalog and not a market scale catalog.

Fairness and bias. As indicated in [4] the accuracy varies with skin color, and there have been shown to be variations in the performance of face detectors between demographic groups [12]. The system cannot be presented for general use until it has been stratified and the limits of the system should be stated on the interface.

Safety and privacy. The system is a cosmetic device, not a medical device. The chatbot is programmed not to diagnose and to always direct any severe concerns to a dermatologist, but a language model may still make a mistake, which is reiterated on the interface. Facial photographs are sensitive; deployments should provide deletion on request, keep such photos to a minimum, and ensure there is explicit consent. The maximum upload size of 100 MB is plenty large as well and ought to be decreased to safeguard the service.

Threats to validity. The photographs taken with various cameras and various lighting conditions (with and without make-up) will vary from the training distribution. The profile is based on self-reported skin type which may differ from the model and the system is not there yet to reconcile them.

Conclusion and Future Work

This paper introduced an end-to-end web platform that maps one facial photograph to a skin type, a set of localized skin conditions and a ranked list of cosmetics that are safe for the user's skin, along with a description in a PDF report and a chatbot to explain the result. The system is a staged pipeline system with explicit quality gates, a region-aware condition detector, a hybrid recommender system with explainable scores to users, and can be deployed on serverless infrastructure. A survey of 10 related papers revealed that each of these consumer pipelines have received individual study, but no one made any attempt to associate them with allergen safety, and provided a conversational explanation.

Future work will include training and testing skin-type and condition models according to the protocol outlined in Section VI, stratification by skin tone, replacement of the proportional regions by landmark-based segmentation, user-provided feedback and tracking of skin progress over time, and enforcing privacy via consent flows, retention limits and deletion tools, in order of priority.

References

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] Y. Liu *et al.*, "A deep learning system for differential diagnosis of skin diseases," *Nature Medicine*, vol. 26, no. 6, pp. 900–908, 2020.

[3] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, Art. no. 180161, 2018.

[4] M. Groh, C. Harris, L. Soenksen, F. Lau, R. Han, A. Kim, A. Koochek, and O. Badri, "Evaluating deep neural networks trained on clinical images in dermatology with the Fitzpatrick 17k dataset," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1820–1828.

[5] X. Shen *et al.*, "An automatic diagnosis method of facial acne vulgaris based on convolutional neural network," *Scientific Reports*, vol. 8, Art. no. 5839, 2018.

[6] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[8] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 73–105.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. I-511–I-518.

[12] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proc. Conf. Fairness, Accountability and Transparency (PMLR 81)*, 2018, pp. 77–91.

[13] A. Dubey *et al.*, "The Llama 3 herd of models," arXiv:2407.21783, 2024.