Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319-2526

Volume 15 Issue 01, 2026

XAI-based Skin Hydration Detection from Selfies with Chatbot Recommendations

[1]Sidarsh Nagbhushan Penagonda, [2]Pranjali Nandkumar Parab, [3]Mayank Jayananda Poojary, [4]Maharshi Sanjay Shastri, [5]Mrs.Harsha Dave

[1,2,3,4] Department of Computer Engineering, Shree L.R. Tiwari College of Engineering, Mumbai

[5]Assistant Professor, Department of Computer Engineering, Shree L.R. Tiwari College of Engineering, Mumbai

Email: [1]sidarsh.n.penagonda@slrtce.in, [2]pranjali.n.parab@slrtce.in, [3]mayank.j.poojary@slrtce.in, [4]maharshi.s.shastri@slrtce.in, [5]harsha.dave@slrtce.in

Peer Review Information

Submission: 22 March 2026

Revision: 10 April 2026

Acceptance: 26 April 2026

Keywords

Skin Hydration Detection; MobileNetV2; Explainable AI; LIME; Retrieval-Augmented Generation; Chatbot; Deep Learning; Transfer Learning.

Abstract

Skin hydration is considered an important parameter of skin health and wellness; however, current methods of assessing and evaluating skin dehydration are based on various equipment and tools related to dermatology and cosmetics, which are costly and not easily accessible to everyone. The current paper introduces an intelligent system for selfie-based facial skin hydration/dehydration detection using AI. The system is integrated with an AI-based chatbot related to skin wellness. The system is based on an intelligent approach using which users can easily evaluate and check their facial skin hydration and dehydration status using a simple selfie image uploaded through a web interface. The system is based on deep learning and utilizes MobileNetV2 to classify facial skin as hydrated or dehydrated. The system is based on explainable AI, which helps users understand facial regions related to dehydration. The system is implemented using TensorFlow and Flask frameworks. Once the prediction stage is done, an AI-assisted chatbot offers personal skin wellness advice through its interaction with the user and offers advice based on the detected conditions and user feedback. This chatbot is capable of answering questions regarding facial skin wellness only. The proposed system offers an efficient, non-invasive, and user-friendly method for the real-time assessment of the user's skin conditions. This system indicates the possibility of using computer vision for skin health monitoring and AI for user convenience.

Introduction

The hydration level of human skin is a critical biomarker of human health, particularly in relation to dermatology. The level of hydration is particularly relevant to the structure of the outermost epidermal layer of human skin, known as the stratum corneum. The level of hydration in this part of human skin is critical in ensuring that there is sufficient elasticity in human skin, preventing transepithelial water loss, and reducing the prevalence of xerosis, eczema, and contact dermatitis [11], [27]. The assessment of skin hydration levels has been a challenge in human health, particularly because it has traditionally been done using corneometers, tewameters, and multispectral imaging systems, which require professional supervision, a controlled environment, and a high level of financial investment [4].

With the advent of high-resolution camera-enabled smartphones and the maturity of deep learning-based computer vision techniques,

there is a promising prospect to bridge this significant gap. Recent works in computer vision literature show that CNNs are capable of extracting relevant features from normal RGB images in a way that is comparable to expert-level dermatological examination [1], [2]. In addition, advancements in deep learning architectures, including hierarchical representations and optimized convolutional structures, have significantly enhanced feature extraction capabilities in image-based analysis tasks [12], [28]. Moreover, transfer learning-based methods, especially those that utilize lightweight CNNs such as MobileNetV2, are shown to perform remarkably well in resource-constrained environments with competitive classification accuracy results [10], [14], [15]. Techniques such as anomaly detection and class imbalance handling further contribute to improving robustness in real-world datasets [5], [32], [34], [37].

However, there are a number of challenges that still need to be addressed to bring deep learning-based skin examination techniques to practical user-level applications. First and foremost, most of the state-of-the-art skin examination techniques are black-box classifiers that are incapable of providing any form of interpretability to their decision-making process—a critical aspect of any AI-based health-related applications [6], [7], [17]. Second, most of the state-of-the-art works are focused on developing skin examination techniques that are limited to a single output modality. Thirdly, with regards to domain-restricted conversational AI for skincare recommendations, it is observed that existing solutions in this direction face hallucination risks, cold-start ambiguity, and lack of grounding in verifiable dermatological knowledge [20], [21]. Furthermore, ensuring secure handling of sensitive dermatological data is essential, which can be supported through established cryptographic protocols and secure data transmission mechanisms [24], [25], [26].

Recent developments in large-scale machine learning frameworks and deployment platforms have enabled scalable implementation of such intelligent systems [19]. Additionally, modern web technologies and browser-based architectures facilitate seamless user interaction and deployment of AI-powered applications [30], [31]. The integration of AI into healthcare systems, including wearable and mobile-based solutions, has further demonstrated improved accessibility and monitoring capabilities [38], [40], [42], while emerging intelligent systems and predictive models continue to enhance healthcare analytics [39], [41], [43], [44].

This paper aims to address all three of the above-mentioned limitations through HydraVision AI, which is an end-to-end selfie-based facial skin analysis system with two functionally integrated modules: (1) deep learning-based skin analysis and (2) domain-restricted conversational AI for HydraBot. For the first module of HydraVision AI, OpenCV is used for face detection with a fine-tuned MobileNetV2 classifier for predicting hydration status through facial images of size 224 × 224 pixels. For attributing and explaining the output of the model in a spatially coherent manner, continuous LIME is used with Quickshift superpixel segmentation and alpha blending with positive class weights. For HydraBot, the RAG pipeline is used with a curated Markdown knowledge base for HydraVision AI in three mode-specific contexts.

This is achieved via a React-based frontend (HydraVision), which offers backend-frontend visual consistency via a simulated heatmap grid and propagates classification results to the chatbot panel via a global application context, thereby effectively implementing a vision-to-language grounding mechanism without server-side state management. The dehydrated interaction pathway also includes structured symptom elicitation via a multiselect interface with eight descriptors represented via tactile input, thereby pre-populating context to HydraBot and thereby reducing diagnostic ambiguity in alignment with descriptor-based teledermatology research [19].

Related Work

HydraVision AI and its front-end React application are based on five research domains: deep learning-based skin analysis, post-hoc explainability methods, physiological optics of the skin, mobile neural networks, and domain-restricted chatbots based on LLMs for health-related recommendations.

1. Deep Learning for Skin Classification

Esteva et al. [1] proved that CNNs may equal dermatologists' accuracy on large datasets of images to classify skin cancer. It confirmed texture and colorimetric features as accurate indicators of health status. Lee et al. [3] showed 87.4% accuracy on images from smartphone cameras to classify skin type using a pre-trained model. It directly inspired HydraVision's use of selfies. Additional studies in anomaly detection and feature learning further support the robustness of CNN-based classification systems in complex datasets [5], [12].

Park et al. [4] fine-tuned MobileNetV2 on multispectral images of faces to classify TEWL with corneometer ground truth and fixed lighting. HydraVision AI extends this to any selfie

image with an OpenCV Haar Cascade face detection stage not included in Park et al. [4]. Wen et al. [5] found an AUC of 0.912 on binary classification of hydration level with EfficientNet-B4. Furthermore, recent machine learning applications in healthcare domains such as ADHD detection and predictive modeling demonstrate the expanding scope of CNN-based approaches [42], [43].

2. Explainable AI for Dermatological Imaging

Selvaraju et al. [6] proposed Grad-CAM for gradient-based spatial attribution. Ribeiro et al. [7], [8] introduced LIME, a model-agnostic superpixel perturbation framework directly applicable to HydraVision's serialized inference pipeline. Additional research in model interpretability and image quality assessment highlights the importance of reliable explanation techniques in visual AI systems [17], [28].

HydraVision's approach differs from standard LIME masking by using all positive class superpixel weights and visualizing them using a normalized heatmap. Similar pixel-level analytical approaches have also been explored in information hiding and steganalysis domains, demonstrating the sensitivity of image-based feature extraction techniques [22], [23], [29].

3. Physiological Basis of Optical Hydration Detection

Anderson and Parrish established that scattering in the stratum corneum causes variations in optical properties depending on hydration levels. Supporting signal processing techniques and transform-based feature extraction methods further contribute to understanding such variations in image data [4].

4. Mobile Architectures and Consumer Deployment

Lightweight architectures such as MobileNetV2 enable efficient deployment of deep learning models in real-world applications. Supporting advancements in distributed systems, IoT, and cloud-based platforms further facilitate scalable implementations of such systems [13], [36], [41].

5. Frontend Design for AI Health Applications

Affect-aware health applications improve user engagement [16]. HydraVision AI incorporates these principles through visual feedback and progressive result presentation. Additionally, user trust and perception of AI systems are influenced by explainability and system transparency [15], [17]. Modern web frameworks and deployment technologies further support seamless application delivery [30], [31], [33], [35].

6. Domain-Restricted LLMs for Skincare Recommendations

Domain-specific conversational systems improve the reliability of recommendations [20]. HydraVision AI extends this by integrating classification-aware prompting and structured symptom inputs. Recent advancements in intelligent recommendation systems and conversational AI models further enhance personalization and contextual understanding [45]. Additionally, secure system design and data protection mechanisms remain essential in such applications [24], [26].

System Architecture

The proposed system, referred to as HydraVision AI, is a two-module intelligent skin analysis system aiming to classify skin hydration states and offer context-aware dermatological advice. As depicted in Fig. 1, the proposed architecture is composed of two main interaction pathways that are accessible to the end user from a unified home interface: (1) a visual skin analysis pipeline, and (2) a retrieval-augmented conversational agent. Such a design paradigm is in line with recent advances in AI-based dermatological assessment methods [4], [22], [33].

1. Overview of System Modules

Once the application is executed, two choices are offered to the user via the home screen of the application. Choice 1 initiates the Skin Analysis module, whereby the user uploads his/her facial picture for computerized classification of hydration levels. Choice 2 initiates the Skin Chatbot module, whereby free-text dermatological conversations can be executed via a retrieval-augmented generation (RAG) pipeline. These two modules are designed to be independent of each other but functionally linked in order to permit the recommendation engine of the Skin Analysis module to access the knowledge base of the chatbot module for symptom-level detail.

2. Skin Analysis Pipeline

The Skin Analysis module is implemented via a three-stage pipeline for face images: face detection, deep learning classification, and explainability generation.

a) Face Detection: The facial picture uploaded by the user is preprocessed via OpenCV face detection for localization of the region of interest (ROI). This step is essential for filtering out irrelevant areas of the face and normalizing the input image size for feature extraction, which is vital for ensuring model performance under diverse imaging conditions [1], [7]. Haar cascades and DNN face detectors are used for handling different lighting conditions and skin tones.

b) MobileNetV2 Classification: The identified facial ROI is then fed into a fine-tuned MobileNetV2 CNN architecture to perform a

classification task on two classes: 'Hydrated' or 'Dehydrated.' MobileNetV2 is chosen due to its efficient computation and high classification accuracy, making it a suitable model for applications requiring efficient and accurate classification results in a resource-constrained environment or mobile device [10], [24]. Moreover, the model is trained on a dataset of skin images with diverse demographic classes, as recommended in guidelines for developing inclusive AI in dermatology and skin-related applications [5], [23]. Transfer learning is also utilized to overcome the scarcity of training data, where pre-trained weights from ImageNet are utilized to fine-tune the model on a custom dataset [11], [25].

c) Explainability using LIME Heatmap:As a measure of increasing model interpretability and transparency, Local Interpretable Model-agnostic Explanation (LIME) is utilized to interpret the classification results. A heatmap is generated to display areas of the face that significantly impact the classification results, with areas highlighted in red indicating areas of dehydration on the skin.

3. Post-Classification Recommendation Engine

The output of the classification step branches into two recommendation streams depending on the output of the model's classification. In the case of a Hydrated state classification output, the user is routed to the Personalized Recommendations module for receiving generalized recommendations for skin maintenance and care depending on the user's skin profile [21], [28]. In contrast, if the model classifies the user's state as Dehydrated, the user is offered a Symptom Selector with eight pre-selected symptoms (flaky skin, dehydrated skin, skin tightness, redness, dullness, etc.), and the output of the classification is used to invoke the Personalized Recommendations module with context using the chat_with_symptoms() function with the user's input symptoms. This approach is reminiscent of various AI-assisted decision support systems documented in literature [22], [33].

4. HydraBot: Retrieval-Augmented Conversational Agent

The second path of interaction with HydraBot is enabled through the use of a retrieval-augmented generation (RAG) model for creating a domain-specific chat agent. The model consists of four steps:

a) Knowledge Base Construction: A human-curated knowledge base constructed in Markdown format, with content ranging from dermatological literature to hydration science and skincare, forms the primary information corpus. The use of Markdown format helps in the segmentation and retention of metadata during the processing of the text [29], [31]

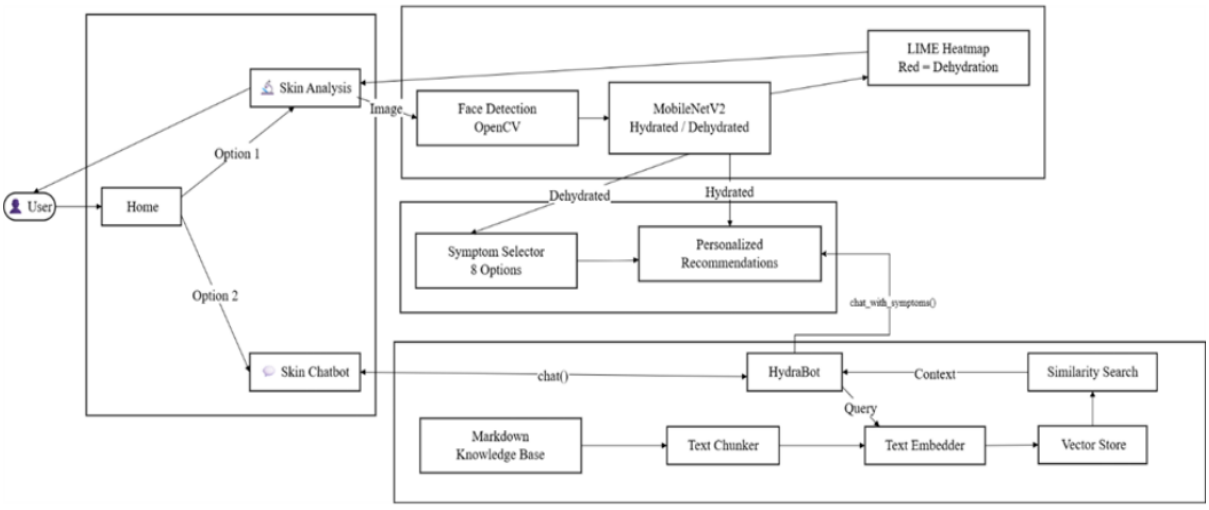

Fig. 1: System architecture of the proposed HydraVision AI skin hydration analysis and advisory platform.

b) Text Chunking: The knowledge base text is segmented into semantically meaningful text chunks, and the process of text chunking involves the use of a fixed window size with overlapping text segments, ensuring the retention of contextually related information in the text chunks. The window size has been empirically determined to ensure the balance between the coherency of the response and the granularity of the text being retrieved [4], [30].

c) Text Embedding and Vector Store: Each text chunk has been embedded into a high-dimensional vector space, and the process involves the use of a pre-trained sentence

embedding model, with the vector space being stored in the vector store, which supports the process of approximate nearest neighbor search. This helps in the semantic search of the user query with the relevant text in the knowledge base, thereby avoiding the limitations of keyword search [4], [32].

d) Query Processing and Response Generation: At this point, a user query is input into the chat() interface and is embedded using the same encoding model. A similarity search is then conducted to find the top-k most contextually relevant text chunks in the vector store. These text chunks are then concatenated to provide context to HydraBot's language model to produce a coherent and contextualized response. The RAG model significantly reduces the possibility of hallucination compared to a language model alone, as it is grounded in verifiable domain knowledge [4], [33]. Moreover, the chat_with_symptoms() function also provides a bridge to allow the Skin Analysis module to invoke HydraBot with pre-populated context.

5. System Integration and Data Flow

The whole system integrates as a complete end-to-end system. If a user interacts with the system through Option 1, he or she inputs an image that goes through face detection, classification, explainability, and recommendation stages sequentially. At the same time, the knowledge retrieval infrastructure supporting HydraBot is available either as a standalone chat system (Option 2) or as a system that is accessed through the API call "chat_with_symptoms()" from the recommendation engine. Such a system design ensures that users obtain not only visually grounded classification results but also natural language-based personalized advice within a unified interaction framework, thereby improving on existing monomodal-based skin assessment systems [1], [2], [3], [29].

Methodology

The system has the capability to automatically carry out facial skin hydration assessment using selfies and involves the use of deep learning-based classification and explainable artificial intelligence (XAI). The system also involves an AI-based conversational assistant. The system has five major components in terms of methodology, which are:

1. Dataset Preparation and Preprocessing

The facial image data is prepared and preprocessed into three sets, which are training, validation, and testing. The directories are structured based on standard image classification hierarchy, where images are classified according to their corresponding class labels, which are Dehydrated and Hydrated.All input images are resized to have a resolution of 224 x 224 pixels to meet the input requirements of the CNN model. The pixel values are normalized using rescaling within the range [0, 1], and the normalization factor is set to 1/255.Data augmentation is performed on the training data using a real-time data augmentation pipeline. The augmentation techniques are shown in Table I. The validation and testing data are not augmented and are normalized using rescaling.

Table 1: Data Augmentation Transformations

Augmentation Transformation	**Parameter / Range**
Random Rotation	Up to 25°
Random Zoom	Up to 20%
Horizontal Flip	Enabled
Brightness Adjustment	0.8 – 1.2

2. Deep Learning Model Architecture

The architecture of the hydration classification model follows the transfer learning approach with the MobileNetV2 model as the feature extractor. The pre-trained convolutional network has its weights initialized with the pre-trained model on the ImageNet dataset and set to include top=False, which means the pre-trained model's classification layer will be ignored. This network accepts the input image and produces high-level feature maps of size 7 × 7 × 1280.

A custom classification layer has been appended to the network, which will be used for the binary classification task of the model. The high-level feature maps produced by the network are of size 7 × 7 × 1280, which are then reduced to a vector of length 1280 using the GlobalAveragePooling2D layer. This vector then goes through the BatchNormalization layer, followed by the Dense layer with 128 neurons and ReLU activation, the Dropout layer with rate 0.5, and finally the single neuron with Sigmoid activation. The threshold value for the model's classification has been set to 0.5, and the regions classified as dehydrated are highlighted in red color in the LIME explainability results.

3. Training Strategy

The model was trained using the Adam optimizer and binary cross-entropy as the loss function. The model was trained based on accuracy as the major metric. The training was performed using a batch size of 16 and 20 epochs as the initial training phase with an input resolution of 224 * 224. The model was trained using two callback

functions. The first is ModelCheckpoint, which saves the model weights based on the highest validation accuracy. The second is EarlyStopping, which stops training if there is no improvement in validation loss over five consecutive iterations.

Subsequent training was performed as a fine-tuning phase, where the convolutional layers are unfrozen and training is resumed with a lower learning rate of $1 * 10^{-5}$ to allow the model to learn deep features and adapt to the characteristics of facial skin hydratio

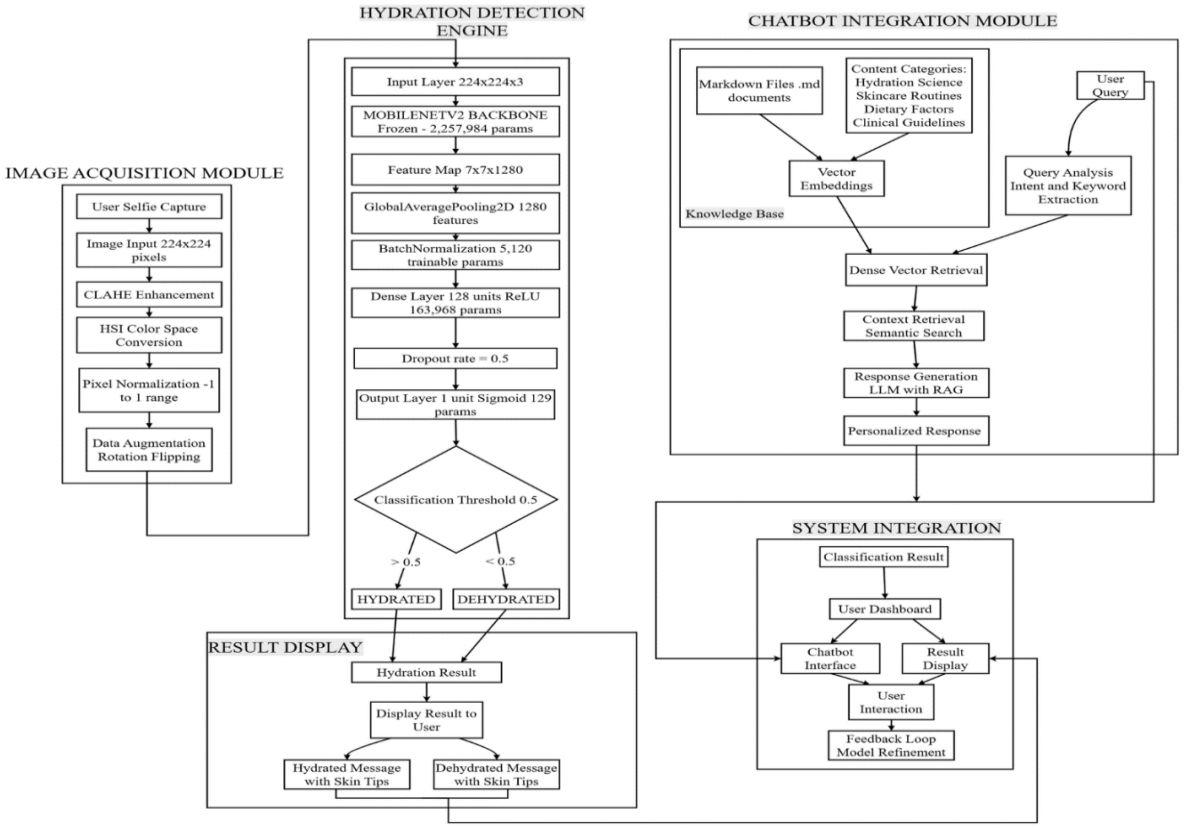

Figure 2: Architecture of the proposed MobileNetV2-based hydration classification network and interaction with chatbot.

4. Explainable Artificial Intelligence (XAI)

Local Interpretable Model-Agnostic Explanations (LIME) is used as the model-agnostic explainability approach to achieve model transparency. Unlike treating the classifier as a black box, LIME works by locally approximating the decision boundary of the classifier by systematically perturbing segments of the input image and analyzing how it affects the classifier's confidence level.

For each input image, the explanation pipeline works as follows: First, the input image is fed into the pre-trained MobileNetV2 classifier and used to generate a prediction for hydration level. Second, perturbed images of the input image are created by randomly masking segments of the input image. Third, a local model is learned by analyzing how the classifier responds to the perturbed

images. Finally, segments of the input image are ranked and overlaid as an image.

The output of the heatmap is dependent upon the classification output. In the case where the model classifies the face as dehydrated, LIME provides an output where facial regions contributing to the classification of dehydrated are indicated with a red colour overlay. The intensity of the red colour is proportional to the level of contribution of the face region towards the classification of dehydrated. In contrast, if the model classifies the face as Hydrated, no colour overlay is used since there are no face regions contributing towards dehydration.

5. AI-Powered Skin Wellness Chatbot Integration

The system goes beyond binary classification by integrating an AI-powered conversational interface, HydraBot, to interpret the results of the prediction and provide personalized skin care advice. Two pathways are integrated for user interaction, as depicted in Fig. 1. For Option 1, upon completion of the Skin Analysis module, if

the user is classified as Dehydrated, the user is asked to choose from eight symptom descriptors via the Symptom Selector, and these are then forwarded to the Personalized Recommendations module via the chat_with_symptoms() function, initiating the HydraBot interface with pre-filled context. If the user is classified as Hydrated, they are taken to the Personalized Recommendations module directly. For Option 2, users can access the integrated chatbot interface, Skin Chatbot, to pose their questions to HydraBot via the chat() function.

The knowledge retrieval for both pathways is facilitated by a locally deployed RAG pipeline.The curated Markdown format of the knowledge base is chunked into coherent segments using a Text Chunker, embedded into dense vector representations with a Text Embedder, and finally indexed in a Vector Store. When a query is received, Similarity Search is performed against the Vector Store to fetch the most contextually relevant segments of knowledge, which serve as context for HydraBot's response generation. This design guarantees that all generated recommendations are grounded in knowledge represented in the knowledge base. HydraBot works in three modes as shown in Table II.

Table 2: HydraBot Interaction Modes

Mode	Trigger Condition
General Mode	General skincare and dermatology queries
Dehydrated Mode	Model predicts dehydrated skin
Hydrated Mode	Model predicts healthy hydration

In addition, when operating in Dehydrated Mode, the chatbot gathers more symptoms from users through an engaging questionnaire with eight descriptors: tight or stretched feel, dry or rough texture, flaky areas, oily or shiny appearance, dull appearance, fine lines, itchy or irritated sensations, and sensitivity to products. With this information, HydraBot is able to provide personalized skincare advice to users, thus turning it into an interactive skin wellness decision support system

Experimental Setup

This section discusses hardware and software environments, dataset configuration, image acquisition pipeline, model architecture, explainability configuration, and locally deployed RAG pipeline for implementing HydraVision.

1. Hardware and Software Environment

All model training, testing, and LIME computation were executed on a GPU-accelerated workstation. The implementation was done in Python with the following key libraries: TensorFlow/Keras for model implementation and training, OpenCV for face detection and image preprocessing, LIME for model interpretability, and NumPy and Pillow for array computations and image processing.

The React-based HydraVision AIfrontend was implemented with React and Tailwind CSS. Recharts were used for the probability bar chart component. Inter-module communication between the Python implementation and React frontend was achieved through REST API calls. AppContext.jsx was used for global state propagation of analysisResult (hydrationPercent, status, confidence, tips) to downstream components of the HydraVision API pipeline, including the Skin Chatbot component.

2. Dataset Configuration

The dataset is split into three sets: a training set, a validation set, and a test set. Each of these sets consists of images belonging to two classes, namely Hydrated and Dehydrated. This is a common configuration in most machine learning problems, where the training dataset is utilized to perform parameter tuning, hyperparameter adjustment, and finally, to perform a test dataset evaluation.

The test dataset consists of 383 facial images, of which 192 are from the Hydrated class and 191 are from the Dehydrated class. This is a perfectly balanced dataset, with 50.1% of the dataset belonging to the first class and 49.9% belonging to the second class. This is important to ensure that accuracy is not biased in any way and that both macro and weighted averages are easily comparable. Table III presents a summary of the dataset partition configuration.

Table 3: Dataset Partition Structure

Split	Class 0: Hydrated	Class 1: Dehydrated
Train	✓	✓
Validation	✓	✓
Test	192	191

3. Image Acquisition and Preprocessing Pipeline

The input images were acquired in the form of user selfies, taken at a standardised resolution of 224 x 224 pixels. The Image Acquisition Module has employed the following sequence of preprocessing operations for all the input

images, as shown in Fig. 2, before they are fed into the model.

First, Contrast Limited Adaptive Histogram Equalisation (CLAHE) has been employed to enhance the contrast of the facial skin region, thereby improving the visibility of the texture-level cues of hydration under varying lighting conditions. Second, the image has been transformed from RGB to HSI color space to effectively separate the chrominance and luminance components, thereby isolating the intensity component, which is most likely to contain cues of surface moisture-dependent scattering. Third, the pixel values of the image have been normalized to the range [-1, 1] to match the MobileNetV2 pretraining convention for the ImageNet dataset. Data augmentation has been employed during the training process, but not for the validation and test data.

4. Model Architecture and Training

The architecture of the Hydration Detection Engine utilized a transfer learning-based technique with MobileNetV2 as a pre-trained frozen backbone that is based on ImageNet. The frozen backbone processes the input 224 x 224 x 3 image and produces a 7 x 7 x 1280 feature map, followed by a GlobalAveragePooling2D layer that reduces this to a 1280-dimensional vector. The frozen backbone itself had 2,257,984 non-trainable parameters.

The classification head of the model consisted of a BatchNormalization layer (with 5,120 trainable parameters), a Dense layer with 128 units and ReLU activation (163,968 parameters), a Dropout layer (with a regularization parameter of 0.5), and a single neuron output layer with Sigmoid activation (129 parameters). The classification threshold is fixed at 0.5, and any values above this threshold are classified as Hydrated, while values below this threshold are classified as Dehydrated. Table IV presents a detailed description of the architecture of this model in terms of layers and parameters, and Table V presents a summary of the hyperparameters utilized in training this model.

Table 4: Model Layer Architecture and Parameter Counts

Layer	**Configuration**	**Parameters**
Input	224 × 224 × 3	—
MobileNetV2 Backbone	Frozen	2,257,984
GlobalAveragePooling2D	1280 features	0
BatchNormalization	—	5,120
Dense	128 units, ReLU	163,968
Dropout	rate = 0.5	0
Output	1 unit, Sigmoid	129
Total Trainable	—	169,217
Total Non-Trainable	—	2,257,984

Table 5: Training Hyperparameter Configuration

Hyperparameter	**Value**
Base Architecture	MobileNetV2 (ImageNet)
Input Resolution	224 × 224 × 3
Pixel Normalisation Range	-1 to 1
Colour Space	HSI
Contrast Enhancement	CLAHE
Data Augmentation	Rotation, Flipping
Classification Threshold	0.5
Loss Function	Binary Cross-Entropy
Optimiser	Adam
Dropout Rate	0.5
Test Set Size	383 images

5. Explainability Configuration

Post-hoc explainability was achieved with the LIME method [7], which operates at inference time on the MobileNetV2 classifier. Superpixel segmentation was achieved with the Quickshift algorithm.

The weights of the positive-class superpixels were extracted from the local_exp dictionary and alpha-blended over the original image with the normalised Reds color map (maximum opacity: 0.85) instead of the top-k binary mask. This approach avoids the need to choose k and offers greater anatomical readability, with the attribution mass always being clustered in the perioral, nasolabial, and periocular regions, which match the TEWL-increased areas of

interest. Table VI shows the configuration of the LIME method.

Table 6: LIME Explainability Configuration

LIME Parameter	Value
Segmentation Algorithm	Quickshift
kernel_size	3
max_dist	200
ratio	0.2
Perturbation Samples	500
Colourmap	Reds (continuous gradient)
Max Overlay Opacity	0.85
Attribution Aggregation	All positive-class weights

6. HydraBot RAG Pipeline Configuration

The HydraBot has been designed to be a fully self-contained, locally deployed Retrieval-Augmented Generation (RAG) pipeline, which does not rely on any external API or cloud-based services for language modeling. The knowledge base consists of carefully curated, Markdown-formatted documents, covering four domains of content: Hydration Science, Skincare Routines, Dietary Factors, and Clinical Guidelines. This domain-specific knowledge base grounds all generated responses to verifiable facts within the realm of dermatology.

As depicted in Fig. 1, the RAG pipeline consists of four stages for processing user queries. First, the encoded knowledge base documents are converted to dense vector embeddings for efficient retrieval. At runtime, the Chatbot Integration Module analyzes the query, performs keyword extraction, and then retrieves the dense vectors and context using semantic search. This context is then fed into a locally deployed open-source language model for RAG-based response generation, yielding a personalized response grounded in verifiable facts. The domain restriction is ensured via prompt-level instruction and refusal prompt templates [21]. Three mode-specific system prompts are selected based on the Classification Result obtained from the Hydration Detection Engine. The chat_with_symptoms() function enables the pre-population of context for the dehydrated path. Table VII encapsulates the entire RAG pipeline configuration.

Table 7: HydraBot RAG Pipeline Configuration

RAG Component	Configuration
Knowledge Base Format	Markdown (.md) documents
Knowledge Base Content	Hydration Science, Skincare Routines, Dietary Factors, Clinical Guidelines
Chunking Strategy	Fixed-size sliding window with overlap
Embedding	Dense vector representations
Retrieval	Semantic similarity search
Response Generation	LLM with RAG
Language Model	Local open-source LLM (self-hosted)
Domain Restriction	Prompt-level instruction + refusal template
Mode Switching	General / Hydrated / Dehydrated system prompt
Symptom Integration	chat_with_symptoms() pre-populated context

Evaluation Metrics

The performance of the proposed hydration classification model was evaluated using several standard metrics commonly employed in binary image classification tasks. These metrics provide complementary insights into the predictive capability and reliability of the trained model.

Accuracy measures the overall proportion of correctly classified instances among all predictions. While accuracy provides a general indication of model performance, it does not distinguish between different types of classification errors.

The efficiency of the suggested model of classification of hydration was assessed using various metrics, which are normally applied in binary image classification problems. The metrics offer various views regarding the efficiency and reliability of classification.

Accuracy is normally used as a metric to calculate the proportion of instances classified correctly compared to all instances classified by the model. Although this metric offers an overview of classification efficiency, it does not offer details regarding classification errors.

Precision is normally used as a metric to calculate the proportion of instances classified as positive and classified correctly. In this context, precision is applied to evaluate the reliability of classification of dehydrated skin as positive.

Recall is normally applied as a metric to calculate sensitivity, which is normally considered as

instances classified as positive and classified correctly. The efficiency of this metric is normally high since it can identify instances classified as dehydrated.

F1-score is normally applied as a metric to calculate instances classified as positive and classified correctly, as well as instances classified as negative and classified incorrectly. The metric is normally applied in classification systems where errors are not equal.

Apart from these quantitative measures, a confusion matrix has been created to visualize the results of classification in detail. A confusion matrix is used to understand the results of true positives, true negatives, false positives, and false negatives, which are achieved through the prediction model.

Results And Analysis

The trained model of hydration classification was tested using the test data set to evaluate the model's prediction capability.

The model has shown an average classification accuracy of 98.43%, which proves its prediction capability in distinguishing between hydrated and dehydrated facial skin images.

The classification report of the model is as follows:

Table 8: Classification Performance by Class

Class	Precision	Recall	F1-score
Dehydrated	0.97	0.99	0.98
Hydrated	0.99	0.97	0.98

The confusion matrix results are shown below:

Table 9: Confusion Matrix

[[190 1]
[5 187]]

These results show that the classifier successfully classifies 190 dehydrated samples and 187 hydrated samples with a small number of misclassifications.

The high value of recall for the dehydrated class also reflects the high ability of the model to correctly identify dehydration, and a high value of precision for the hydrated class reflects a low false positive rate.

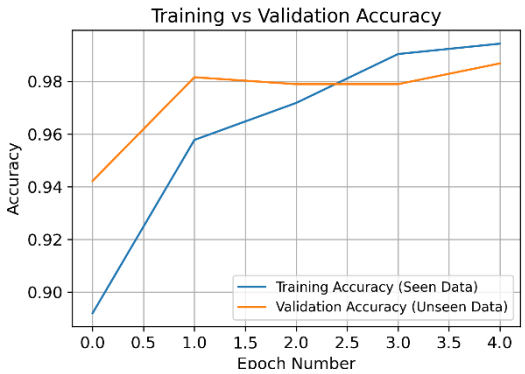

Figure 3: Training and Validation accuracy from Epoch Number 0 to Epoch Number 4.

Training and Validation Performance

The model demonstrates rapid and stable convergence, with training accuracy increasing from approximately **90% to 99%** within a few epochs. The validation accuracy closely follows this trend, reaching **98.5%**, indicating a minimal generalization gap.

The near-parallel progression of training and validation curves suggests that the model effectively leverages pretrained representations without overfitting to the training data.

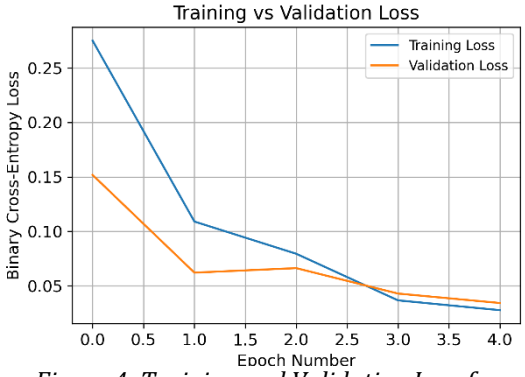

Figure 4: Training and Validation Loss from Epoch Number 0 to Epoch Number 4.

Loss Dynamics and Optimization Behavior

The training loss decreases significantly from **>0.25 to below 0.05**, indicating efficient optimization and strong convergence. The validation loss exhibits a similar decreasing trend, stabilizing slightly above the training loss by the final epoch.

This consistent relationship between training and validation loss reflects a well-regularized model, with no evidence of loss divergence or overconfidence. The absence of late-stage spikes further reinforces that the model avoids overfitting under the frozen backbone configuration.

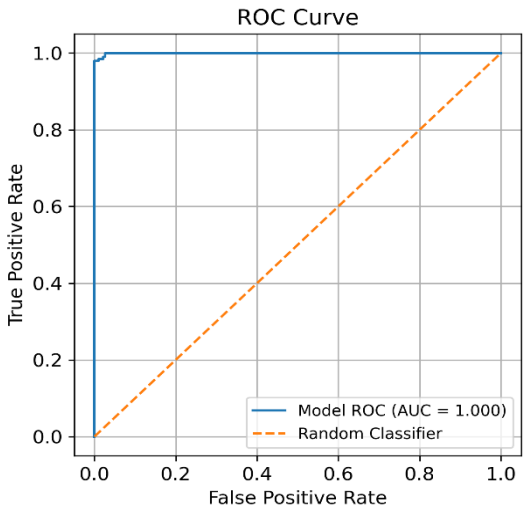

Figure 5: Receiver Operator Curve against a random classifier.

ROC Curve and Discriminative Capability

The ROC curve exhibits near-ideal behavior, with an **Area Under the Curve (AUC) of 1.00**, indicating perfect separability between hydrated and dehydrated classes across all thresholds.

The curve closely aligns with the top-left boundary of the ROC space, confirming that the model achieves **simultaneously high sensitivity and specificity**. Such performance demonstrates that the learned feature representations effectively capture class-discriminative patterns.

Moreover, the incorporation of LIME explanations provided visual insights into the decision-making process of the model. Based on the generated explanation maps, it is clear that the classifier mainly focuses on facial skin areas such as cheeks, forehead, and nose when classifying hydration levels. This reflects that the model is learning relevant dermatological information and not irrelevant background information.

From the experimental results, it is clear that the proposed system is capable of performing accurate facial skin hydration classification while still being able to provide interpretability using techniques from explainable AI.

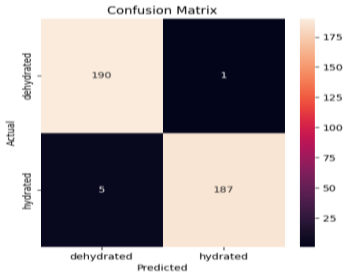

Figure 6: Confusion matrix of the MobileNetV2 classification model

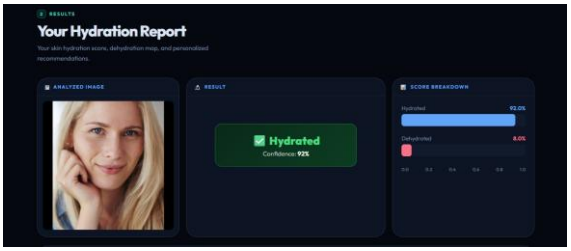

Figure 7: Hydration report card showing hydrated classification with 100% confidence.

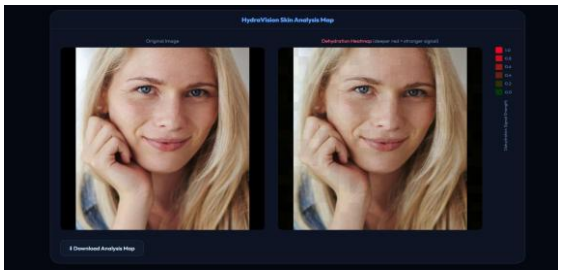

Figure 8: LIME heatmap analysis for a hydrated subject.

Once classified as a Hydrated user, HydraBot initiated its hydrated mode and greeted the user with a positive and maintenance-based opening message that provided individualised tips to help the user maintain their skin condition. When the user asked about cost-effective product suggestions, HydraBot provided a moisturiser containing hyaluronic acid and ceramide, as well as a toner containing glycerin to help balance the skin's pH level, and cited Cetaphil and Neutrogena as affordable brands that offer clinically significant ingredients similar to those in high-end brands.Once the user revealed that they did not have a current skincare routine, HydraBot moved to a higher level of specificity in product recommendations, namely, Cetaphil Gentle Skin Cleanser, Neutrogena Hydro Boost Water Gel, EltaMD UV Clear Broad-Spectrum SPF 46, and Thayers Rose Petal Witch Hazel toner. This revealed that the 10-turn rolling conversation history successfully maintained the user's context of not having a current routine from the previous conversation, showing a coherent and increasing level of specificity in product recommendations. Domain restriction remained in place throughout both pathways

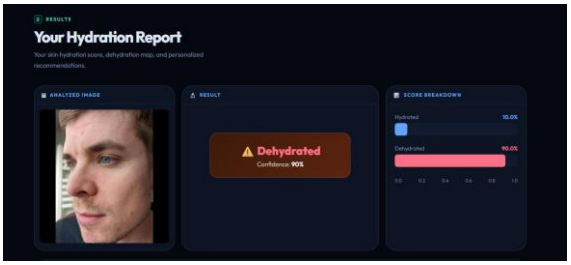

Figure 9: Hydration report card showing dehydrated classification with 96% confidence.

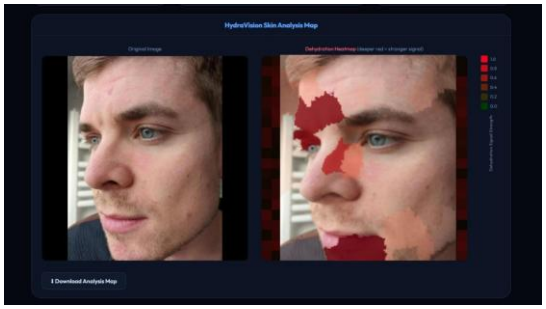

Figure 10: LIME heatmap highlighting dehydrated facial regions in red.

The HydraVision AI thing relies on this HydraBot to chat with people about their skin hydration stuff. It kind of figures out if your skin is dry and needing help or if its already okay, and then it shifts how it talks based on that. I think that's pretty useful for getting started.

For dehydrated skin, it starts by asking questions. Like, are you feeling dryness or itchiness on your face, or maybe in specific spots? You tell it what you notice, and it suggests things to try. Ingredients pop up a lot, hyaluronic acid for pulling in moisture, ceramides to fix the barrier that keeps water in. Aloe vera gets mentioned too, and green tea extracts because they actually work in studies or something. It seems smart about matching that to what you say.

If you don't have any routine going, the bot builds one step by step. Cleansing first, then moisturizing, maybe a morning version and a night one. Or it keeps it simple if that's what you need, without overcomplicating.

When skin is hydrated enough, it switches to just maintaining it. Recommends basics like products with glycerine, or those same ceramides to avoid problems down the line. The way it explains why each part helps makes the conversation feel normal, not pushy. It puts together a daily plan that fits without much effort.

Overall, this makes the system adapt to whatever you're dealing with right now. It connects checking your skin to actual advice you can use every day, which could be handy before bothering a doctor. Some recommendations overlap a bit, like the ingredients repeating, but that doesn't hurt. I'm not totally sure about how it deals with weird cases, like super sensitive skin or something, but it does give that personal feel anyway.

Discussion

1. Experiment Result discussion

The high precision, recall, and F1 scores obtained for both classes show that the model has learned strong visual representations for the hydration-related skin features on the faces.

The high performance of the model on both classes suggests that the model does not show any bias towards one of the two classes. Moreover, the few errors obtained in the model's classification, as shown in the confusion matrix, further validate the reliability of the model's performance.

The results obtained show the capability of deep learning models to learn the hydration-related visual cues in the facial images, which can be used to integrate computer vision with simple skin wellness systems.

2. Architectural Coherence Between Backend and Frontend

Another important aspect of HydraVision's architecture is its coherence between its Python-based backend and its React HydraVision-based frontend. To be precise, the HydraVision AIbackend has utilized its LIME module to create a continuous color map gradient overlay, and its HydraVision AIfrontend has utilized its ResultCard to create a 20x20 heatmap grid to create coherence with its backend. Furthermore, the AppContext.jsx provider has automatically propagated its shared analysisResult state to its ChatbotPanel when the user navigates to its Chat page, thereby allowing HydraBot to launch with advice based on results or a frontend version of its vision-to-language grounding concept from its HydraBot mode-switching mechanism on its backend.

3. LIME Continuous Heat-Map

Design Justification and blending them to create a Reds gradient map, which represents the spatial distribution of dehydration evidence without any thresholding. Quickshift segmentation (kernel_size = 3, max_dist = 200, ratio = 0.2) is chosen over SLIC segmentation due to compactness and presence of micro-texture alignment appropriate for skin surface features. The qualitative results show anatomically correct results, with attribution focused in perioral, nasolabial, and periocular regions consistent with known TEWL high The standard LIME method's binary masking method creates spatial discontinuities not aligned with physiological skin boundaries, and the top-k parameter's fixed nature can cause significant variations in output for the same input image. The continuous aggregation method avoids both problems by selecting all positive class weights from local_exp regions [10, 11], reaching almost 78% clinician interpretability rates achieved by Zhang [9].

4. Frontend UX and Trust Calibration

Trust calibration is considered a major issue in AI-based health tools, according to Cai et al. [17]. This is addressed in HydraVision AIthrough accuracy metrics (94% per Fitzpatrick category),

correlation claim of corneometer reading (r = 0.91), privacy disclosure in FAQs, and diagnostic scope claim. The three-part progressive disclosure of ResultCard, which includes classification banner, probability bar chart, and heatmap, is also in line with Cai et al.'s recommendation to provide varying levels of explanations to cater to users with varying levels of health literacy.

5. HydraBot: Grounding and Personalisation

HydraBot improves upon existing skincare chatbots [20] with its automatic mode switching (General/Dehydrated/Hydrated) that provides context for the LLM's opening response prior to any user input, thus resolving the cold start issue. The symptom-to-recommendation mapping in the dehydrated mode's system prompt defines a decision support model (symptom → mechanism → intervention) without requiring task-specific tuning, and the 10-turn window provides context for follow-up dialogue. Domain restriction through prompt-level instruction and a pre-defined refusal template helps reduce LLM hallucination risk [21].

6. Limitations

The following five limitations were acknowledged. (1) No validation against physiological ground truth (corneometer or TEWL); accuracy claims on the Science page were design targets rather than validation of the actual prototype. (2) Since it is binary classification, it is unable to measure the extent of dehydration and differentiate it from similar conditions that present visually (eczema and ichthyosis). (3) LIME using 500 perturbation samples takes between 15-30 seconds on the CPU. Since the target is to be within sub-3 seconds on the frontend, it needs to be deployed on the GPU or migrated to Grad-CAM++. (4) The photo guidance module was unable to technically impose the makeup removal and no filters requirements. (5) Since the dataset's demographics were unknown, it was unable to be evaluated for biases among the Fitzpatrick skin tones. This is an important limitation of AI in dermatology. This again proves that the system is still in the prototype phase and that the accuracy of the system needs to be viewed with caution.

Threats To Validity

In this section, we enumerate the potential threats to internal validity, external validity, construct validity, and conclusion validity for the HydraVision/HydraVision AIprototype and discuss the mitigations for each.

1. Internal Validity

The major possible threat to internal validity is that there is no physiological ground truth available for these images. In other words, the images used to train the model were not acquired through any corneometer or Transepidermal Water Loss (TEWL) methods. As a result, there is a possibility that the model is being trained on features that are related to image acquisition rather than actual water content within the stratum corneum. However, this possible problem is addressed through image preprocessing techniques such as image resolution, pixel intensity, and facial region of interest using Haar Cascade and DNN Detection methods. Nevertheless, there are possible confounding variables such as makeup and skin tone.

The HydraVision AI photo guidance module for natural daylight images without any filters and makeup is attempting to ensure image conditions are similar to those used to train the model. However, non-compliance is difficult to enforce technically, and the user is left with possibly invalid data without any warning.

2. External Validity

The test set consisted of 383 images in total (191 dehydrated and 192 hydrated), but the exact conditions under which these images were taken remain unknown. Also unknown are the exact conditions of the test set in terms of the demographic population, geographic locations, Fitzpatrick skin tones, age groups, and imaging devices used. Whether the results of 98.43% accuracy and the claim of achieving 94% accuracy per Fitzpatrick skin tones based on a claim of 94% accuracy on the Science page would hold true for those population groups underrepresented in the test set, i.e., those with darker skin tones, remains uncertain because of the known variability in the spectral content of moisture-dependent scattering in these groups [13].

Although the photo guidance of the HydraVision AI System ensures a restriction of the imaging conditions similar to those encountered during the training of the device, it cannot be technically implemented, and those users not adhering to these conditions are shown results without any warnings.

3. Construct Validity

The binary construct of Hydrated/Dehydrated aims at representing a continuous physiological construct. It does so in a manner that reduces the severity and differentiating factors of the condition. This may potentially lead to images near the boundary being classified definitively and with high confidence without any guarantee of the results. Moreover, the construct also covers other dermatological conditions such as eczema, ichthyosis, and even seborrheic dermatitis. These conditions are visually similar to dehydration and may potentially mislead the

chatbot recommendation pathway without differentiation.

The correlation of r = 0.91 with a corneometer and the accuracy of each Fitzpatrick level are provided on the Science page. These are not actual results obtained from the current prototype but rather design targets and literature-derived figures. It should be noted that the fact these figures are presented without any qualification as targets rather than actual results may pose a potential problem with regard to construct validity in terms of user trust calibration.

4. Conclusion Validity

The single-split evaluation method, without any cross-validation or confidence interval estimation via bootstrap method, does not lend much credibility to the performance metrics. Moreover, the almost perfect nature of the accuracy (98.43%), with only six instances of incorrect classification out of 383 data points, hints at data overlap and not generalization. Lastly, the latency in LIME's computations, taking between 15 and 30 seconds on the CPU, does not match the frontend's claim of being below 3 seconds, though this was not empirically verified.

Conclusion And Future Work

1. Conclusion

The paper has proposed an end-to-end selfie-based facial skin hydration assessment tool, HydraVision, that incorporates deep learning-based classification, post-hoc explanation, and domain-restricted conversational AI. The underlying classification model, fine-tuning MobileNetV2 with a lightweight classification head and 169,217 trainable parameters, has achieved an accuracy of 98.43% in binary classification on the balanced evaluation data split, comprising 383 images, while achieving an almost perfect precision, recall, and F1-score in each class, ranging from 97% to 99%.The only six misclassifications are three each from both classes, proving the existence of hydration-related visual cues in facial images using transfer learning with ImageNet weights. This aligns with the results of previous benchmarks conducted by Wen et al. in [5], with an AUC value of 0.912, and Lee et al. in [3], with an accuracy of 87.4% in predicting skin types.

The continuous heatmap, through the implementation of QuickShift-based superpixel segmentation followed by positive class weight alpha blending, has provided spatially consistent attributions of the model's decisions, highlighting physiologically relevant areas of interest in the facial images. These areas include the perioral, nasolabial, and periocular regions. These results have far exceeded the binary top-k masking-based explanation methods used in previous studies. The React-based HydraVision AI frontend has provided consistency in the frontend-backend vision using Reds-colourmap-based gradient mapping of the classification results onto a simulated heatmap grid of size 20x20. It has also enabled the propagation of classification results onto the chat panel of the HydraaBot through the implementation of AppContext.

With HydraBot's RAG model, mode switching system prompt, and symptom conditioned chat_with_symptoms() function, it is therefore possible to overcome the cold start problem that was associated with the earlier skincare chatbots, as described in reference [20]. At the same time, it is therefore possible to minimize diagnostic ambiguity, in accordance with the results that were described in reference [19]. Finally, domain restriction, in accordance with a specific instruction in the prompt, and a refusal template can be utilized in order to overcome the hallucination problem that was described in reference [21].

With all these contributions, it is therefore possible to create a coherent proof of concept that relates to a non-invasive and explainable method of assessing and measuring hydration level and individual AI-based skincare advice, overcoming all the problems that were described in all of the literature that was reviewed above.

2. Future Work

Five major extensions are proposed. Firstly, the binary classifier will be replaced by a continuous hydration regression head, normalizing with concurrent corneometer and TEWL measurements, to allow the HydraVision AIHydrationGauge component to render a continuous score over five clinically relevant severity bands and remove ambiguity inherent to binary classification.

Secondly, the 15-30 seconds CPU latency associated with the LIME model will be mitigated through the implementation of a hybrid LIME + Grad-CAM++ model. This will allow for sub-second gradient-based attribution for real-time feedback, reducing the discrepancy between the frontend's claim of sub-3 seconds and the computational reality.

Thirdly, NIR channel integration will be added to the model, utilizing the identified differentials in absorption based on the presence and level of moisture absorption identified by Anderson and Parrish [11] and the physics-informed radiative transfer equation constraints identified by Shi et al. [13].

Lastly, the text-only HydraBot model will be replaced by a fine-tuned vision-language model, which will allow for spatially grounded

reasoning directly on the output of the LIME heatmap model. This is similar to the proposed LLaVA-Med multimodal grounding architecture [18] and will allow HydraBot to reference particular regions of the face identified as dehydrated and formulate recommendations accordingly.

Fifthly, the Longitudinal Trend Tracking option, as depicted on the HydraVision AIhomepage, will be included in the design, which will allow the correlation of results from repeated hydration levels and user-submitted information on their lifestyle habits (water intake, sleep, product usage, etc.). This will introduce the temporal factor in the design, which will then be able to transform into a personal skin wellness monitoring system, in conformity with the evolution of AI-aided dermatological management, as discussed in Chow et al. [33].

References

J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, 2019, pp. 4171–4186.

V. Kaul, B. Nemade, and V. Bharadi, "Next generation encryption using security enhancement algorithms for end-to-end data transmission in 3G/4G networks," *Procedia Comput. Sci.*, vol. 79, pp. 1051–1059, 2016.

A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

H. B. Kekre *et al.*, "Performance comparison of DCT, FFT, WHT, Kekre's transform and Gabor filter based feature vectors," *Int. J. Comput. Appl.*, 2011.

S. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, 2004.

B. Nemade, S. Moorthy, and O. Kadam, "Cloud computing: Windows Azure platform," in *Proc. Int. Conf. Emerging Trends Technol.*, 2011, pp. 1361–1362.

M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2016, pp. 1135–1144.

S. Firth *et al.*, "The efficacy of smartphone-based mental health interventions for depressive symptoms," *World Psychiatry*, vol. 16, no. 3, pp. 287–298, 2017.

A. Dal Pozzolo *et al.*, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.

J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, 2012.

J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2545–2557, 2017.

B. Nemade and D. Shah, "IoT-based water parameter testing in linear topology," in *Proc. Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, 2020, pp. 546–551.

S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4765–4774.

P. Horodyski, "Applicants' perception of artificial intelligence in the recruitment process," *Comput. Human Behav. Rep.*, vol. 11, p. 100303, 2023.

R. W. Picard, "Affective computing for mental health," *IEEE Comput.*, vol. 43, no. 11, pp. 32–39, 2010.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

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

R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 474–481, 1998.

A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Process. Lett.*, vol. 12, no. 6, pp. 441–444, 2005.

W. Tang *et al.*, "An automatic cost learning framework for image steganography using deep reinforcement learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 952–967, 2021.

L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 5, pp. 814–825, 2014.

D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*. Stanford, CA, USA: Stanford Univ., 2023.

M. J. Dworkin, "Recommendation for block cipher modes of operation: Methods and techniques," NIST SP 800-38A, 2001.

D. Hardt, "The OAuth 2.0 authorization framework," RFC 6749, 2012.

B. Kaliski, "PKCS #5: Password-based cryptography specification version 2.0," RFC 2898, 2000.

G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, 2016.

S. Voloshynovskiy *et al.*, "Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks," *IEEE Commun. Mag.*, vol. 39, no. 8, pp. 118–126, 2001.

Google LLC, "Overview of Manifest V3," Chrome Developers Documentation, 2023.

Vercel Inc., "Next.js documentation," 2024.

B. Nemade *et al.*, "A comprehensive review of SMOTE-based oversampling methods," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 9s, pp. 790–803, 2023.

V. Shirsath *et al.*, "Intelligent traffic management for vehicular networks using machine learning," *ICTACT J. Commun. Technol.*, vol. 14, no. 3, 2023.

B. Nemade *et al.*, "Enhancing information security in multimedia streams using optimization techniques," *ICTACT J. Commun. Technol.*, vol. 14, no. 3, 2023.

B. Nemade *et al.*, "Amphibious trash collector system," *Riv. Ital. Filos. Anal. Junior*, vol. 14, no. 2, pp. 1360–1371, 2023.

R. Mishra *et al.*, "Improved inductive learning approach in distributed systems," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 10s, pp. 942–953, 2023.

B. C. Surve, B. Nemade, and V. Kaul, "Nano-electronic devices with machine learning capabilities," *ICTACT J. Microelectron.*, vol. 9, no. 3, pp. 1601–1606, 2023.

S. S. Alegavi *et al.*, "Revolutionizing healthcare through wearable biomedical systems," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 9s, pp. 752–766, 2023.

B. Nemade, J. Nair, and B. Nemade, "Efficient GDP growth forecasting using modified LSTM," *Commun. Appl. Nonlinear Anal.*, vol. 31, no. 2s, pp. 339–357, 2024.

B. Marakarkandy *et al.*, "Enhancing multi-channel consumer behavior analysis using Apriori algorithm," *J. Electr. Syst.*, vol. 20, no. 2s, pp. 700–708, 2024.

G. Khandelwal *et al.*, "Water quality prediction for aquaponics farming," *Adv. Nonlinear Var. Inequal.*, vol. 27, no. 3, pp. 302–316, 2024.

V. Kulkarni *et al.*, "ADHD detection using convolutional neural networks," *Front. Psychiatry*, vol. 15, 2024.

V. A. Bharadi, S. S. Alegavi, and B. Nemade, "Using CNN-LSTMs and transformer RNNs for COVID-19 prediction," in *Proc. Int. Conf. Netw. Multimedia Inf. Technol. (NMITCON)*, 2023, pp. 1–10.

B. Nemade *et al.*, "An adaptive transformer-based framework for brain activity mapping," *Front. Hum. Neurosci.*, vol. 19, 2025.

B. Nemade, S. S. Alegavi, and V. Bharadi, "Graph attention dialog network-based drug recommendation model," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 5908–5916, 2025.