

Dermfridge AI

Kenneth Vaughn Daniel

vaughndanielsiburian@gmail.com
Hanyang University
Seoul, South Korea
Universitas Gadjah Mada
Yogyakarta, Indonesia

Lahraoui Amina

aminalahraoui12@gmail.com
Hanyang University
Seoul, South Korea
Al Akhawayn University
Ifrane, Morocco

Lam Henrik

henriklam5555@gmail.com
Hanyang University
Seoul, South Korea
Chalmers University of Technology
Gothenburg, Sweden

Howlader Shajnin

shajninhowlader@gmail.com
Hanyang University
Seoul, South Korea
Muhlenberg College
New York, United States

Belarbi Leo

belarbi.leo@pm.me
Hanyang University
Seoul, South Korea
ESGI
Paris, France

Abstract—This project presents an AI-powered personal assistant designed to connect nutrition and skincare through intelligent data analysis. By combining facial image recognition with food identification, the system determines the user's skin type and analyzes the contents of their refrigerator to recommend personalized meals, snacks, and beverages that promote healthier skin. Beyond personalized dietary guidance, the application encourages users to make optimal use of their available ingredients, thus reducing food waste.

Driven by the rise of personalized health technologies and the increasing accessibility of computer vision tools, this assistant bridges the gap between skincare and nutrition-focused applications. It supports both personal well-being and environmental sustainability by helping users understand the impact of dietary habits on their skin condition and guiding them toward smarter, skin-friendly food choices. Ultimately, the project aims to empower users to maintain healthy skin naturally while fostering more sustainable and informed eating practices.

I. INTRODUCTION

A. Problem Statement

The assistant is an initiative that seeks to examine the situation of the user with regards to the quality of his or her skin and prescribe foods that can contribute to improving or sustaining the skin condition depending on the food that he or she already has in his or her refrigeration. The application determines the skin type of the user (e.g., oily, dry, acne-prone, dull) by image recognition and nutritional analysis and cross-checks it with the recognized food to recommend meals, snacks, or drinks that are best suited to the needs of their skin.

On the one hand, when the fridge of a particular user does not contain the nutrients that are helpful to their skin type, the application could remind the user about what would be missing and provide recommendations about the simplest alternatives or grocery selections.

Diet may affect skin health in many ways, but not everyone knows that their daily meals have an effect on the state of their skin. Consequently, people can eat foods that unintentionally aggravate certain skin problems or omit the type of foods that might enhance their complexion. Moreover, individuals can leave food in their fridge as they do not know how to use it properly or in a healthy manner.

We are working towards offering a smart assistant, which bridges between two spheres nutrition and skincare, allowing users to make the most of the items at their disposal, as well as helping them maintain and improve the condition of their skin. The system is meant to inform the users on the impact of diet on their skin and provide them with better knowledge on how to make their food more skin-friendly.

B. Motivation

This project concept is based on the growing trend in personalized health and beauty through artificial intelligence especially in the region of South Korea. Although many people use skincare products, they do not consider the significance of food in the attainment of healthy skin. As an example, some of the nutrient deficiencies or surpluses (such as sugar or dairy) may have a severe impact on the state of acne, dryness, or dullness.

The AI-powered skin scanners and food identification technologies have, meanwhile, become more affordable. Nevertheless, the majority of the tools available are either based on skincare products or overall nutrition not on links between the two.

Thus, we have the motivation to create the application that will close this gap: a personal AI assistant that will identify skin conditions based on facial analysis and will suggest the meals based on the ingredients that the user has and use computer vision and food data.

The project can also be associated with the increasing popularity in the worldwide context of eliminating food waste and ensuring sustainable and healthy lifestyles. Our system promotes personal and environmental wellness objectives by motivating users to utilize what is already present in their fridge by utilizing it in a manner that helps them treat their skin.

In the end, the application will offer its users convenient and AI-based insights that assist them in eating smarter to get a better skin and spend less money on unnecessary purchases or wasted food.

C. Existing Products

It is not a surprise that there are already existing smart fridges and AI mirror tools in today's market, however, there is no current tool that combines the two, leaving a gap which we hope to fill.

Samsung MicroLED smart mirror [1]: A classic mirror with the functionality of a skin analysis from a salon. The mirror was developed by Samsung in collaboration with leading Korean beauty brand Amorepacific, and is able to analyze the user's skin in 30 seconds, and provide personalized skin care recommendations, and a list of products that matches specific needs. The algorithm was developed by Amorepacific, using data from 20,000 skin diagnoses, providing users with 85% accuracy. Artificial intelligence also plays a role in this.

Samsung AI Vision Inside [2]: Automatically tracks what is being put in your fridge, and taken out, using a camera positioned above. Allows you to view what is inside your fridge from your phone, using their smartThings app. The system leverages AI technology to identify foods, recommend customized recipes, and connect with cooking appliances. The system additionally tracks expiration dates.

MySkin by Cetaphil [3]: MySkin by Cetaphil assesses your skin and provides a tailored skincare solution just for you through the power of innovative artificial intelligence (AI) skin technology. You have to scan their website's QR code to be directed to the browser for MySkin by Cetaphil, where you are able to take a selfie, which is then analyzed. You are able to receive a skincare analysis report and skincare product recommendations. Their AI is powered by PerfectCorp AI, and provides usage for their API on their site.

Stroke Striker [4]: Strokes can be pre-diagnosed with a method called BE-FAST (Balance, Eyes, Face, Arm, Speech, Terrible headache) that analyzes facial expression changes due to the paralysis of facial muscles. This AI-Based application aims to provide a preemptive and active health care service that periodically checks the health of the people living in a household, in order to detect signs of diseases in advance, by scanning faces using a camera that would be placed inside the fridge at head level.

II. ROLE ASSIGNMENTS

TABLE I
ROLE ASSIGNMENTS

Role	Name	Responsibilities
User	Shajnin, Amina	
Customer	LG Electronics	
Developer	Leo, Vaughn, Amina, Henrik, Shajnin	
Development Manager	Leo	

III. REQUIREMENTS

A. Web Interface

The application takes the form of a simple and fluid web interface. When opened, the user can choose between two main modes of interaction:

Face Scan: allows real-time or photo-based facial analysis to evaluate skin condition and generate personalized recommendations.

Inventory: opens a dashboard displaying the user's fridge contents, detected automatically or entered manually.

The interface is fully responsive and adapts to computers, tablets, and smartphones, ensuring clarity and comfort for all users.

B. Authentication and Profile

An authentication system allows users to create and manage their personal profiles. Each account securely stores facial analysis history, preferences, and inventory data. This ensures personalized recommendations and consistent tracking across sessions.

C. Settings

A configuration section enables users to customize their experience. Options include dietary preferences, notification frequency, language selection, and privacy controls. This personalization ensures an adaptable and user-friendly interface.

D. Analysis

Each captured image or video frame is processed by the AI model to extract relevant indicators. The system interprets facial features to assess hydration, oiliness, or potential deficiencies, and generates corresponding dietary advice. In the inventory mode, objects are recognized, categorized, and linked to nutritional information to match facial recommendations.

E. Error and anomaly management

The system detects and handles common issues such as camera access failure, poor lighting, communication loss, or data storage errors. In every case, the user is clearly notified, and the software minimizes data loss and service interruption.

F. Database

All analyses and user data are stored in a structured and secure database. Each record includes timestamps, extracted indicators, recommendations, and user identification. This organization allows tracking, statistics, and continuous model improvement while maintaining privacy standards.

G. Performance

The application has been designed to run efficiently on most hardware configurations, optimizing image processing via the CPU without requiring a dedicated GPU. The performance goal is to maintain a processing time of less than 1.5 seconds per image on a standard workstation, in order to guarantee a smooth experience, even when used in real time via camera.

H. IoT integration

A future upgradeable version of the project is embedded integration on connected devices such as smart trash cans or sorting terminals. This version will be based on a lightweight containerized architecture, combined with a camera and sensors, enabling food detection and processing without dependence on the cloud.

Artificial Intelligence:

The system relies on AI modules that process facial data and fridge inventory to produce insights and recommendations.

IV. PART 3 DEVELOPMENT ENVIRONMENT

A. Development Platform

We choose to work on Docker as our work environment. This solution ensures consistency across all development setups while allowing each team member to use their preferred operating system, which for our team is Windows or macOS,..

With Docker, each developer benefits from an isolated, reproducible environment identical to those of other team members, eliminating compatibility issues and simplifying dependency management.

This approach facilitates the transition to production, where services will be deployed as containers orchestrated by Kubernetes.

B. Technical Stack

For the application's development, we decided to rely exclusively on open-source tools to ensure flexibility, transparency, and compatibility with modern software development standards.

1) Frontend:

- **React:** Used to build a dynamic, component-based web interface that ensures a smooth and responsive user experience.
- **TypeScript:** Adding more specificity to JavaScript variable, parameter, naming, etc. It helps with understanding code errors, readability, and maintainability of the frontend codebase.
- **CSS:** Enables design for pages and creates a consistent aesthetic that matches our branding.

2) Backend:

- **FastAPI:** A high-performance Python framework used for building the REST API, providing efficient request handling and easy integration with AI modules.

3) Analysis AI:

- **?:** Used for image recognition and object detection, enabling accurate food and skin analysis through deep learning models.

4) Database:

- **PostgreSQL:** A robust open-source relational database used to securely store user profiles, scan data, and analysis results.

5) Development Tools:

- **Git & GitHub:** Version control and collaborative platform for managing code changes and coordinating team work.
- **Docker:** Provides containerized environments for consistent deployment and development across all systems.
- **Kubernetes:** Used for container orchestration and automated deployment in production environments.
- **Visual Studio Code:** Main development environment offering extensibility and integrated debugging tools.
- **GoogleDocs:** Used for document editing and report preparation.
- **KakaoTalk:** Serves as a lightweight communication channel for daily coordination within the development team.

C. Cost Estimation

Software costs are minimal, as all development tools used are free and open-source. Hardware costs are covered by each team member's personal machine.

Deployment costs will involve hosting the application on Amazon Web Services for production, which simplifies infrastructure management, scalability, and security. Main resources include an Amazon EKS (Elastic Kubernetes Service) cluster, an RDS instance for the backend database, S3 storage for images and files, and a Domain Name System (DNS) for hosting the web interface.

V. PART 4 SPECIFICATION

A. Web Interface

The application presents a simple and responsive web interface. Upon opening, the user can choose between two interaction modes:

Face Scan

Accesses the facial analysis module.

Users can choose between:

Live Camera Scan: Real-time video stream for continuous facial analysis.

Take a Photo: Captures a single image for one-time assessment

Import a Photo: Allows upload of an image from the device.

The results are processed and linked to the user's profile.

Inventory

Opens the user's fridge inventory dashboard.

Displays a structured list of all detected or manually added items (fruits, vegetables, meats, drinks, etc.).

Two action buttons are located at the bottom of the screen:

Scan Items: Uses the prototype inventory camera to detect items inside the fridge.

Manual Input: Allows users to add food or drink items that are not automatically detected.

```
INPUT: user_mode (Face Scan or Inventory)
PROCESS:
1. Load main interface
2. IF user_mode == "Face Scan":
   Open facial analysis module
   Capture or upload image
   Display SKIN_PROFILE and recommendations
3. ELSE IF user_mode == "Inventory":
   Open inventory dashboard
   Detect or add food items
   Show FOOD_LIST and nutritional data
4. Adjust interface layout based on
device type
OUTPUT: Interactive web view displaying
analysis or inventory results
```

B. Authentication and Profile

The application includes an authentication system that allows users to create and manage personal profiles. Each profile enables consistent tracking and recommendations.

Core account features include:

Registration and login with encrypted credentials.

Secure session management for authenticated access.

Each user profile stores:

Facial scan history and generated recommendations.

Dietary preferences and restrictions.

Linked fridge inventory data.

Privacy controls ensure:

Encrypted database storage.

Ability to clear or delete personal history.

```
INPUT: user_email, user_password,
user_action
PROCESS:
1. IF user_action == "Register":
   Validate input fields
   Hash password and create account
   Store user data securely
2. IF user_action == "Login":
   Verify credentials
   IF valid:
      Load user_profile and session
   ELSE:
      Show "Invalid credentials" message
3. Allow user to update preferences or
clear history
OUTPUT: Authenticated session with
stored user profile
```

C. Settings

The application provides configurable settings for a personalized user experience. Key requirements include:

Dietary restrictions and preferences management.

Notification frequency and recommendation detail level.

Data privacy and security settings, including account management and control over stored scan history.

Enable/disable automatic inventory detection from the prototype camera.

The settings module ensures that users can fully control their experience, privacy, and the prototype device configuration.

```
INPUT: user_profile, updated_settings
PROCESS:
1. Load current_settings from
user_profile
2. Display options:
   diet_type, notification_preference,
language_choice, privacy_option
3. FOR each setting in updated_settings:
   IF setting value is valid:
      current_settings[setting] ←
updated_settings[setting]
   ELSE:
      show_error("Invalid setting for " +
setting)
4. Save current_settings to database
5. Apply current_settings to active user
session
6. Notify user: "Settings successfully
updated"
OUTPUT: Updated user_profile with new
settings applied
```

D. Analysis

The software processes each facial image to extract and interpret key indicators. The analysis provides:

Skin condition: hydration, oiliness, acne, redness, or wrinkles.

Possible nutritional deficiencies derived from facial indicators.

Lifestyle-related insights based on facial tone or texture.

Based on these results, the system generates food recommendations categorized by their benefits:

Foods that improve hydration or glow.

Foods rich in vitamins and minerals identified as lacking.

General lifestyle or dietary advice.

```

INPUT: face_image, fridge_image,
user_profile
PROCESS:
1. Load face_image from user input
2. Extract skin indicators using facial
analysis model
skin_data = {hydration, oiliness, acne,
redness, wrinkles}
3. Detect and categorize food items from
fridge_image
food_list = {fruits, vegetables, meats,
drinks}
4. Compare skin_data with nutritional
values in food_list
nutrient_match = matchNutrition(skin_data,
food_list)
5. Generate personalized recommendations
recommendations =
createRecommendations(nutrient_match)
6. Save results to user_profile for
history tracking
7. Display summary of analysis and
recommended foods to user
OUTPUT: Personalized nutrition and
skincare recommendations

```

E. Error and anomaly management

The system must be able to detect and manage different types of errors that may occur during use:

Camera access error: permission denied or no video signal detected.

Analysis error: poor lighting or incomplete face capture.

Communication error: interruption between frontend and backend.

Database error: failure to record data.

The user is clearly informed of the issue, and data loss or service interruption is minimized.

```

INPUT: system_event
PROCESS:
1. IF system_event == "Camera Error":
Notify user to check camera access
2. IF system_event == "Lighting Error":
Ask user to improve lighting conditions
3. IF system_event == "Network Error":
Retry connection and log failure
4. IF system_event == "Database Error":
Save unsaved data and alert
administrator
5. Display clear message to user for
each error type
OUTPUT: Error resolved or safely handled
with user notification

```

F. Database

All scan and analysis data are stored in a structured database. Each record includes:

Timestamp of the scan.

Processed image or extracted features.

Detected indicators and confidence levels.

Generated food and nutrition recommendations.

Associated user ID.

The database supports secure storage, history tracking, and statistical processing while complying with privacy standards.

```

INPUT: user_profile, analysis_results
PROCESS:
1. Open connection to main database
2. Create data_entry with:
user_id = user_profile.id
timestamp = getCurrentTime()
indicators = analysis_results.skin_data
recommendations =
analysis_results.recommendations
3. Encrypt user_id and sensitive fields
before saving
4. Insert data_entry into table
"user_records"
5. Verify successful insertion
6. Close database connection
OUTPUT: Stored analysis record linked to
user_profile

```

G. Performance

The system ensures optimal performance under all conditions. Key performance requirements include:

Real time facial scan processing for instant feedback and recommendations

Accurate and responsive detection of fridge inventory
Optimized resource usage to ensure smooth operation

The system is designed to maintain consistent speed and reliability also providing accurate recommendations.

```

INPUT: image_request, device_specs,
system_resources
PROCESS:
Start performance_timer
Run facial_analysis =
processFace(image_request)
Run food_detection (image_request)
total_time = measure(performance_timer)
Record metrics including total_time,
CPU_load, and device_specs
Return combined results to user
interface
OUTPUT: Stable, optimized performance
with system feedback.

```

H. IoT integration

The application integrates seamlessly with IoT-enabled devices. Key requirements include:

One camera dedicated to facial scanning for real-time analysis

One camera dedicated to detecting available fruits, vegetables, and meats, simulating fridge inventory.

Manual input option for foods not detectable by the inventory camera.

AI driven recommendations based on facial analysis and detected inventory for personalized advice.

The IoT prototype ensures that the system can demonstrate functionality and provide accurate recommendations without a fully connected smart fridge

```

INPUT: camera_data, sensor_data,
network_status, user_profile
PROCESS:
    Collect camera_data from both
    facial_camera and fridge_camera
    Analyze captured images using local
    detection models
    face_result = camera_data
    inventory_result =
    detectInventory(camera_data)
    Combine results into analysis_report
    Generate recommendations based on
    face_result and inventory_result
    OUTPUT: IoT operation with synchronized
    data and offline support
  
```

I. Artificial intelligence and model

The system relies on AI modules that process facial data and fridge inventory to produce insights and recommendations.

Facial Analysis Model interprets key indicators:

Skin condition: hydration, oiliness, acne, redness, or wrinkles.

```

INPUT: User facial image (FACE_IMG)
PROCESS:
    1. Load Facial Analysis Model
    2. Detect skin indicators:
        hydration_level ←
        analyze_hydration(FACE_IMG)
        oiliness_level ← analyze_oiliness(FACE_IMG)
        acne_presence ← detect_acne(FACE_IMG)
        redness_intensity ←
        detect_redness(FACE_IMG)
        wrinkle_depth ← measure_wrinkles(FACE_IMG)
    3. Summarize results into SKIN_PROFILE:
    SKIN_PROFILE = {
        "hydration": hydration_level,
        "oiliness": oiliness_level,
        "acne": acne_presence,
        "redness": redness_intensity,
        "wrinkles": wrinkle_depth
    }
    OUTPUT: SKIN_PROFILE
  
```

Based on these findings, targeted recommendations are generated:

Foods that improve hydration, glow, or elasticity.

Ingredients rich in missing vitamins and minerals.

Reminders to avoid foods that aggravate skin imbalance.

```

INPUT: SKIN_PROFILE
PROCESS:
    1. Analyze SKIN_PROFILE for imbalances
    or deficiencies
    2. Based on findings:
        IF hydration_level is LOW:
  
```

```

Recommend foods rich in water, vitamin
E, and omega-3
(e.g., cucumbers, avocados, salmon)
IF oiliness_level is HIGH:
Recommend foods that regulate sebum
(e.g., leafy greens, zinc-rich nuts)
IF acne_presence is TRUE:
Recommend antioxidant and vitamin A
sources
(e.g., carrots, berries, green tea)
IF redness_intensity is HIGH:
Recommend anti-inflammatory foods
(e.g., turmeric, oatmeal)
IF wrinkle_depth is DEEP:
Recommend collagen-boosting foods
(e.g., citrus fruits, tomatoes)
3. Identify foods to AVOID that worsen
detected conditions
(e.g., sugar, fried foods, processed
meat)
OUTPUT:
RECOMMENDATIONS = {
    "foods_to_add": [...],
    "foods_to_avoid": [...],
    "nutrients_needed": [...]
}
  
```

Food Detection Model supports inventory analysis:
Recognizes items using YOLOv8-based object detection.

Extracts categories such as fruits, vegetables, meats, and beverages.

Associates confidence scores with each detected product.
Links to nutritional data for recommendation matching.

```

INPUT: User pantry or fridge image
(FOOD_IMG)
PROCESS:
    1. Load YOLOv8-based Food Detection
    Model
    2. Detect food items in FOOD_IMG
    3. For each detected object:
        item_name ← object.label
        confidence ← object.confidence_score
        category ← classify_into_group(item_name)
        # (e.g., fruit, vegetable, meat,
        beverage)
    4. Create FOOD_LIST containing all
    detected items and metadata
    5. Link each detected item to nutrition
    database entry:
    FOOD_LIST[i].nutrients =
    lookup_nutrients(item_name)
    OUTPUT:
    FOOD_LIST = [
        {"name": ..., "category": ...,
        "confidence": ..., "nutrients": ...}
    ]
  
```

REFERENCES

- [1] Samsung MicroLED Smart Mirror. Available: <https://samsungmagazine.eu/en/2025/01/12/chytre-zrcadlo/>
- [2] Samsung AI Vision Inside. Available: <https://www.idsa.org/awards-recognition/idea/idea-gallery/ai-vision-inside/>
- [3] MySkin by Cetaphil. Available: <https://www.cetaphil.com/us/skin-analysis.html>

- [4] Stroke Striker concept video. Available: https://www.youtube.com/watch?v=XgII0ev_tus&t=20s