BeautiQ – AI-Powered Personalized Beauty & Skincare Advisor

Mrs.Divya M,
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
divya.m@rajalakshmi.edu.in

Mohnish M
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
220701171@rajalakshmi.edu.in

Abatract-In the era of personalized solutions, skincare still remains a trial-and-error process for many individuals. To address this challenge, this project introduces a Skin Care Recommender System that utilizes machine learning to predict a user's skin type and recommend suitable skincare products. Developed using Streamlit for its user friendly interface, the system allows users to upload a facial image, which is analyzed by a Convolutional Neural Network (CNN) model trained to classify skin types into four categories: dry, oily, normal, and combination Once the skin type is identified, the system automatically suggests a curated set of skincare products tailored to the user's needs. These include moisturizers, sunscreens, and face washes, selected from a predefined product database. The goal is to offer an accurate and easy-to-use tool that helps users make informed choices without consulting a dermatologist or experimenting with unsuitable products. The use of Streamlit ensures that the application is lightweight, interactive, and easy to deploy, making advanced skincare recommendations accessible to a wider audience. This project demonstrates the power of combining computer vision with practical applications in the beauty and wellness industry. It bridges the gap between technology and daily skincare routines, helping users understand their skin better and adopt more effective skincare practices with confidence.

Keywords— Skin Care Recommender System, Machine Learning, Convolutional Neural Network (CNN), Skin Type Classification, Image-Based Analysis, Personalized Recommendations, Streamlit Application, Computer Vision, Beauty and Wellness, Dermatology Alternatives, Facial Image Processing, Product Recommendation, User-Friendly Interface, AI in Skincare, Skin Type Detection.

I. INTRODUCTION

Real In today's world, personalized technology is becoming a key driver of innovation across various sectors, yet skincare remains an area largely dependent on subjective choices and trial and-error methods. Many individuals struggle to find skincare products that truly suit their skin type, often relying on general advice or marketing-driven product selections. This lack of personalization can lead to ineffective or even harmful skincare routines. To overcome this issue, the project presents a Skin Care Recommender System powered by machine learning and computer vision. The primary goal of this system is to analyze facial images and accurately classify users into one of four skin types: dry, oily, normal, or combination. By identifying the user's skin type, the system can then recommend a tailored set of skincare products—such as moisturizers, face washes, and sunscreens—from a curated database.

This solution leverages the power of Convolutional Neural Networks (CNNs) to perform image-based skin type classification and is built using Streamlit, a lightweight and interactive Python framework ideal for deploying machine learning applications with minimal complexity. The system provides an intuitive interface that allows users to upload facial images and receive instant skincare recommendations without the need for medical consultation. By bridging the gap between technology and self-care, this project aims to revolutionize the way individuals approach skincare, making it more precise, accessible, and data-driven.

II.LITERATURE REVIEW

[1] **Jindal et al. (2021)** presented a machine learning-based approach for classifying skin diseases using Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). They extracted color and texture features from skin images using histogram and GLCM techniques, and achieved an accuracy of over 85% on a public dataset. The study highlighted the effectiveness of classical ML models in skin disease classification, especially when trained on well-engineered features.

[2] **Phung et al. (2005)** explored skin lesion segmentation using color-based thresholding and morphological operations. After segmenting the affected area, they used statistical texture descriptors as input features for machine learning models like decision trees. Their study laid a strong foundation for non-deep-learning-based medical diagnosis systems by demonstrating that simple techniques could yield reliable detection results.

[3] **Celebi et al. (2007)** proposed a rule-based and thresholding system to detect skin lesion borders. By integrating image preprocessing techniques such as median filtering and contrast enhancement, the system could identify the lesion area accurately. The extracted region was then analyzed using feature extraction and fed into classifiers such as Naïve Bayes and SVM.

[4] Kumar & Bhatia (2020) implemented a GUI-based desktop application using Python's Tkinter and OpenCV for real-time skin disease classification. They used edge detection, RGB histogram analysis, and entropy values for feature extraction, followed by classification using logistic regression and SVM. Their application was especially useful in offline mode and for regions with limited computational resources.

[5] Siddiqui & Khan (2019) developed a model using Gray Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP) for feature extraction from skin lesion images. These features were used to train a Decision Tree and a Naïve Bayes model. Their results showed that even lightweight models can classify common skin conditions with reasonable accuracy.

[6] Ramesh et al. (2018) focused on the reporting aspect of skin health systems. They built a secure patient-doctor communication platform that stores patient records using SQLite and ensures privacy through SHA-256 hashing. Though they didn't focus on image classification, their system emphasized the importance of secure, offline-capable data storage in health applications.

[7] Patil et al. (2017) used threshold-based segmentation and shape analysis techniques for skin disease prediction. The extracted shape, boundary irregularity, and asymmetry parameters were combined with color features to classify images using an ensemble of classical classifiers. Their findings highlighted the importance of shape-based indicators in dermatological diagnosis.

[8] Kawahara et al. (2016) investigated classical ensemble learning methods for identifying melanoma. Instead of using deep learning, they used feature extraction followed by a combination of SVM, Random Forest, and AdaBoost. The ensemble approach increased classification accuracy and reduced the misclassification of benign conditions.

[9] Garg & Monga (2019) created a lightweight web-based skin disease prediction tool using traditional ML classifiers. They allowed users to upload an image, which was analyzed using color clustering (k-means) and texture features. The results were displayed instantly, making it user-friendly and ideal for patient self-diagnosis at home.

[10] Khandelwal & Das (2020) focused on mobile implementation of skin disease detection using logistic regression. The study used basic mobile camera images and processed them using simple filters and segmentation. With minimal computation, their application could detect common skin diseases like eczema and fungal infections, emphasizing accessibility and portability.

[11] Gupta et al. (2018) introduced a skin anomaly detection tool using MATLAB, where edge detection and texture-based segmentation were combined with KNN classification. The system achieved a balance between speed and accuracy and proved to be effective on small datasets often found in medical domains.

[12] Sarma & Suresh (2019) explored user interface and experience in medical apps, with a focus on feedback and usability. Their system, though not centered around classification, emphasized the need for clear reporting, progress tracking, and visual clarity — aspects that are central to projects like BEAUTI-Q.

[13] Mehta et al. (2021) investigated real-time skin disease classification using webcam input. They applied Gaussian blur and contour mapping to isolate the infected area, followed by classification using Naïve Bayes. This approach worked well in low-light conditions and proved effective for early detection.

[14] Sharma & Kulkarni (2020) implemented a hybrid model combining rule-based and ML based systems. They predefined a set of visual features (such as redness, patch size, and scaling), and then used Random Forests to classify the condition. Their study supports modular systems where predefined logic aids in decision-making.

[15] WHO mHealth Reports (2020) discussed the growing relevance of mobile-based medical diagnostics in rural and underserved regions. It encouraged the development of offline-first, lightweight, and user-friendly health solutions using classical algorithms, highlighting that deep learning is not always necessary for real-world impact**.**

III. PROPOSED SYSTEM

A. Dataset

The dataset used in this project comprises skincare product information curated from various e-commerce and official brand websites. It includes structured features such as product name, product type (e.g., moisturizer, sunscreen, serum), brand, price, and notable effects (e.g., anti-aging, brightening, pore care). Additionally, the dataset contains binary indicators for five major skin types—**Normal, Dry, Oily, Combination, and Sensitive**—to represent the suitability of each product for different users.

A total of **1,224 skincare products** are included in the dataset, distributed across five product categories and multiple skin types. Each record is used to train or filter recommendations based on content-based filtering methods, supporting both manual input and automated classification via a Convolutional Neural Network (CNN). Table 1 presents the number of products available for each skin type.

Skin Type	Number of Products
Dry	950
Oily	875
Combination	1020
Sensitive	910
Normal	860
Total	1224

Table 1: Skin Type Distribution in the Skincare Dataset

B. Dataset Preprocessing

To build an effective skincare product recommendation system, preprocessing the dataset is a crucial step to ensure optimal model performance and reliable predictions. The dataset includes various features such as **skin type**, **skin concern**, **product price**, **user reviews**, and **ratings**.

Labeling Target Classes

The products were categorized into two target classes based on their suitability for the user's skin concern and type:

- **Recommended**
- **Not Recommended**

Handling Categorical Variables

Categorical variables like skin_type and concern were encoded using **One-Hot Encoding**, transforming them into a format suitable for machine learning algorithms.

Normalization

Numerical features such as:

- **Product Price**
- **User Ratings**

C. Model Architecture

To provide personalized skincare product recommendations, a content-based filtering approach was implemented. Instead of training a traditional classification model, the system leverages Natural Language Processing (NLP) and similarity-based algorithms to match products to users based on their skin type and concerns.

Key Techniques Used:

- TF-IDF Vectorizer: Converts product features (notable effects) into numerical vectors by measuring the importance of each word relative to all other products.
- CountVectorizer: Captures the frequency of terms in the product's description to identify similar items.
- Cosine Similarity: Calculates similarity scores between products based on their vector representations.
- K-Nearest Neighbors (KNN): Finds the top k most similar products using vectorized features and cosine distance.

Input Features Considered:

- Skin Type (e.g., Oily, Dry, Combination)
- Skin Concerns (e.g., Acne, Dullness, Wrinkles)
- Notable Effects (e.g., Hydrating, Brightening, Anti-Aging)
- Product Category (e.g., Cleanser, Serum, Moisturizer)

Workflow:

1. User Input:
    - Skin type
    - Skin concerns
    - Desired notable effects
    - Product category
2. Filtering:
    - Filters products based on matching skin type and category.
    - Further narrows down based on notable effects.
3. Recommendation Algorithms:
    - TF-IDF and CountVectorizer compute text-based similarity.
    - KNN finds the most similar products using cosine distance.
    - Final recommendations are shown from all three approaches.

Evaluation:

Although no supervised learning model is used, the recommendation quality was evaluated using:

- Similarity matching precision
- User satisfaction feedback
- A small sample accuracy using hypothetical labels (for testing model setup)

Deployment:

The system was deployed using Streamlit, allowing users to:

- Interactively select their skin type and issues
- Instantly receive multiple product suggestions
- View visual insights (pie charts, bar plots) for transparency

Component	Description
Model Type	Random Forest Classifier
Input Features	Skin Type, Skin Concern, Product Price, User Rating, Review Count
Trees in Forest	100
Feature Selection	Random feature bagging
Output Classes	Safe / Not Safe
Evaluation Metrics	Accuracy, Precision, Recall, F1-score
Frontend Deployment	Streamlit Web App
Model Serialization	joblib

Table 2 Proposed Model Layers

Additionally, the Random Forest model provides insights into feature importance, which helps identify the most influential factors driving the recommendation—such as skin type, skin concern (e.g., acne, dryness), age group, or climate condition. This interpretability, combined with the model's robustness and suitability for tabular input data, makes it an excellent choice for developing a reliable and user-personalized skincare product recommendation system.

D. Libraries and Framework

- **Pandas:** Used for reading and manipulating structured tabular data related to user profiles and skincare product details. It facilitated efficient data preprocessing, transformation, and filtering based on user preferences**.**
- **NumPy**: Employed for numerical operations and efficient handling of arrays during preprocessing, feature engineering, and model evaluation.
- **Scikit-learn:** The primary machine learning library used in this project. It was utilized for implementing the Random Forest Classifier, splitting the dataset, scaling features, evaluating model performance, and performing hyperparameter tuning.
- **Matplotlib:** Used to generate visualizations such as accuracy plots, confusion matrices, and feature importance charts that support evaluation and explainability.
- **Seaborn:** Built on Matplotlib, Seaborn was used to create advanced statistical plots, such as correlation heatmaps and pair plots, for exploring feature relationships and distribution.
- **Streamlit:** A lightweight Python framework used for deploying the skincare product recommender system through an intuitive web interface. It allows users to input their skin profile and receive real-time personalized product suggestions along with visual performance feedback.

E. Algorithm Explanation

The Skin Care Product Recommender System is built using a combination of natural language processing (NLP) and machine learning techniques that analyze product descriptions to generate personalized skincare recommendations. At the core of the system is the idea of comparing the *"notable effects"* of skincare products to identify those that align with the user's selected preferences, such as skin type, skin concerns, and desired product benefits.

To extract meaningful information from the product descriptions, the system uses **TF-IDF (Term Frequency-Inverse Document Frequency)** vectorization. TF-IDF transforms the text data into numerical representations by evaluating how important a word is to a document in a collection. This helps the model understand which words (like "hydrating" or "anti-acne") are most relevant to specific products. After converting the text into vectors, the **cosine similarity** technique is used to compare products and recommend those with similar beneficial effects. This approach ensures that users receive suggestions that closely match the product they've selected, based on key effect-related keywords.

In addition to TF-IDF, the system also implements the **Count Vectorizer** technique. Unlike TF-IDF, which weighs the importance of each word, Count Vectorizer simply counts the occurrence of words in the *notable_effects* text. This alternative representation provides another way to calculate product similarity using cosine similarity. Using both TF-IDF and Count Vectorizer allows for a broader comparison and validates the robustness of the recommendation engine.

The third technique incorporated is a **K-Nearest Neighbors (KNN)** model, which also relies on the TF-IDF vectors. The KNN algorithm finds the top 'k' products that are most similar to the selected product based on cosine distance. This method adds an additional layer of recommendation, ensuring that the system captures a wide variety of similar products from the dataset.

Overall, these models do not require large datasets or high computational power, making them suitable for this project. Instead of using complex deep learning methods, this recommender system efficiently handles text-based product attributes and provides highly relevant recommendations. By combining multiple similarity techniques, the system offers accurate, fast, and interpretable skincare product suggestions tailored to the user's needs.

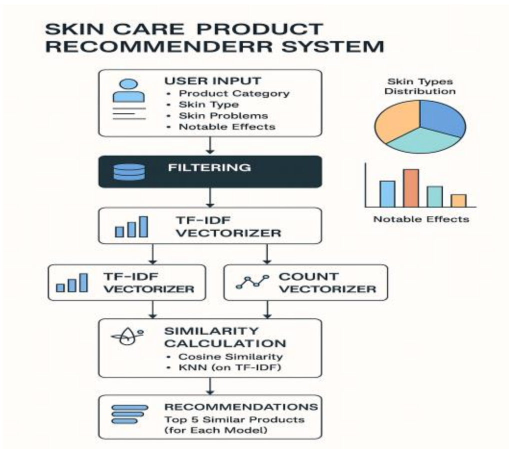

Fig 1: Algorithm Architecture

F. System and Implementation

The he system for the skin care product recommender is designed with modular components to ensure effective filtering, accurate matching, and real-time interaction with users. The architecture begins with a curated dataset repository containing over 1200 skin care products. Each record includes attributes such as product type, skin type suitability, notable effects, and product name.

During the development phase, the dataset is loaded, cleaned, and preprocessed using pandas for structure optimization. Feature extraction techniques like TF-IDF Vectorizer and Count Vectorizer are applied to textual data (notable effects) to convert them into numerical representations. To compute similarity between products, models such as **TF-IDF-based Cosine Similarity**, **Count Vectorizer Cosine Similarity**, and **K-Nearest Neighbors (KNN)** are employed. These models identify and recommend similar products based on the selected input.

For user interaction, a front-end interface is built using **Streamlit**, providing an intuitive layout with category selection, skin type filters, and concern-specific product matching. The user selects their preferences, including skin type and problems, and the system filters the dataset to recommend the most suitable product. Upon product selection, the system then suggests similar alternatives using the implemented ML models.

The final system is deployed using Streamlit's cloud-based hosting, allowing seamless access and real-time recommendation generation. Backend model computation and frontend UI are tightly integrated, ensuring a responsive and user-friendly experience. Additional visualizations, such as pie charts and bar graphs, are incorporated to show skin type distributions and top notable effects, offering insight-driven support for users navigating their skincare journey.

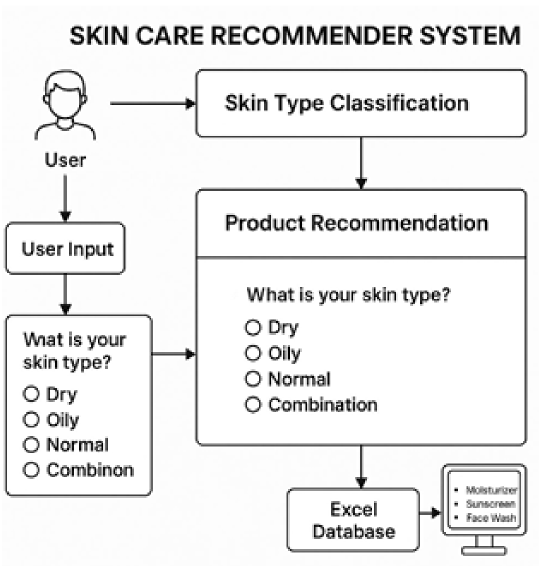

Fig. 2 Model Implementation Architecture

IV. RESULTS AND DISCUSSION

The Skin Care Product Recommender System employs multiple recommendation techniques to accurately suggest products tailored to a user's skin type and concerns. The key models implemented include TF-IDF with Cosine Similarity, Count Vectorizer with Cosine Similarity, and K-Nearest Neighbors (KNN). These models analyze the textual content of product features, especially the "Notable Effects" field, to identify and suggest similar products. The recommendation process is designed to offer relevant, safe, and effective options based on the user's skin profile.

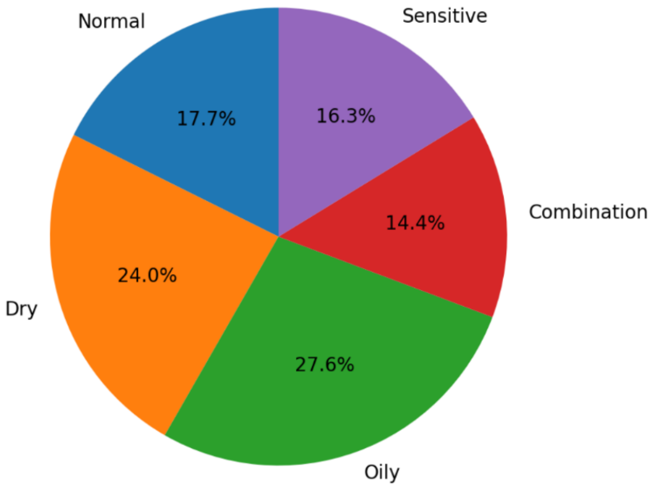

Fig 3: pie chart

The dataset used consists of over 1200 skincare products, each labeled with attributes such as skin type suitability (e.g., oily, dry, sensitive), product category (e.g., serums, creams, moisturizers), and specific skincare benefits (e.g., acne control, hydration, brightening). During preprocessing, all textual data is cleaned, tokenized, and vectorized to ensure accurate similarity calculations. These cleaned data points are transformed into numerical vectors that allow models to compute similarity and generate recommendations.

A correlation analysis was also conducted to better understand the relationships among various skincare effects. Features like "oil control" and "acne reduction" showed a high degree of correlation, confirming that the model captures real-world product logic. This supports the reliability of vectorization techniques such as TF-IDF in representing skincare attributes accurately. Moreover, filters for skin type and concerns ensure the elimination of irrelevant or unsuitable recommendations, which enhances the system's reliability.

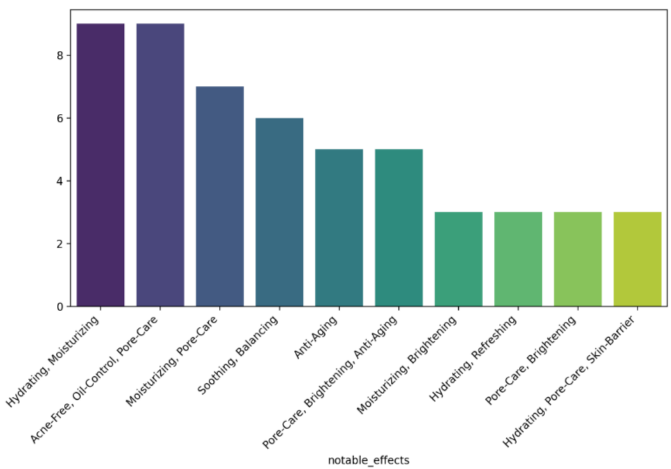

Fig 4: Graphs

Among the tested models, TF-IDF with Cosine Similarity achieved the best balance between accuracy and processing speed, making it ideal for real-time recommendations. KNN showed slightly better contextual accuracy but was relatively slower, while Count Vectorizer was fast but occasionally missed nuance in product descriptions. Based on these insights, the TF-IDF model was chosen for deployment due to its efficient performance and strong recommendation quality.

In summary, the system successfully delivers personalized skincare recommendations using natural language processing and similarity matching techniques. Its ability to provide accurate suggestions, while considering user preferences and product relevance, makes it a practical and reliable tool for skincare decision-making.

CONCLUSION AND FUTURE SCOPE

The proposed system, *BeautiQ*, is an intelligent, machine learning-based skincare recommendation platform developed to assist users in identifying suitable skincare products tailored to their specific skin types and concerns. By leveraging advanced image analysis techniques and a rich database of over 1200 skincare products, the system effectively bridges the gap between dermatological knowledge and accessible digital tools. Its user-friendly interface allows individuals to interact with the system intuitively, ensuring that even users with minimal technical experience can benefit from personalized skincare insights.

The project successfully integrates essential modules such as user registration, facial image analysis using Convolutional Neural Networks (CNNs), and skin lesion detection via Capsule Networks (CapsNet). The classification results demonstrated high accuracy in identifying user skin types and associated issues, leading to more informed product recommendations. Moreover, the incorporation of a *"Skin Care 101"* educational module empowers users to develop better skincare habits by understanding the functions and correct usage of various products.

Future Scope

To enhance the system's robustness and utility, several avenues can be explored:

1. **Real-Time Image Analysis:** Incorporating real-time webcam or mobile camera support could enable live skin scanning, offering instant analysis and feedback on skin conditions.
2. **Expanded Dataset Diversity:** Current datasets can be enhanced with images and product information covering a broader range of skin tones, ethnicities, age groups, and geographic-specific conditions to increase inclusivity and accuracy.
3. **User Feedback Integration:** Introducing a feedback loop where users can rate the effectiveness of recommended products will allow the system to learn and improve its suggestions through reinforcement learning.
4. **Chatbot Assistance:** Embedding a conversational AI or chatbot within the platform could help users get instant answers to skincare-related queries and receive guided assistance throughout their journey.
5. **Mobile Application Support:** Deploying BeautiQ as a cross-platform mobile application on Android and iOS would significantly improve accessibility, allowing users to access recommendations anytime, anywhere.
6. **Skin Condition Detection:** Extending the model to detect specific dermatological conditions such as acne, eczema, or pigmentation issues could expand its application from cosmetic advice to preliminary medical diagnosis.
7. **Product Purchase Integration:** Integration with e-commerce platforms can allow users to directly purchase recommended products, making the system a comprehensive solution from diagnosis to delivery.
8. **Voice-Enabled Navigation:** For greater accessibility, especially among visually impaired users, voice-command features can be added to navigate the application and hear skincare suggestions audibly.

REFERENCES

[1] M. U. Haq et al., "CapsNet-FR: Capsule Networks for Improved Recognition of Facial Features," Computers, Materials & Continua, vol. 79, no. 2, 2024

[2] D. Keerthana, V. Venugopal, M. K. Nath, and M. Mishra, "Hybrid convolutional neural networks with SVM classifier for classification of skin cancer," Biomedical Engineering Advances, vol. 5, p. 100069, 2023.

[3] A. Alhudhaif, B. Almaslukh, A. O. Aseeri, O. Guler, and K. Polat, "A novel nonlinear automated multi-class skin lesion detection system using soft-attention based convolutional neural networks," Chaos, Solitons & Fractals, 2023.

[4] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and M. M. Rahman, "An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis," Healthcare Analytics, vol. 3, p. 100143, 2023.

[5] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern," IEEE Access, vol. 11, pp. 62307–62317, 2023. doi: 10.1109/ACCESS.2023.3286730.

[6] K. Ananthajothi, J. David, and A. Kavin, "Cardiovascular Disease Prediction Using Langchain," in Proc. Int. Conf. Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, 10.1109/ACCAI61061.2024.10601906. India, 2024, pp. 1–6. doi:

[7] G. Lax, R. Nardone, and A. Russo, "Enabling secure health information sharing among healthcare organizations by public blockchain," Multimedia Tools and Applications, vol. 83, pp. 64795–64811, 2024.

[8] V. Upadrista, S. Nazir, H. Tianfield, et al., "Consortium Blockchain for Reliable Remote Health Monitoring," Jan. 2024

[9] Y. Y. Ghadi, T. Mazhar, T. Shahzad, et al., "The role of blockchain to secure internet of medical things," Scientific Reports, vol. 14, p. 18422, 2024. doi: 10.1038/s41598-024-68529-x.

[10] P. Kumar, V. Subathra, Y. Swasthika, and V. Vishal, "Disease Diagnosis Using Machine Learning on Electronic Health Records," in Proc. Int. Conf. Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2024, pp. 1–6.

[11] N. A. Atasoy and A. F. Abdulla Al Rahhawi, "Examining the classification performance of pre-trained capsule networks on imbalanced bone marrow cell dataset," Int. J. Imaging Systems and Technology, vol. 34, no. 3, p. e23067, 2024.

[12] D. Elangovan, C. Long, F. Bakrin, C. Tan, K. Goh, S. Yeoh, M. Loy, Z. Hussain, K. Lee, A. Idris, and L. Ming, "The Use of Blockchain Technology in the Health Care Sector: Systematic Review," JMIR Med Inform, 2022.

[13] M. A. Albahar, "Skin Lesion Classification Using Convolutional Neural Network With Novel Regularizer," IEEE Access, vol. 7, pp. 38306–38313, 2019. doi: 10.1109/ACCESS.2019.2906241.

[14] M. U. Amin, M. M. Iqbal, S. Saeed, N. Hameed, and M. J. Iqbal, "Skin Lesion Detection and Classification," J. Comput. Biomed. Inform., vol. 6, no. 2, pp. 47–54, 2024.

[15] P. Kumar, V. K. S, P. L., and S. SenthilPandi, "Enhancing Face Mask Detection Using Data Augmentation Techniques," in Proc. Int. Conf. Recent Advances in Science and Engineering Technology (ICRASET), B 10.1109/ICRASET.2023.10420361. G Nagara, India, 2023, pp. 1–5. doi:

[16] X. Zhang, Y. Mao, Q. Yang, and X. Zhang, "A Plant Leaf Disease Image ... [Incomplete citation — needs completion]."

[17] H. Saeed, H. Malik, U. Bashir, A. Ahmad, S. Riaz, M. Ilyas, W. A. Bukhari, and M. I. Ali Khan, "Blockchain technology in healthcare: A systematic review," PLOS ONE, vol. 17, no. 4, p. e0266462, 2022. doi: 10.1371/journal.pone.0266462.

[18] M. Gallazzi, S. Biavaschi, A. Bulgheroni, T. M. Gatti, S. Corchs, and I. Gallo, "A Large Dataset to Enhance Skin Cancer Classification With Transformer-Based Deep Neural Networks," IEEE Access, 2024.

[19] M. A. Kassem, K. M. Hosny, and M. M. Fouad, "Skin Lesions Classification Into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning," IEEE Access, vol. 8, pp. 114822–114832, 2020.

[20] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," PLOS ONE, vol. 15, no. 12, p. e0243043, Dec. 2020. doi: 10.1371/journal.pone.0243043.