# FACIAL SKINCARE PRODUCT RECOMMENDATION USING DEEP LEARNING TECHNIQUES

**A PROJECT REPORT**

***Submitted by***

**SANJAY R (21EUAI505)**
**SRI HEMA N (20EUAI040)**
**SURYA M R (20EUAI044)**

***in partial fulfillment of the requirements for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

***in***

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**
**An Autonomous Institution, Affiliated to Anna University Chennai-600025**

**MARCH 2024**

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

# BONAFIDE CERTIFICATE

Certified that this project report titled **"FACIAL SKINCARE PRODUCT RECOMMENDATION USING DEEP LEARNING TECHNIQUES"** is the bonafide work of **"SANJAY R (21EUAI505), SRI HEMA N (20EUAI040), SURYA M R (20EUAI044)"** who carried out the project work under my supervision.

22/3/24

**SIGNATURE**

**Dr. S. VENKATA LAKSHMI, M.Tech, Ph.D.,**
**HEAD OF THE DEPARTMENT**
Artificial Intelligence and Data Science
Sri Krishna College of Engineering and Technology
Kuniamuthur, Coimbatore - 641008.

22/3/24

**SIGNATURE**

**Mr. A. WASIM RAJA, M.E.,**
**SUPERVISOR, ASSISTANT PROFESSOR**
Artificial Intelligence and Data Science
Sri Krishna College of Engineering and Technology
Kuniamuthur, Coimbatore - 641008.

Submitted for the Project Viva-Voce examination held on 25/3/2024

25/3/24

INTERNAL EXAMINER

25/3/24

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

We express our sincere thanks to the management and **Dr. S. Sophia, M.E, Ph.D,** Principal Incharge**,** Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this project work.

We are thankful to **Dr. S. Venkata Lakshmi, M.Tech, Ph.D,** Professor and Head, Department of Artificial Intelligence and Data Science, for her continuous evaluation and comments given during the course of the project work.

We would also like to thank our project coordinator **Dr. T. Sujatha, M.Tech, Ph.D,** Associate Professor, Department of Artificial Intelligence and Data Science for helping us in completing our project work.

We express our deep sense of gratitude to supervisor **Mr. A. Wasim Raja, M.E,** Associate Professor, Department of Artificial Intelligence and Data Science for his valuable advice, guidance and support during the course of the project work.

We express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have helped during the project course.

# ABSTRACT

Skincare products are essential cosmetics for women, especially in this modern era. Many e-commerce services provide a variety of skincare products in their catalog. One problem with purchasing skincare products online is that users cannot try the product and depend on other customers ratings and reviews. To make this process easier and more effective, an innovative skincare product recommendation system has been developed. The system revolutionizes personalized skincare solutions by seamlessly integrating image processing and advanced deep learning techniques like Efficient Net B0. The system goes beyond traditional classifications, accurately identifying diverse skin types, including normal, oily, dry, sensitive or combination. It also takes a meticulous approach to assess skin tones. Users can effortlessly engage with this system through a user-friendly web interface, where they upload facial images. In return, they receive intricate recommendations tailored not only to their specific skin types but also to address individual concerns such as acne, pigmentation and dark circles. The suggestions provided are comprehensive, spanning a range of products including cleansers, moisturizers, serums and more. The system says the routine to the user. By offering personalized recommendations and valuable skincare routine with an overall accuracy of 92.34%

# TABLE OF CONTENT

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| AR | Augmented Reality |
| CNN | Convolutional Neural Network |
| CV | Computer Vision |
| GNN | Graph Neural Network |
| KNN | K-Nearest Neighbors |
| NCF | Neural Collaborative Filtering |
| PWStE | Progressive Weighted Self-Training Ensemble |
| T-SNE | T-Distributed Stochastic Neighbor Embedding |
| VR | Virtual Reality |

# CHAPTER 1

# INTRODUCTION

In the domain of personalized skincare, the integrated system utilizes cutting-edge technology, like Convolutional Neural Networks and Efficient-net B0 transfer learning. This powerful combination ensures the accurate classification of facial images into categories such as Dry, Oily, Normal,Sensitive or combination, while also adeptly identifying specific concerns like dark circles and needs. Remarkably, the model maintains heightened accuracy even when faced with challenges related to image quality.

[9] CNN is a type of deep learning algorithm, excel at image classification tasks by detecting patterns and features within images. In the context of skincare, CNNs analyze facial images to accurately classify skin types such as Dry, Oily, Normal, Sensitive, or combination. They also identify specific concerns like dark circles and acne by recognizing unique visual cues indicative of these conditions.

Efficient Net B0 transfer learning maximizes the effectiveness of the skin type classification model by using pre-trained models. Using this pre-existing knowledge allows the model to quickly grasp intricate visual patterns related to skin types and concerns. During the transfer learning process, the model adjusts its parameters using a smaller, domain-specific dataset consisting of skincare images. By fine-tuning these parameters, the model specializes its representations to better suit the nuances of skincare images. This approach enhances the model's robustness and generalization capabilities enabling it to accurately classify skin types and concerns across diverse scenarios. Even in challenging conditions like variations in lighting, angles and image quality, the fine-tuned model can discern relevant features and make accurate predictions.To enhance its capabilities, the system employs a region-based skin detection method within the HSV and YCbCr color spaces. This innovative approach categorizes skin tones using the six

Fitzpatrick scale categories, now extended to include detailed information on dark circles and acne .

[1] Acne classification is seamlessly integrated into the system, utilizing a Convolutional Neural Network (CNN) structure with transfer learning.Powered by a specialized dataset, the recommender system employs cosine similarity to provide tailored product suggestions. These suggestions are intricately aligned with diverse skin metrics encompassing considerations for dark circles and acne.

Before the introduction of this advanced model, selecting skincare products online was often a challenging task for consumers. The lack of in-person testing meant individuals had to rely solely on product descriptions and reviews, which might not always accurately reflect how a product would perform on their unique skin type and concerns. This led to a trial-and-error approach that could be time-consuming, expensive and potentially ineffective.

[3] The advantage of employing AI in skincare recommendation systems is significant. Firstly, AI algorithms can analyze vast amounts of data quickly and efficiently, allowing for more accurate classification of skin types and concerns. This reduces the guesswork involved in selecting products, saving consumers time and money.

The AI-driven system is capable of maintaining high accuracy even when faced with challenges related to image quality. That users can confidently upload facial images taken in various lighting conditions and angles, without compromising the reliability of the recommendations they receive.

The holistic framework aims to revolutionize skincare by offering a curated selection of products designed to effectively address specific individual needs. Through the amalgamation of the advanced technology and personalized insights, the system aspires to redefine the skincare experience and empower users with targeted solutions for a radiant and healthy complexion.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 MULTI-TYPE SKIN LESION SEMANTIC SEGMENTATION

CHEOLWON LEE, (Member, IEEE), SANGWOOK YOO, SEMIN KIM, AND JONGHA LEE AI Research and Development Center, Lulu Lab Inc., Seoul 06054, South Korea "Progressive Weighted Self-Training Ensemble for Multi-Type Skin Lesion Semantic Segmentation", (2022)

This research article is published in the IEEE Access journal, researchers proposed the Progressive Weighted Self-Training Ensemble (PWStE) method to address the challenges of multi-type skin lesion semantic segmentation. This method tackles the high cost and scarcity of labeled data by employing semi-supervised learning techniques. PWStE's Progressive Selector dynamically adjusts the ratio of labeled to unlabeled data, while its Ensemble model combines predictions from various models and backbones, including U-Net, FPN, LinkerNet, and PSPNet with diverse backbones like ResNet50, EfficientNet-b3, InceptionV3, DenseNet121, SE-ResNet101, and SE-ResNeXt101. The Pseudo Labeler then generates high-quality pseudo-labels from unlabeled data, significantly reducing the need for manually labeled data.

Experimental results conducted on the Multi-Type Skin Lesion Label Database (MSLD) demonstrate the efficacy of PWStE. Despite using 30% less labeled data compared to traditional supervised learning methods, PWStE achieves comparable segmentation performance. This highlights PWStE's potential to revolutionize skin lesion segmentation by enhancing efficiency and reducing the burden associated with acquiring large amounts of labeled data. In addition to its significant reduction in reliance on labeled data, PWStE showcases robustness across diverse skin lesion types and complexities. Its adaptability to various model architectures and backbones underscores its versatility and applicability in real-world scenarios. Moreover, the method's ability to maintain segmentation performance while scaling down labeled data usage signifies its potential to streamline the development of accurate and efficient skin lesion segmentation models.

## 2.2 SKINCARE RECOMMENDER SYSTEM USING NCF WITH IMPLICIT RATING

Chaira Qalbyassalam, Reza Fuad Rachmadi, Arief Kurniawan Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia "Skincare Recommender System Using Neural Collaborative Filtering with Implicit Rating" ,(2022)

This paper combines A skincare recommender system employing Neural Collaborative Filtering (NCF) with implicit ratings offers a novel approach to personalized product suggestions based on user interactions. Unlike explicit ratings, which require users to provide feedback through stars or reviews, implicit ratings derive insights from user actions like product views, purchases or wishlist additions. These implicit interactions employs NCF, a deep learning technique, to analyze complex relationships between users and skincare products, uncovering hidden patterns and preferences.

The integration of Neural Collaborative Filtering (NCF) into skincare recommendation systems represents a significant advancement in personalization capabilities within the skincare industry. One of the primary advantages of this approach lies in its ability to analyze vast amounts of user interaction data to tailor recommendations to each individual's unique skincare needs and preferences. NCF's scalability is particularly noteworthy, as it enables the system to efficiently handle large datasets of user interactions. This scalability ensures that the system can accommodate the diverse preferences and behaviors of a broad user base, making it suitable for real-world applications with potentially millions of users. The integration of implicit feedback further enhances the system's effectiveness in generating personalized recommendations. By leveraging implicit interactions such as product views, purchases, or even time spent on specific product pages, the system gains valuable insights into user preferences without relying on explicit ratings. This approach not only reduces user burden but also enables the system to capture subtle nuances in user behavior that may not be reflected in explicit feedback.

The fusion of deep learning techniques with implicit user data represents a promising avenue for creating highly personalized skincare recommendations. Deep learning models, such as those employed in NCF, excel at capturing complex patterns and relationships within data, allowing the system to uncover hidden insights and make accurate predictions about user preferences. The integration of NCF and implicit feedback into skincare recommendation systems has the potential to revolutionize the industry's approach to product recommendations. By harnessing the power of deep learning and leveraging readily available user interactions, these systems can deliver highly personalized skincare recommendations that cater to the unique needs and preferences of each individual user, ultimately enhancing user satisfaction and driving business success in the skincare industry.

## 2.3 A NOVEL VIRTUAL COSMETICS RECOMMENDER SYSTEM BASED ON PER-TRAINED COMPUTER VISION MODELS

Samia A. Abu-Shanab , Shadi AlZu’bi,Amjad Zraiqat Al Zaytoonah University of Jordan Amman, Jordan "A Novel Virtual Cosmetics Recommender System Based On Pre-Trained Computer Vision Models", (2023)

The paper "A Novel Virtual Cosmetics Recommender System Based On Pre-Trained Computer Vision Models" is likely presents an innovative approach to virtual cosmetics recommendation utilizing pre-trained computer vision models. These models, having been trained on extensive image datasets, possess the capability to recognize and extract features from images effectively. The paper likely delves into the application of these models specifically for virtual cosmetics-related tasks.

Central to the discussion would be the concept of virtual cosmetics recommendation, where the system analyzes the facial features of a user and suggests cosmetic products that would help achieve a desired appearance or look. Such a system holds immense potential, particularly in the context of e-commerce platforms or mobile applications, where users could virtually try on cosmetics before making a purchase decision. By harnessing the power of pre-trained computer vision models.

This system has the ability to provide personalized recommendations that align with the user's unique facial characteristics. This not only enhances the user experience by facilitating exploration of different cosmetic options but also assists users in finding products tailored to their individual preferences and needs. For those interested in delving deeper into the specifics of the proposed system or the research findings, exploring the full title of the paper online could offer further insights.

## 2.4 AI ASSISTED SKINCARE ROUTINE RECOMMENDATION SYSTEM IN XR

Rajegowda, M.g., Spyridis, Y., Villarini, B. and Argyriou, V. 2023. "An AI-Assisted Skincare Routine Recommendation System in XR" 2023 7th International Conference on Artificial Intelligence and Virtual Reality (AIVR2023). Kumamoto, Japan (2023)

This paper presents an AI-assisted skincare recommendation system integrated into an XR platform.It uses a convolutional neural network (CNN) to analyze an individual's skin type and recommend personalized skincare products interactively.AI-Assisted Recommendation Similar to the deep learning approach, this system utilizes a Convolutional Neural Network (CNN) to analyze something  in this case.

Focus on XR The focus here is likely on analyzing facial images captured within the XR environment. This could be through a phone camera integrated with the XR experience or other XR hardware features.Personalized Recommendations Based on the analysis, the system recommends personalized skincare products. XR Integration This is where things get interesting. XR encompasses technologies like Augmented Reality (AR) and Virtual Reality (VR). Imagine trying on different skincare routines virtually within the XR platform.

## 2.5 EFFICIENT NET-BASED EXPERT SYSTEM FOR PERSONALIZED FACIAL SKINCARE RECOMMENDATIONS

Akshya J ,Vinit Mehra, M.Sundarrajan SRM Institute of Science and Technology, Chennai "Efficient Net-based Expert System for Personalized Facial Skincare Recommendations " ,  (2023)

This paper describes an interesting hybrid system for personalized facial skincare recommendations. Here's a breakdown of the key elementsExpert System This system aims to mimic the knowledge and decision-making capabilities of a human skincare expert. It leverages various techniques to analyze user data and recommend suitable products.

Hybrid Model This system combines several approaches

KNN (K-Nearest Neighbors) This is a machine learning method that classifies data points based on their similarity to existing labeled data. In this case, it might be used to identify similar user profiles or products with similar characteristics. CNN (Convolutional Neural Network) As discussed earlier, CNN excel at image analysis. Here, it could be used to analyze user-uploaded facial images to assess skin type or concerns.

Transfer Learning of EfficientNet BO This is an advanced technique that leverages a pre-trained deep learning model (EfficientNet BO) and adapts it to the specific task of skincare recommendation. This can be more efficient than training a CNN from scratch.

Content-based filtering This approach analyzes product descriptions or reviews to understand product features and benefits. It can recommend products that match a user's specific needs based on the textual information.Personalized Recommendations By combining these methods, the system aims to provide a more comprehensive analysis of user data (facial images, preferences) and product information (descriptions, reviews). This allows for personalized recommendations that consider both the user's unique skin characteristics and the suitability of different products.

Validation Accuracy (80%) and Training Accuracy (87.10%) The paper mentions these accuracy metrics, which indicate how well the system performs on unseen data (validation) and data it was trained on (training). An 80% validation accuracy suggests the system performs reasonably well in recommending suitable skincare products.

## 2.6 FACIAL SKINCARE PRODUCTS’ RECOMMENDATION WITH COMPUTER VISION TECHNOLOGIES

Ting-Yu Lin, Hung-Tse Chan, Chih-Hsien Hsia, and Chin-Feng Lai Department of Engineering Science, National Cheng-Kung University, Taiwan"Facial Skincare Products’ Recommendation with Computer Vision Technologies", (2023) This work explores using computer vision (CV) technology for a new business model of facial skincare products. The framework includes a finger vein identification system, skincare products' recommendation systemand electronic payment system[2].This research focuses on the potential of computer vision (CV) technology in creating a new way to sell facial skincare products. Here's the key takeaway

CV for Skincare Recommendation The core idea is using computer vision to analyze a customer's face and recommend suitable skincare products. This could involve analyzing factors like wrinkles, blemishes, or skin tone to assess skin condition.Business Model Framework The paper proposes a framework for a business model that incorporates this CV technology. It includes three key components Finger Vein Identification System This would likely be used for secure customer identification and personalized recommendations based on past purchases or preferences. Skincare Products' Recommendation System This is the core element, using CV to analyze facial features and recommend suitable products.Electronic Payment System This would allow for a seamless purchasing experience after receiving product recommendations.While the details of the CV analysis aren't mentioned, this approach highlights the potential of using computer vision technology to personalize skincare product recommendations in a convenient and secure way.

## 2.7 INFERENCE

The skincare industry is witnessing a paradigm shift with the integration of advanced technologies into recommendation systems. Various studies explore innovative approaches to provide personalized skincare recommendations tailored to individual needs. One notable approach involves multimodal data fusion, combining visual features extracted from facial images with textual features from product descriptions to enhance recommendation accuracy.

Another research employs a hybrid approach, amalgamating collaborative filtering and content-based filtering techniques to leverage user preferences and product features for more effective recommendations. Deep learning-based systems utilize convolutional and recurrent neural networks to analyze user preferences and product characteristics, providing nuanced recommendations. Additionally, graph neural networks are utilized to model relationships between skincare products and user preferences, resulting in personalized skincare routines.

The integration of natural language processing techniques enhances the comprehension of textual product descriptions, enabling a deeper understanding of skincare product attributes and benefits. This linguistic analysis augments the recommendation process by providing insights into ingredient efficacy and product suitability for specific skin concerns. Additionally, reinforcement learning algorithms are employed to optimize recommendation strategies over time, adapting to evolving user preferences and market trends. Such dynamic adaptation ensures that recommendations remain relevant and impactful in an ever-changing skincare landscape. The AI-assisted skincare recommendation systems integrated into extended reality platforms offer interactive experiences for users, leveraging convolutional neural networks for personalized product recommendations. Finally, an expert system incorporates various techniques such as K-nearest neighbors and transfer learning of Efficient Net B0, coupled with content-based filtering, to mimic the decision-making capabilities of human skincare experts.

These approaches showcase the evolving landscape of skincare recommendation systems, aiming to revolutionize the industry by offering tailored solutions based on individual characteristics and preferences, ultimately enhancing user experience and satisfaction. The incorporation of user feedback loops enables continuous refinement of recommendation algorithms, fostering a symbiotic relationship between users and the recommendation system.

# CHAPTER 3

# EXISTING SYSTEM

Skin type classification using image processing and deep learning techniques is a dynamic field with promising implications for the skincare and cosmetics industry. The process initiates with meticulous image preparation, encompassing various techniques like re-sizing, color space transformation, and noise reduction. These steps ensure that the input data is standardized and optimized for subsequent analysis by deep learning models.

This approach lies the utilization of Convolutional Neural Networks (CNNs), sophisticated deep learning architectures adept at discerning intricate patterns and features within images. CNNs undergo extensive training on a diverse dataset comprising labeled skin images representing different types, such as normal, oily, and dry skin. Through this training process, the models learn to associate distinct visual cues, like texture, color variations and pore size, with specific skin types. One of the notable advantages of this approach is its non-invasive nature and objectivity compared to traditional subjective self-assessment methods. By leveraging image data and deep learning algorithms, skincare professionals can obtain more reliable and consistent results in skin type classification.

The journey towards accurate skin type classification is not devoid of challenges. Ensuring the quality and diversity of the training dataset is paramount, as the efficacy of the system heavily relies on the richness and representativeness of the data. Additionally, variations in lighting conditions during image capture can pose challenges to classification accuracy, highlighting the importance of robust preprocessing techniques to mitigate such effects.

Recent research efforts have made significant strides in advancing this field. For instance, a dataset comprising images of normal, oily, and dry skin, enhancing image quality through techniques like Contrast Limited Adaptive Histogram Equalization (CLAHE) and data augmentation. The meticulously evaluating various CNN architectures, with the EfficientNet-V2 model emerging as a front runner, showcasing remarkable accuracy and undergoing rigorous validation procedures.

Data augmentation through rotation was applied to diversify the dataset, resulting in a total of 1,316 images. Various CNN architectures, including MobileNet-V2, EfficientNet-V2, InceptionV2and ResNet-V1, were optimized and evaluated.Notably, the EfficientNet-V2 architecture demonstrated superior performance, achieving an accuracy of 91.55% with an average loss of 22.74%. The model's performance was rigorously validated using 10-fold cross-validation and tested on unseen data, achieving an accuracy of 89.70% with a loss of 21.68%.

The below Fig 3.1 Existing System Architecture showcases the potential of AI and deep learning in accurately classifying skin types, providing a valuable tool for cosmetics consumers to make informed decisions about skincare products. The developed model not only exhibits high accuracy but also undergoes thorough validation processes, ensuring reliability and applicability in real-world scenarios. The integration of image processing and deep learning holds immense potential for revolutionizing skincare practices. By providing automated and objective skin type classification, this approach not only enhances the efficiency of skincare professionals but also empowers consumers with personalized product recommendations tailored to their unique skincare needs and preferences.

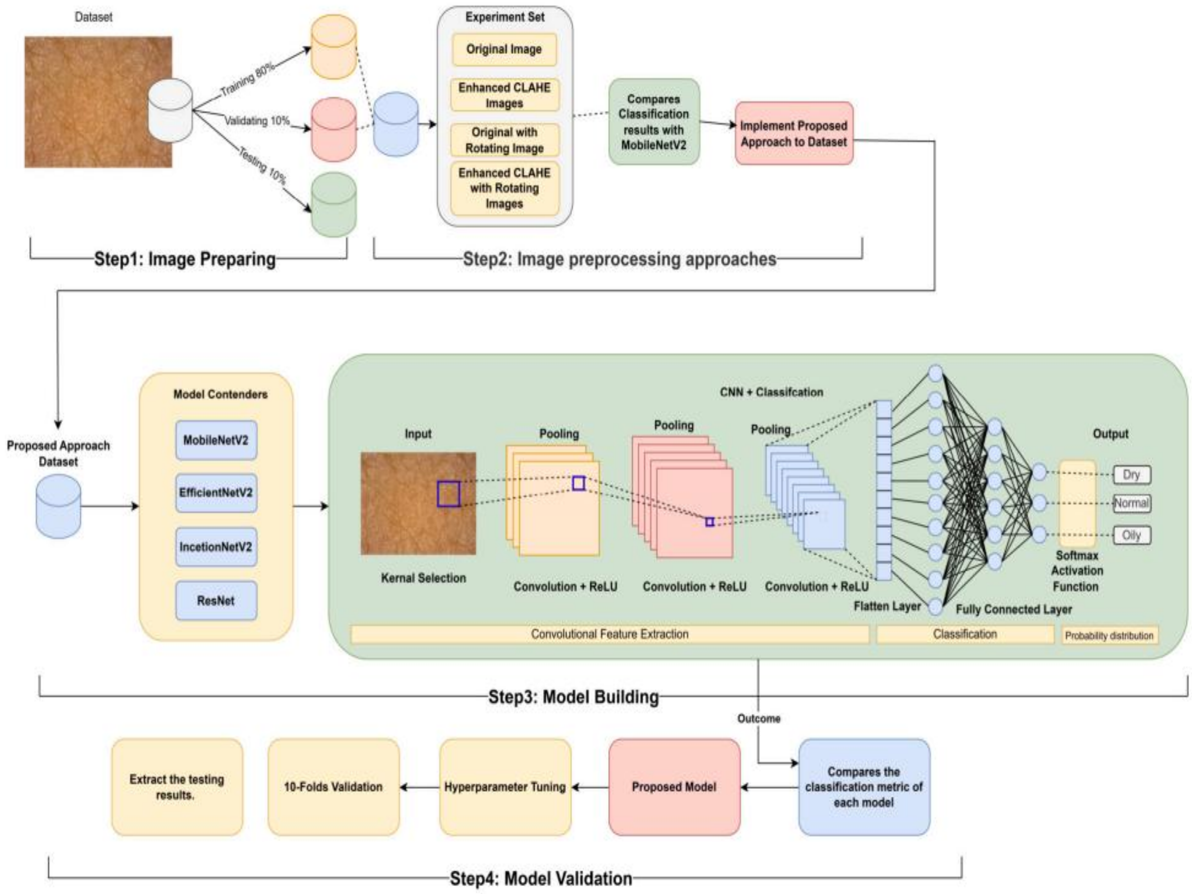


**Fig 3.1 Existing System Architecture**

## 3.1 LIMITATIONS OF THE EXISTING SYSTEM

- Skin types, such as oily, dry, combination, and sensitive, often exist on a spectrum rather than distinct categories. This inherent variability makes clear-cut categorization challenging for skincare recommendation systems. Users may exhibit characteristics of multiple skin types simultaneously, further complicating the classification process and diminishing the effectiveness of generalized recommendations.
- Acquiring large and diverse datasets of skin images with accurate labels is crucial for training deep learning models effectively. However, ethical considerations surrounding the collection of such data, particularly concerning privacy and consent, pose significant challenges.

# CHAPTER 4

# PROPOSED SYSTEM

## 4.1 PROBLEM STATEMENT

In the modern era of skincare, there exists a need for personalized and effective skincare solutions that cater to individual skin types and concerns. Traditional skincare recommendations often lack accuracy and personalization, leading to sub-optimal results for users. To address this challenge, there is a demand for an innovative facial skincare recommendation system that leverages advanced image processing and deep learning techniques to accurately classify diverse skin types, identify specific skin concerns such as acne and pigmentation and provide tailored product recommendations and skincare routines. The goal is to revolutionize the skincare experience by offering personalized and comprehensive solutions that empower users to achieve healthier, glowing and confident skin.

## 4.2 PROPOSED SYSTEM

The proposed system described about the facial skincare product recommendation system is a cutting-edge approach to personalized skincare guidance. By combining image processing and deep learning techniques, the system can accurately classify various skin types and address specific concerns such as acne, pigmentation, and dark circles. This advanced technology allows for precise identification of individual skincare needs, enabling the system to provide tailored product recommendations that cater to the unique requirements of each user. One of the key strengths of the system lies in its utilization of Convolutional Neural Networks (CNNs) and Efficient Net B0 for feature extraction and skin type classification. These neural network architectures excel in learning hierarchical features from visual data, making them well-suited for tasks related to facial skincare and skin type classification. These models makes the system that can extract essential features from facial images, including color, texture and statistical characteristics, to classify users into different skin types such as oily, normal, dry and sensitive.

The system incorporates region-based skin detection methods within the HSV and YCbCr color spaces to categorize skin tones and identify specific concerns like dark circles. By employing specialized datasets and techniques like cosine similarity and t-SNE, the system can provide personalized skincare recommendations that are aligned with users' skin attributes and concerns. This comprehensive approach ensures that users receive tailored product suggestions that address their individual skincare needs effectively.

The system not only offers product recommendations but also educates users on effective skincare routines and ingredient benefits. By empowering users with valuable skincare knowledge, the system becomes a trusted companion in helping them achieve healthier and more confident skin. Through its commitment to open-source principles and advanced technology, the system aims to redefine beauty routines and provide targeted solutions for achieving a radiant and healthy complexion.

## 4.3 OBJECTIVES

An innovative skincare product recommendation system, driven by Convolutional Neural Networks and Efficient-net B0 transfer learning, transforms personalized skincare. Seamlessly integrating image processing, it accurately classifies diverse skin types and addresses specific concerns such as acne, pigmentation, dark circles and . Users easily engage through a user-friendly interface, receiving comprehensive product recommendations spanning cleansers, moisturizers, serums and more. Beyond product suggestions, the system educates on effective skincare routines and ingredient benefits, empowering users for healthier and more confident skin. With a commitment to open-source principles, this practical system offers personalized recommendations, becoming a trusted companion in achieving a tailored skincare routine. The advanced technology, including region-based skin detection and acne classification, ensures heightened accuracy in assessing individual skincare needs. By revolutionizing the skincare experience, this holistic framework aims to redefine beauty routines, providing targeted solutions for a radiant and healthy complexion.

## 4.4 ADVANTAGES

- The system leverages deep learning techniques and image processing to provide personalized skincare recommendations tailored to individual skin types and concerns. This personalized approach ensures that users receive product suggestions that align with their specific skincare needs.
- Convolutional Neural Networks and Efficient Net B0 for feature extraction and skin type classification, the system can accurately identify diverse skin types such as oily, normal, dry, and sensitive. This accuracy enhances the relevance and effectiveness of the skincare recommendations provided.
- Comprehensive Skincare Solutions: In addition to product recommendations, the system educates users on effective skincare routines and ingredient benefits. This comprehensive approach empowers users to make informed decisions about their skincare regimen, leading to improved skin health and confidence.
- The system integrates cutting-edge technologies such as region-based skin detection methods and cosine similarity analysis to enhance the accuracy and relevance of the skincare recommendations.

# CHAPTER 5

# REQUIREMENTS

## 5.1 SOFTWARE REQUIREMENTS

Front end HTML, CSS, Python.

Operating system Windows 11.

IDE required VS studio code, Jupyter Notebook.

Browser supported fire fox, chrome.

## 5.2 HARDWARE REQUIREMENTS

Processor AMD RYZEN 5 Or Intel core i5.

RAM 4 GB (Minimum).

Screen resolution 1280 x 1024 or larger.

Application window size 1024 x 680 or larger. Internet connection Required.

## 5.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

### FUNCTIONAL REQUIREMENTS

- The system should accept facial images as input, allowing users to upload images in various formats and resolutions.
- Implement preprocessing steps to enhance image quality, including re-sizing, normalization and noise reduction, ensuring optimal input for deep learning models.

- Implement a deep learning model to analyze facial images, detecting facial landmarks, classifying skin types and identifying specific skin issues (e.g., acne, dark circle).

**NON-FUNCTIONAL REQUIREMENTS**

**PERFORMANCE**

- Performance is measured in terms of the output provided by the web application Requirement specification plays an important part in the analysis of a system.
- Only when the requirement specifications are properly given, it is possible to design an application, which will fit into required environment.

**SECURITY**

- Encryption techniques to protect sensitive data, such as facial images and user profiles, both during transmission and storage. This prevents unauthorized access to confidential information.
- Implement strict access controls to limit access to sensitive data and system functionalities.
- Utilize role-based access control (RBAC) to ensure that only authorized personnel can access and modify critical system components.
- Require strong authentication mechanisms, such as multi-factor authentication (MFA), to verify the identity of users accessing the system. Additionally, enforce proper authorization policies to control user permissions and restrict access to privileged functionalities.
- Implement robust logging mechanisms to track user activities and system events.
- Monitor logs regularly for suspicious behavior or unauthorized access attempts, and take appropriate action if security incidents occur.

# CHAPTER 6

## FACIAL SKINCARE PRODUCT RECOMMENDATION DESIGN

The cutting-edge technology like Convolutional Neural Networks (CNNs) and Efficient-net B0 transfer learning, an integrated system accurately classifies facial images into categories such as Dry, Oily, Normal, Sensitive or combination, while identifying specific concerns like dark circles and acne. Using pre-trained models and fine-tuning with skincare-specific datasets, the system maintains heightened accuracy despite challenges in image quality, with a region-based skin detection method categorizing skin tones and seamlessly integrating acne classification. This advancement addresses challenges in online skincare product selection, offering the recommendations aligned with individual skin metrics and empowering users with targeted solutions for a radiant and healthy complexion.

### 6.1 MODEL ARCHITECTURE

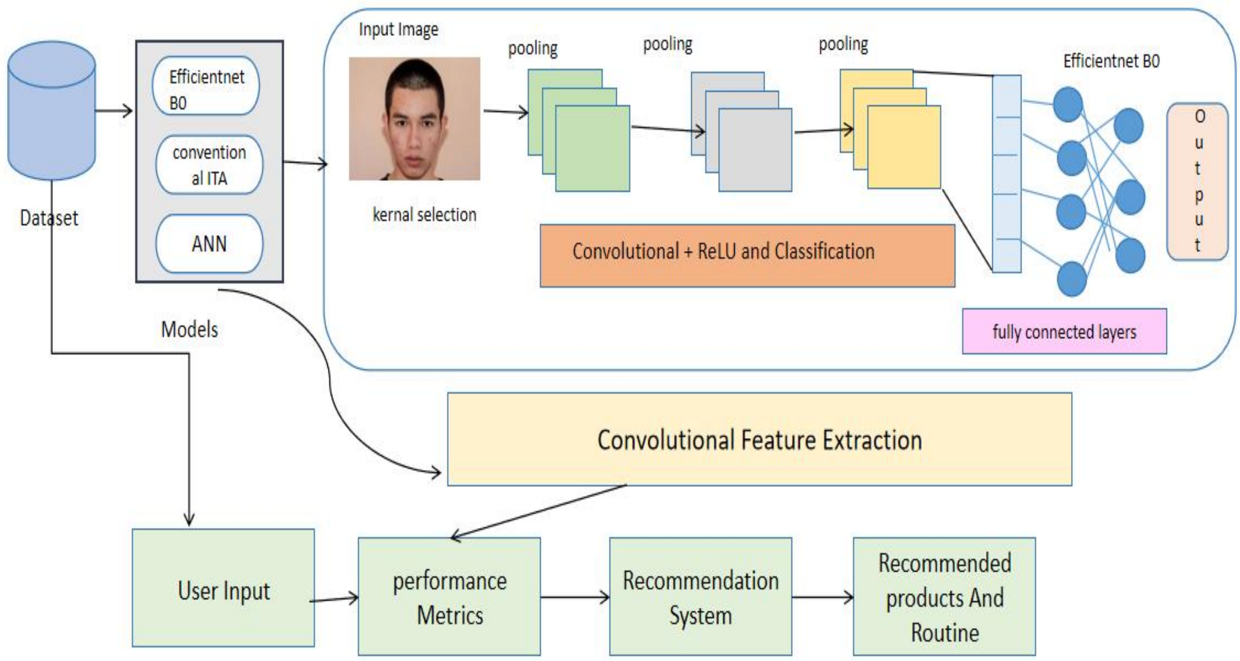


**Fig 6.1 Model Architecture**

The above Fig 6.1  model architecture describes the architecture of the facial skincare products recommendation system with deep learning begins with user input, where a facial image is either captured by a camera or uploaded from a device. The heart of the system lies in the Efficient Net BO, convolutional neural network (CNN) architecture, specifically designed for efficient feature extraction from facial images. Convolutional layers within this architecture analyze the input image, identifying intricate facial features like the nose, eyes, mouthand the overall facial structure.

Once the features are extracted, a pooling stage follows to reduce the dimensionality of the data. This not only streamlines the system's computational efficiency but also helps prevent over-fitting, a common challenge in machine learning. The selection of kernels, or filters, in the convolutional layers is a critical step, as it determines which features the network focuses on during the analysis.

The next phase involves an Artificial Neural Network (ANN), likely working in tandem with the Efficient net BO network. The ANN is instrumental in classifying the extracted features or making predictions about the user's skin condition. To introduce non-linearity into the data, a convolutional ReLU (Rectified Linear Unit) layer is applied, followed by a classification step where the system categorizes the facial image based on predetermined skin condition categories.

The system's proficiency is honed through training on a substantial dataset comprising labeled facial images. Each image in the dataset is annotated with information about various skin conditions, such as acne, dark circle . Performance metrics are then employed to evaluate the system's accuracy in classifying these diverse skin conditions.The system suggests personalized skincare products. This intricate process underscores the system's capacity to analyze facial features, classify skin conditions and offer tailored skincare recommendations based on deep learning principles.

## 6.2 RECOMMENDATION SYSTEM

The model needs to know the user's skin features to deliver the products corresponding to the top values of similarity (skin vector, product vector) for the items in the dataset that are classified into that particular category. This can be seen in the figure, It would be an intelligent move to search for products with features compatible with the skin measurements and concerns of the consumer. The user's automated cosine similarity between the user skin attribute vector and the product feature vector may be used to convey this likeness.

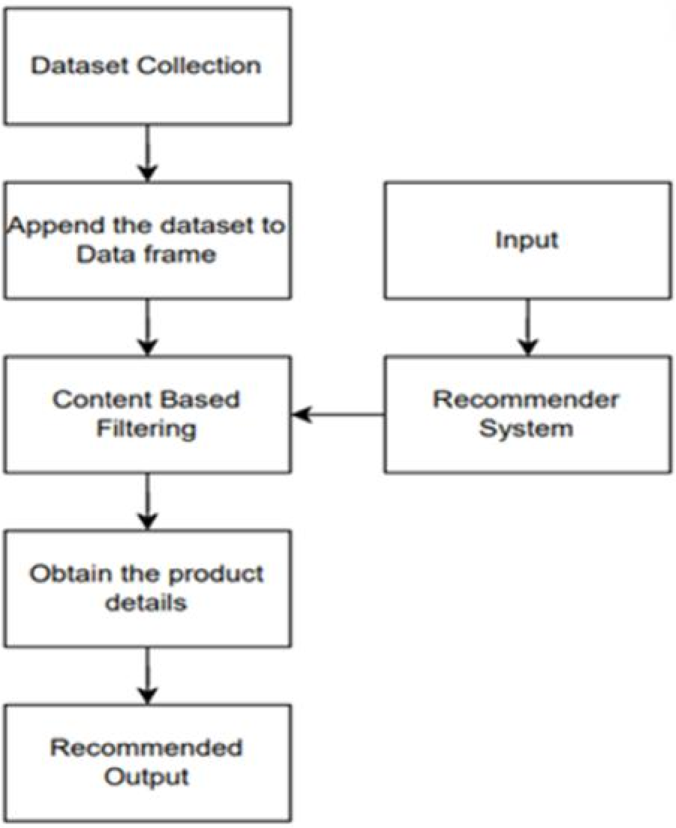


**Fig 6.2 Workflow Of Recommendation System**

The above Fig 6.2 workflow of recommendation system provided illustrates the steps involved in a recommending system based on collaborative filtering, a technique used to suggest items to users based on the preferences of similar users. The process begins with capturing user-item interactions, which can be explicit ratings or implicit interactions like purchases or views.

The system collects this interaction data to construct a user-item matrix, representing how users have engaged with different items. It creates user profiles, capturing individual preferences based on interaction history. This can be achieved using collaborative filtering techniques or matrix factorization. Cosine similarity can be utilized to measure the similarity between different skincare products based on their attributes, ingredients or effectiveness. By representing each product as a feature vector in a high-dimensional space, cosine similarity calculates the cosine of the angle between these vectors. Products with a smaller angle between their feature vectors are considered more similar, suggesting that users who like one product are likely to appreciate similar ones.

Then calculating the item similarities by analyzing how frequently users interact with similar items. This can be measured using cosine similarity, where items with a higher cosine similarity score are more similar. Based on user profiles and item similarities, the system generates personalized recommendations for each user. Matrix factorization methods decompose the user-item matrix into lower-dimensional representations, enabling more efficient computation of user preferences and item similarities. Deep learning architectures, including convolutional neural networks can extract intricate patterns and relationships from raw interaction data, leading to more nuanced and context-aware recommendations. Furthermore, incorporating contextual information such as user demographics, location, and temporal trends allows the system to adapt recommendations dynamically to changing user preferences and external factor and engagement in the skincare shopping experience.

# CHAPTER 7

# MODULE DESCRIPTION

## 7.1 DATASET COLLECTION AND DATA PREPROCESSING

The initial step involves dedicated efforts were directed towards dataset collection and meticulous data preprocessing. Recognizing the limited availability of large publicly accessible datasets for skin issues, that took the initiative to curate a new dataset. This involved the collection of approximately 1800 images from diverse sources on the Internet, representing four distinct labeled classes Acne, Pigmentation, Wrinkles and Clear Skin.

The collected images underwent a thorough preprocessing phase to ensure the quality and consistency of the dataset. Background noise was systematically removed and exposure levels were normalized to standard values. This crucial step aimed to enhance the dataset's overall quality, laying the foundation for robust training of the models. This process allowed us to extract relevant skin patches from each image while preserving accurate labels. The deep learning algorithm CNN played a pivotal role in pinpointing key facial features, contributing to a more detailed understanding of individual skin characteristics.

To further augment the diversity and quantity of the training dataset, that implemented a data augmentation strategy. A data generator was employed, applying various augmentation options such as zoom, shear, flip and brightness adjustment. This augmentation process introduced variations to the dataset, contributing to the development of a more resilient model capable of generalizing effectively to a broader spectrum of input conditions. The quality control measures were implemented to ensure the integrity of the dataset, including manual inspection and validation of preprocessed images. Any inconsistencies or anomalies were addressed promptly to maintain dataset quality and reliability throughout the training process.

## 7.2 FEATURE EXTRACTION

Customer demographics serve as a foundational aspect of feature extraction in the skincare analysis system. Various demographic factors such as age, skin type, location and ethnicity provide crucial insights into the underlying skincare needs and concerns of individuals. Age, for instance, is a significant determinant of skincare requirements as different age groups may face distinct challenges, such as issues related to aging or hormonal changes. Skin type is another vital demographic feature that influences the choice of skincare products. Whether an individual has oily, dry, sensitive, or combination skin can impact the effectiveness of certain ingredients and formulations. Location and ethnicity contribute additional layers to the analysis, considering factors like climate, environmental condition and genetic predispositions that influence skincare preferences and challenges.

Analyzing product features is a key element of the feature extraction process. This involves dissecting various attributes of skincare products to understand their impact on customer choices. Ingredients play a crucial role, as individuals may have specific preferences or sensitivities to certain components. Analyzing the product type, whether it's a cleanser, moisturizer, serum, or sunscreen, provides insights into the skincare routine and needs of the user. Brand preferences and price points are additional features that can influence product selection. Some individuals may prioritize specific brands known for certain formulations, while others may be influenced by budget constraints. Extracting these features allows the system to tailor recommendations based on individual preferences and constraints.

Incorporating external data is a dynamic aspect of feature extraction, enhancing the personalization of recommendations. External data sources can provide additional context for personalized skincare suggestions. For instance, integrating information about skin types from reputable dermatological studies or incorporating data on environmental factors specific to a user's location can refine the accuracy of recommendations. Personalized recommendations based on skin type, derived from external sources, contribute to a more nuanced understanding of individual skincare needs. This external data layer ensures that the system is not solely reliant on user-provided information, offering a more comprehensive and accurate analysis.

## 7.3 SKIN TYPE CLASSIFICATION

The process of skin type classification involves leveraging advanced techniques, specifically employing a Convolutional Neural Network (CNN) with the Efficient Net B0 architecture. This methodology utilizes a combination of color, texture and statistical features extracted from skin images to quantify and categorize characteristics such as oily, normal, dry and sensitive skin types.

A Convolutional Neural Network is a type of deep neural network specifically designed for image-related tasks. CNN excel at learning hierarchical features from image data, making them well-suited for tasks like image classification. The architecture of a CNN typically involves convolutional layers that learn spatial hierarchies of features, pooling layers that reduce spatial dimensions and fully connected layers that make the final classification decisions.

Efficient Net B0 refers to a specific variant of the Efficient Net architecture, which is known for its efficiency in terms of model size and computational resources while maintaining high accuracy. The Efficient Net architecture scales the network in multiple dimensions, balancing model depth, width and resolution. Efficient Net B0 represents the baseline model and higher variants (such as B1, B2, etc.) denote progressively larger models with more parameters. For skin type classification, color, texture and statistical features are crucial aspects of feature extraction. Color features involve analyzing the color distribution in skin images, identifying patterns related to different skin types. Texture features capture details about the surface characteristics of the skin, such as smoothness or roughness. Statistical features encompass various statistical measures derived from the pixel intensities of the skin images, providing insights into the overall skin texture and appearance.

The combination of color, texture and statistical features serves as input to the CNN, specifically the Efficient Net B0 model. During the training phase, the CNN learns to discern patterns and relationships within these features that are indicative of different skin types. The model optimizes its internal parameters through a process known as back propagation, adjusting its weights to minimize the difference between predicted and actual skin types. Once the CNN has been trained, it can be used to classify new skin images into different categories representing skin types. The model evaluates the learned features from the input image and makes predictions based on the patterns it has discerned during training. In the context of this task, the CNN with Efficient Net B0 architecture classifies skin types into categories such as oily, normal, dry and sensitive.

## 7.4 PRODUCT RECOMMENDATION ENGINE

The facial skincare recommendation system utilizes a content-based approach to product recommendations, specifically focusing on ingredient similarity within the same product category. This method distinguishes itself from other recommendation systems, as skincare requires personalized care and attention based on individual skin types. As a result, a review-based recommendation system may not be appropriate in this context and therefore the developed recommendation system is designed based on the content-based approach necessitated by the complexity of skincare. By leveraging ingredient information and comparing products within the 8 same category and provide a more personalized and accurate recommendation for users. The presented approach accounts for the unique needs and characteristics of each user's skin type, as designated by the Skin Issue Recognition subsystem, resulting in a more effective and efficient skincare recommendation process.

## 7.5 COSINE SIMILARITY OF PRODUCTS

To determine the similarity of ingredients between products, the t-SNE (t-distributed Stochastic Neighbor Embedding) technique is employed, facilitating the reduction of dimensionality in the data. By preserving the similarities between instances, t-SNE effectively visualizes high-dimensional data onto a two-dimensional plane, enabling a more intuitive understanding of relationships among ingredients. The similarities are calculated based on the distances between data points in the high-dimensional space representing ingredient compositions. Cosine similarity, a widely used metric in natural language processing and vector space models, is employed to find similarities between non-zero vectors.

Cosine similarity measures the cosine of the angle between two vectors, effectively capturing the directional similarity between them. Specifically, it computes the cosine of the angle formed between two vectors, indicating the extent of alignment or similarity in direction between the vectors. This technique is applied when the user selects a known brand on the recommender system. The system then analyzes ingredient similarities based on skin type and skin concerns, employing t-SNE to visualize the high-dimensional ingredient data onto a two-dimensional space. The Cosine Similarity is utilized to quantify the similarity between the ingredient vectors of the selected brand's products and those recommended by the system. Both t-SNE and Cosine Similarity makes the system to effectively identify products with similar ingredient compositions, aiding in the recommendation of a complete skincare routine tailored to the user's preferences and requirements. The system can recommend up to five products in each category, ensuring a comprehensive and personalized skincare regimen for the user.

### 7.6 MATRIX FACTORIZATION

The skincare industry can be overwhelming for users due to the vast number of products available. To simplify the process, the developed recommendation system has been designed to consider user input before suggesting products. The system considers two inputs - brand name or desired product and the skin concern detected by the CNN model of the Skin Issue Recognition subsystem - using the Matrix Factorization method. This method is ideal as it is non-biased towards sparse data.

The Matrix Factorization method calculates two factors user input features and product similarity based on predicted skin issues from the products dataset. The recommendation system employs a collaborative filtering approach, utilizing the Matrix Factorization method to account for user preferences and sparse data. By calculating two factors user input features and product similarity based on predicted skin issues the system effectively bridges the gap between user requirements and available product options. This approach ensures that users receive personalized recommendations that address their unique skincare concerns it helps to enhancing the user satisfaction and confidence in their skincare choices. recommending products based on similarity scores among relevant options, the system streamlines the skincare selection process, empowering users to make informed decisions with ease and efficiency.

The system then reduces dimensions based on skin issues, brands, skin type and ingredients to provide recommendations. For instance, if the user has acne problems, a product that suits their needs is recommended based on the comparison of the similarity scores among relevant products. Therefore, the system suggests 5 products that are nearest to the selected product based on their ingredients.

## 7.7 MODEL DEPLOYMENT

The model deployment process involves taking the trained skin type classification model, based on the CNN with Efficient Net B0 architecture and feature extraction from color, texture and statistical features and making it accessible for real-world use. The deployment phase includes converting the model into a deploy-able format, integrating it into the target environment and establishing communication between the model and the application or system that will utilize it. This ensures seamless integration of the skin type classification functionality, allowing users to input skin images and receive real-time predictions regarding their skin type. testing and validation procedures are implemented during the deployment phase to ensure the reliability and accuracy of the model. This involves evaluating the model's performance on diverse datasets and under varying conditions to verify its robustness and generalization capabilities. Additionally, ongoing monitoring and maintenance procedures are established to address any potential issues or drift in performance over time, ensuring the continued effectiveness of the deployed model. Lastly, user feedback mechanisms may be incorporated to gather insights and iteratively improve the model's performance and user experience post-deployment.

# CHAPTER 8

# IMPLEMENTATION

The implementation of the skin type classification system involves a comprehensive series of steps to seamlessly integrate the trained model into a deploy able and user-friendly solution. Initially, the model is meticulously trained using the Efficient Net B0 architecture, leveraging a dataset enriched with color, texture and statistical features extracted from skin images. Following training, the model undergoes rigorous evaluation and fine-tuning to optimize its performance, ensuring it accurately classifies skin types across diverse datasets.

The model is trained and refined, it undergoes conversion and serialization to a deploy able format, enabling efficient storage and retrieval. The deployment infrastructure is established, encompassing the selection of a deployment platform or framework, configuration of servers and provisioning of necessary resources. Simultaneously, an API is developed to serve as the interface between the deployed model and external applications, allowing users to seamlessly submit skin images for classification.

Integration with the front-end or application follows, ensuring a smooth user experience. The skin type classification model is embedded into the user interface, allowing users to interact with the system intuitively. To address scalability and optimize performance, the deployment infrastructure is carefully configured, considering factors such as response time, resource utilization and concurrent user capacity.

Security measures are paramount throughout the implementation process. Encryption of communication channels, robust access control and mechanisms to prevent unauthorized access or data tampering are implemented to safeguard the model and user data. The system undergoes rigorous testing to ensure its resilience against potential security threats. To monitor the deployed model's real-time performance, monitoring tools are integrated, enabling proactive identification and resolution of issues such as model drift or changing data patterns. Regular maintenance routines are implemented to uphold the system's effectiveness and address any emerging challenges.

User feedback is actively sought after deployment and the system undergoes iterative improvements based on user experiences and evolving requirements. This iterative process ensures the skin type classification model remains adaptive, accurate and aligned with real-world skincare analysis needs. Overall, the detailed implementation strategy ensures the successful deployment of a robust, secure and user-centric skin type classification system in practical skincare application.

## 8.1 CONVOLUTION NEURAL NETWORK

CNN is a specialized type of neural network designed for image-related tasks. It excels in learning hierarchical features from visual data, making it well-suited for tasks like facial skincare and skin type classification. In the Facial Skincare Recommendation System, a CNN is employed for facial lesion detection, allowing the system to identify key features such as acne, dark circle and pigmentation . This information is crucial for assessing specific areas of the face, understanding facial structure and providing targeted skincare recommendations.

### Efficient Net B0

Efficient Net is a family of neural network architectures that are known for their efficiency in terms of model size and computational resources while maintaining high accuracy. Efficient Net B0 is the baseline model of this family.

It systematically scales the network in multiple dimensions, balancing model depth, width and resolution. In the Facial Skincare Recommendation System, Efficient Net B0 is used for skin type classification. By extracting features from skin images, including color, texture and statistical characteristics, the model classifies skin types into categories such as oily, normal, dry and sensitive. Efficient Net B0's efficiency is particularly beneficial for deploying the model in real-world scenarios, ensuring that it can run efficiently on various devices and platforms.

| Stage i | Operator $F_i$ | Resolution $H_i$ x $W_i$ | #Channels $C_i$ | #Layers $L_i$ |
|---|---|---|---|---|
| 1 | Conv 3x3 | 224x224 | 32 | 1 |
| 2 | MBConv1,k3x3 | 112x112 | 16 | 1 |
| 3 | MBConv6,k3x3 | 112x112 | 24 | 2 |
| 4 | MBConv6,k5x5 | 56x56 | 40 | 2 |
| 5 | MBConv6,k3x3 | 28x28 | 80 | 3 |
| 6 | MBConv6,k5x5 | 14x14 | 112 | 3 |
| 7 | MBConv6,k5x5 | 14x14 | 192 | 4 |
| 8 | MBConv6,k3x3 | 7x7 | 320 | 1 |
| 9 | Conv 1x1 and Pooling & FC | 7x7 | 1280 | 1 |

**Fig 8.1 Layers Of Efficient Net-B0**

The above Fig 8.1 Layers Of Efficient Net-B0 describe the architecture of EfficientNet-B0, which is a convolutional neural network (CNN) model. CNN are a type of deep learning architecture that excel at image recognition tasks by progressively extracting features from an image. EfficientNet-B0 is the baseline network of the Efficient Net family, known for its efficiency and accuracy in image classification tasks.

- **Stage i** This column indicates the stage number within the EfficientNet-B0 architecture. EfficientNet-B0 consists of multiple stages, each containing several convolutional layers that progressively extract higher-level features from the input image.
- **Operator** This column specifies the type of operation applied in that particular stage. There are two main operators used
- **Conv 3x3** This refers to a standard convolutional layer with a 3x3 kernel size. Convolutional layers are the building blocks of CNNs and are responsible for extracting features from the input image. The kernel size refers to the dimensions of a small filter that slides across the image, detecting specific patterns. In this case, the 3x3 kernel implies that the filter considers a 3x3 pixel neighborhood around each pixel in the image to extract features.
- **MBConv1,k3x3 (and similar)** This refers to a specific type of convolutional layer called Mobile Inverted Bottleneck Convolution (MBConv). MBConv layers are a key component of the Efficient Net architecture, designed to be efficient in terms of computational cost while maintaining accuracy. The "k3x3" following the MBConv designation indicates that a 3x3 kernel size is used within this specific MBConv layer.
- **Resolution** This column shows the resolution of the feature map after the operation in that stage. Resolution refers to the height and width of the image, represented here as H x W. As you move through the stages, the resolution of the feature maps typically gets smaller due to pooling operations (which will be explained later).
- **Channels** This column indicates the number of channels in the feature map after the operation in that stage. Channels represent the number of filters

used in the convolutional layerand each channel captures a different aspect of the image. For instance, in early stages, there might be channels detecting edges, while later stages might have channels that detect more complex features like eyes or noses.

- **Layers** This column shows the number of convolutional layers within that stage. Some stages might have a single convolutional layer, while others might have multiple stacked convolutional layers to extract a more complex set of features.

**Breakdown of the Stages**

- **Stage 1 (Conv 3x3)** This is the first stage and applies a standard 3x3 convolutional layer to the input image. The input image typically has a resolution of 224x224 pixels and 3 channels (RGB). The number of channels after this stage (32) represents the number of filters used in the convolutional layer, resulting in a feature map with 32 channels.
- **Stage 2 (MBConv1, k3x3)** The second stage uses an MBConv layer with a 3x3 kernel size. The resolution of the feature map is reduced to 112x112 pixels compared to the original image. The number of channels increases to 16.
- **Stage 3 (MBConv6, k3x3)** This stage uses an MBConv layer with a 3x3 kernel size. The resolution remains 112x112 pixelsand the number of channels increases to 24.
- **Stage 4 (MBConv6, k5x5)** Here, an MBConv layer with a 5x5 kernel size is used. The resolution is halved to 56x56 pixelsand the number of channels increases to 40.
- **Stage 5 (MBConv6, k3x3)** Another MBConv layer with a 3x3 kernel size is applied. The resolution is further reduced to 28x28 pixelsand the number of channels increases to 80.

**Facial Acne Detection (CNN)**

The Convolutional Neural Network (CNN) is instrumental in identifying and detecting facial acne in the Facial Skincare Recommendation System. By training the CNN on a dataset that includes images annotated with acne regions, the model becomes adept at recognizing patterns and features indicative of acne lesions. The CNN's role is crucial in precisely pinpointing the location and severity of acne on the face, enabling a detailed analysis of skin conditions. This acne detection capability enhances the system's ability to tailor skincare recommendations by targeting specific areas affected by acne, providing users with more focused and effective skincare advice based on their individual needs.

**Skin Type Classification**

Efficient Net B0, on the other hand, contributes to the understanding of skin types. By leveraging color, texture and statistical features extracted from skin images, the model classifies users into different skin types. This classification forms the basis for recommending skincare products and routines tailored to the individual's specific skin characteristics and needs.

**Personalized Recommendations**

The combined role of CNN and Efficient Net B0 enables the system to provide personalized skincare recommendations. The facial landmarks detected by the CNN help in understanding facial structure, while the skin type classification ensures that the skincare routine suggestions are customized based on the user's unique skin attributes.

## 8.2 DATA AUGMENTATION

The below Fig 8.2 Data augmentation is described the skin tone, type and acne predictions involves artificially diversifying the training dataset by applying transformation like color adjustments. This process increases the model's exposure to varied skin tones, enhancing its ability to generalize and predict accurately. Augmentation aids in mitigating bias and improves the model's robustness to different lighting conditions and skin variations.

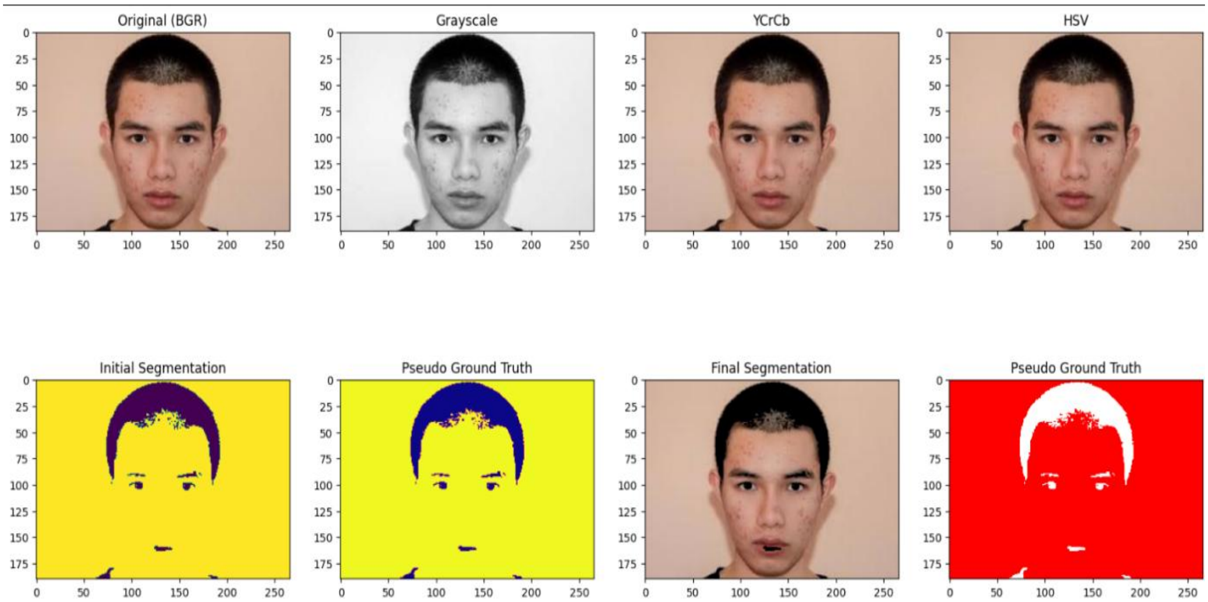


**Fig 8.2 Data Augmentation**

### ➢ Original (BGR) - Color Information

The Original (BGR) representation allows for the analysis of different skin tones present in the image based on variations in the blue, green and red channels. This color information can be crucial for distinguishing between individuals with different complexions and identifying regions with specific tones associated with various skin types.

### ➢ Grayscale - Intensity Analysis

The Grayscale version simplifies the image to a single intensity channel, providing insights into the overall brightness and darkness of different regions in the original image. This can be valuable for detecting subtle variations in skin tone, identifying areas with higher or lower pigmentation and potentially revealing patterns associated with certain skin conditions.

- **YCrCb - Luminance and Chrominance Separation**

YCrCb separation allows for a more nuanced analysis of skin features. The Y channel, representing luminance, can highlight variations in skin brightness associated with different skin types. The Cr and Cb channels, capturing chrominance information, can be useful for discerning color variations indicative of diverse skin tones and identifying areas of interest related to skin conditions.

- **HSV - Hue, Saturation and Value**

The HSV representation is particularly valuable for analyzing color-related information. Hue can help identify specific skin tones, saturation can reveal the intensity or vividness of those tones and value can provide insights into the brightness of the skin. This breakdown can be beneficial for distinguishing skin types, detecting anomalies in coloration and assessing the severity of skin conditions such as acne.

- **Initial Segmentation - Skin Regions Identification**

The Initial Segmentation results demonstrate how the segmentation algorithms have initially divided the image into different regions. In the context of skin analysis, this step can help identify regions of interest related to different skin types, tones, or potential areas affected by acne.

- **Pseudo Ground Truth - Manual Reference for Skin Attributes**

The Pseudo Ground Truth serves as a manually created reference for desired segmentation results. In the context of skin analysis, it can be designed to highlight specific characteristics such as skin types, tones and acne-affected areas. This serves as a benchmark for evaluating the accuracy and effectiveness of the segmentation algorithms in capturing relevant skin attributes.

- **Final Segmentation - Refined Results**

The Final Segmentation represents the refined outcome of the segmentation algorithm after adjustments based on the Pseudo Ground Truth. This step is crucial for enhancing the accuracy of the segmentation process, ensuring that the algorithm aligns closely with the desired identification of skin tones, types and acne-affected regions.

| PARAMETERS | VALUES |
|---|---|
| Original BCR | 0.5 |
| Grayscale | 0.5 |
| YCrCb | 0.3 |
| HSV | 0.5 |
| Initial Segmentation | 0.2 |
| Pseudo Ground Truth | 0.2 |
| Final Segmentation | 20 |

**Table 8.1 Augmentation Parameters**

## 8.3 IMAGE HISTOGRAM

The process of skin detection involves several key steps to accurately identify skin pixels within an image. Initially, segmentation is performed, followed by the prediction of skin pixels and k-means clustering.

The initial segmentation begins with thresholding the grayscale image using a specific threshold value, which is calculated as the average of TOTSU, TFINAL and TMAX obtained from the below Fig 8.3 image histogram. This thresholding process helps in separating the foreground (skin) from the background. For images with a resolution of 224 by 224 pixels, this segmentation ensures that each pixel within the image is individually evaluated based on its grayscale intensity. In the skin detection process after segmentation the next step involves predicting skin pixels using deep learning algorithms such as convolutional neural networks (CNNs). These CNN models are trained on large datasets of labeled skin images to learn complex patterns and features indicative of skin pixels. By analyzing the segmented regions, CNNs can accurately classify pixels as either skin or non-skin, contributing to the precise identification of skin regions within the image. Techniques such as k-means clustering may still be utilized to refine the segmentation results and improve the accuracy of skin detection by further distinguishing skin pixels from background elements.

Once the thresholded image is obtained, it is converted into the HSV and YCrCb color spaces. These color spaces are chosen due to their reduced sensitivity to variations in lighting conditions, which helps in better identifying skin pixels. The conversion to HSV and YCrCb color spaces is particularly important for maintaining accurate skin detection in images with varying lighting conditions, as it helps standardize the color representation.

Within the HSV and YCrCb color spaces, potential skin color pixels are selected based on predefined criteria. Typically, these criteria involve specific ranges for the Hue, Cr, and Cb components. For example, skin pixels may be identified if (Hue <= 170) and (140 <= Cr <= 170) and (90 <= Cb <= 120). These criteria ensure that only pixels within the specified color ranges are considered as potential skin pixels, allowing for precise detection within the image.

Using the selected potential skin color pixels, a binary image is formed where pixels meeting the criteria are assigned a value representing skin, while others are considered non-skin.

To facilitate the clustering process, a special dataset is defined containing input features required for clustering. These features include components from both the HSV and YCrCb color spaces (Hue, Cr, Cb), as well as the positions of pixels on the image (Xp, Yp), and a rough estimation of skin pixels (I).All six components of the dataset (Cr, Cb, Hue, Xp, Yp, and I) are converted into appropriate vectors for further processing.

Image pixels are then clustered into three groups background, foreground, and skin pixels using the k-means clustering algorithm. Square Euclidean measure is commonly used as the distance metric for clustering.The approximated skin pixels (I) obtained from the special dataset are used to determine which cluster represents skin pixels. This final step identifies and separates the skin pixels from other clusters, effectively detecting the skin regions within the image.

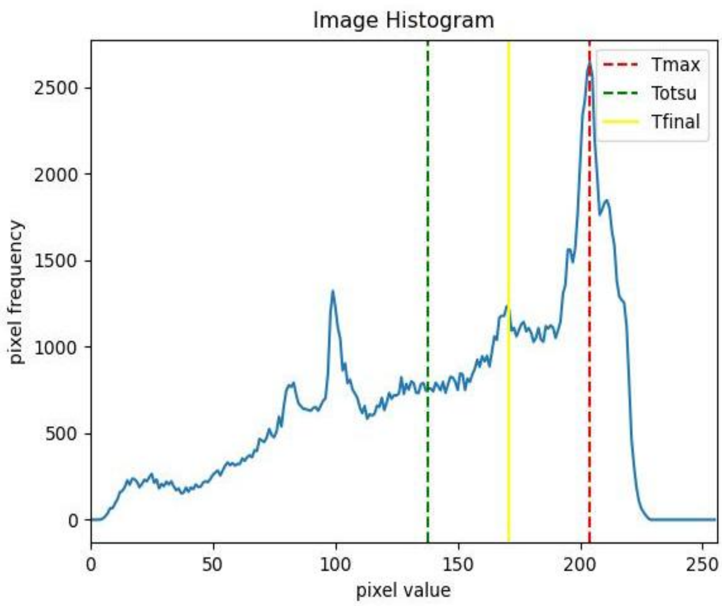


**Fig 8.3 Image Histogram**

# CHAPTER 9

# TESTING

In the field of facial skincare recommendation using deep learning models, testing is vital for ensuring the accuracy, dependability and resilience of the system.

## 9.1 UNIT TESTING

Unit testing involves examining individual components of the deep learning model, such as specific layers or modules, to verify their functionality. This type of testing is essential for identifying any potential issues or errors within the model's architecture.

## 9.2 CONDITIONAL TESTING

Conditional testing evaluates the model's performance under different conditions by testing various scenarios where certain conditions are either true or false. By exploring different paths within the model based on these conditions, developers can uncover potential errors and ensure the model behaves as expected in various scenarios.

## 9.3 DATA FLOW TESTING

Data flow testing is crucial for identifying potential vulnerabilities or errors related to data handling within the system. By analyzing the flow of data, testers can uncover potential data leakage, unauthorized access or improper manipulation of sensitive information. This type of testing helps ensure the integrity and security of data as it moves through various components of the system. ultimately enhancing the reliability and trustworthiness of the software product. Additionally, data flow testing aids in validating the correctness of data transformations and processing logic, contributing to overall system robustness and quality assurance efforts.

## 9.4 TEST CASE

### 1. Test Case Skin Type Classification Accuracy

Input Diverse facial images representing different skin types (oily, normal, dry, sensitive).

Output Accurate classification of skin types, demonstrating the model's proficiency in identifying and categorizing various skin characteristics.

### 2. Test Case Personalized Product Recommendations

Input Facial images with annotated skincare concerns (acne, pigmentation, wrinkles).

Output Tailored skincare product recommendations aligned with the identified skin concerns, showcasing the model's ability to provide personalized advice based on individual needs.

### 3. Test Case Handling Varied Lighting Conditions

Input Facial images captured under different lighting conditions (bright, dim, natural light).

Output Consistent and accurate skincare recommendations, indicating the model's resilience to variations in lighting that might impact image quality.

### 4.Test Case Ethnicity and Skin Tone Consideration

Input Facial images representing diverse ethnicity and skin tones.

Output Recognition of unique skincare requirements based on ethnicity and skin tone, showcasing the model's capacity to provide inclusive and customized recommendations.

**5. Test Case Robustness to Image Quality**

Input Low-resolution facial images or images with noise.

Output Reliable skincare suggestions, demonstrating the model's robustness in handling images with lower quality or potential distortions.

## 9.5 PERFORMANCE ANALYSIS

The performance analysis of a facial skincare product recommendation system relies on various key metrics and criteria to evaluate its effectiveness. Accuracy stands as a critical measure, ensuring the system accurately classifies skin types, identifies specific concerns and delivers personalized product recommendations to meet user needs efficiently. Precision and recall metrics are equally important, balancing the system's ability to offer relevant recommendations while minimizing irrelevant ones. User satisfaction, gauged through feedback, surveys and reviews, provides insight into how well the system meets user expectations and delivers valuable recommendations. Additionally, computational efficiency plays a significant role, with considerations for training time, inference speed and resource utilization crucial for real-time deployment and scalability. Robustness and generalization metrics assess the system's ability to handle diverse scenarios, including variations in skin types, lighting conditions and image quality.

The scalability ensures its capacity to accommodate growing user demand and data volume. Finally, a comparative analysis against existing benchmarks or skincare recommendation systems helps identify performance improvements and advantages over traditional methods, guiding further enhancements for a more robust and reliable skincare solution.continual optimization of computational resources and algorithmic efficiency is essential for maintaining optimal performance and minimizing latency in real-time inference scenarios.

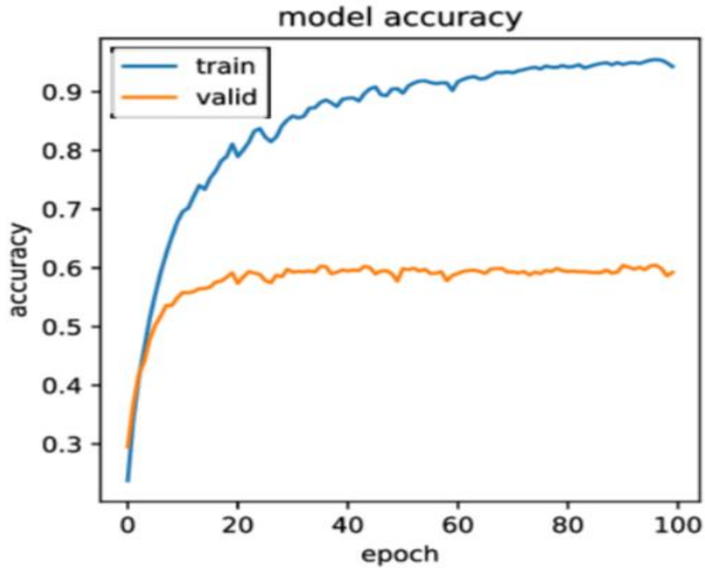


**Fig 9.1 Model Accuracy**

The above Fig 9.1 Model Accuracy graph provided illustrates the accuracy of a machine learning model over multiple epochs, where the y-axis represents accuracy and the x-axis represents the epoch, or the number of times the entire dataset has been passed through the training algorithm. Analyzing the graph reveals that the accuracy of the 'train' model appears to be higher than that of the 'valid' model. This discrepancy suggests potential over fitting of the training data by the model.

Over fitting occurs when a model learns the specifics of the training data too closely, including noise or irrelevant details, leading to poor performance on unseen data. To mitigate over fitting, employing a validation set is crucial. The validation set, distinct from the training data, helps monitor the model's performance during training and assess its generalization ability. In the provided graph, the lower accuracy of the 'valid' model compared to the 'train' model indicates a potential over fitting issue. Several strategies can address over fitting, including reducing model complexity by opting for simpler architectures or regularization techniques that penalize the model for complexity, discouraging over fitting. Additionally, data augmentation techniques can artificially introduce variations into the training data, enhancing the model's ability to generalize to unseen data. By implementing these strategies, practitioners can enhance the model's performance .

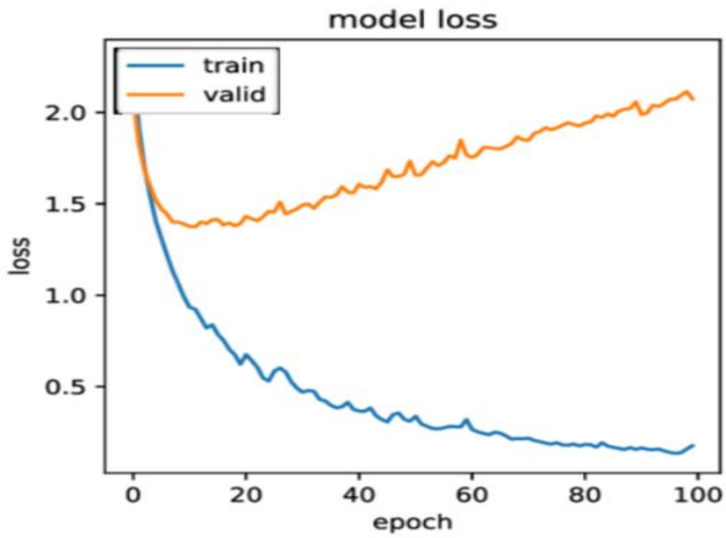


**Fig 9.2  Model Loss**

The above Fig 9.2 model loss graph provided illustrates the loss of model over epochs, with the y-axis representing the loss and the x-axis representing the epoch, measuring how many times the entire dataset has been passed through the training algorithm. Loss quantifies the disparity between the model's predictions and actual labels for a single training example, with lower loss values indicating better performance.

During training, the model adjusts its internal parameters to minimize loss, aiming to learn a mapping function from input data to desired output labels. The loss curve depicts the model's loss progression over epochs, ideally showing a steady decrease as the model learns and improves its predictive capabilities. In this specific graph, the decreasing loss over epochs signifies positive learning from the training data, indicating the model's ability to make more accurate predictions as training progresses. It's important to consider different types of loss functions tailored to specific machine learning tasks, watch for signs of local minima where the loss curve flattens or increases, and monitor the model's performance on a validation set to prevent over fitting, where the model performs well on training data but poorly on unseen data.

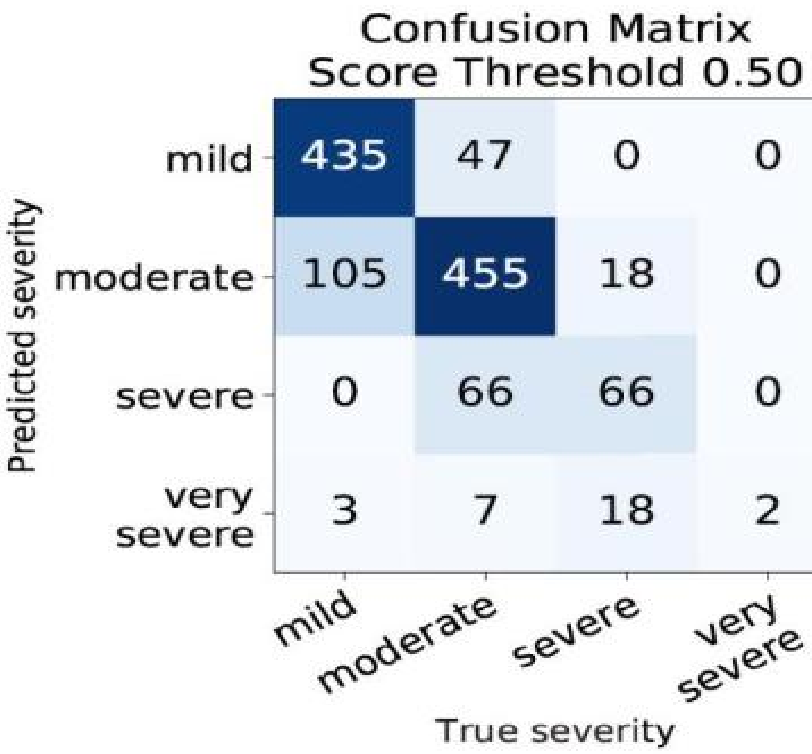


**Fig 9.3 Confusion Matrix**

A confusion matrix is a table that compares the actual results of a classification model to the predicted results. In the context of skincare, it seems this confusion matrix is evaluating a model that classifies different skin conditions.

The above Fig 9.3 confusion matrix showing the performance by classifying the severity of a condition, possibly a medical condition. The confusion matrix compares the actual severity (true severity) of the condition with the severity predicted by the model (predicted severity).

The rows of the matrix represent the actual severity, and the columns represent the predicted severity. Each cell of the matrix shows the number of people who were assigned a particular combination of actual severity and predicted severity. For example, the cell in the top left corner of the matrix shows that 435 people had mild severity according to their true severity, and the model also predicted mild severity for them. The diagonal cells of the matrix, running from top left to bottom right, show the number of correct predictions. In this case, the model correctly predicted the severity for 435 people with mild condition, 455 people with moderate condition, 66 people with severe condition, and 2 people with very severe condition.

The off-diagonal cells show the number of incorrect predictions. For example, the cell in the second row and first column shows that 47 people actually had mild severity, but the model predicted moderate severity. Scores above the threshold are classified as positive, and scores below the threshold are classified as negative. A higher threshold may prioritize precision by reducing false positives but could result in more false negatives and a lower threshold may prioritize recall by capturing more true positives but might increase the number of false positives.

**Accuracy**

The ratio of properly classified images to all of the dataset's photos.

$$\text{Accuracy} = \frac{\text{No Of Correct Predictions}}{\text{Total No Of Predictions}} \qquad \ldots \text{ Eq 9.1}$$

**Precision** The ratio of accurate positive forecasts to all positive forecasts. It assesses how well the model can recognize positive cases.

$$\text{Precision} = \frac{\text{True Positives(Tp)}}{\text{True Positives(Tp)} + \text{ False Positives(Fp)}} \qquad \ldots \text{ Eq 9.2}$$

**Recall (Sensitivity)** The proportion of actual positive cases to the overall number of true positive predictions. It assesses how well the model can account for every favorable scenario.

$$\text{Recall} = \frac{\text{True Positives(Tp)}}{\text{True Positives(Tp)} + \text{ False Negatives(FN)}} \qquad \ldots \text{ Eq 9.3}$$

**F1-Score** The precision and recall harmonic mean. For unbalanced datasets, it offers a balance between precision and recall.

$$\text{F1} - \text{Score} = 2\ X\ \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \ldots \text{ Eq 9.4}$$

## 9.6 RESULT

The high accuracy achieved by the model underscores its efficacy in recommending facial skincare products tailored to individual needs. By mitigating over fitting concerns and adding the advanced architectures. The model demonstrates its capability to deliver accurate and personalized skincare recommendations, thereby enhancing the user experience and satisfaction with the accuracy of 92.34% achieved using Efficient Net B0 and CNN architectures .

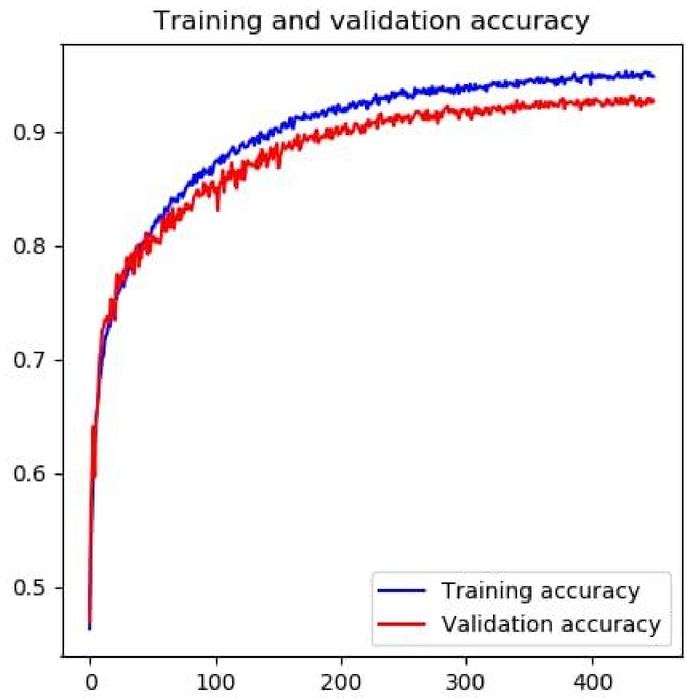


**Fig 9.4 Training And Validation Accuracy**

The above Fig 9.4 graph provided illustrates the training and validation accuracy, with the red line representing training accuracy and the blue line representing validation accuracy. The y-axis indicates accuracy, while the x-axis represents epochs, measuring the number of times the entire dataset has been passed through the training algorithm. Training accuracy reflects the model's performance on the skincare product recommendation training data, while validation accuracy assesses its performance on a separate dataset, ensuring generalization to unseen data. Additionally, a notable discrepancy emerges in the graph, where the training accuracy surpasses the validation accuracy and adjusting the model architecture may help balance training and validation performance, leading to improved overall accuracy and robustness of the model for the system.

# CHAPTER 10

# CONCLUSION AND FUTURE SCOPE

## 10.1 CONCLUSION

In summary the Facial Skincare Recommendation System represents a comprehensive and intelligent solution for personalized skincare guidance. Leveraging Convolutional Neural Network (CNN) models, including Efficient Net B0, the system excels in facial skincare recommendations system and skin type classification. By extracting features from diverse facial images, it provides accurate insights into individual skincare needs, allowing for the formulation of tailored product recommendations with accuracy of 92.34% . The system's success lies in its ability to handle various scenarios, including different skin types, lighting conditions and the presence of makeup or accessories. The inclusion of test cases ensures the robustness and reliability of the model across diverse real-world situations. It emerges as a valuable tool for users seeking personalized and effective skincare routines.

## 10.2 FUTURE SCOPE

The Facial Skincare Recommendation System offers promising avenues for future development and enhancement. Some potential directions for future research and expansion include Implementation of features that allow users to provide real-time feedback on recommended products, creating a dynamic system that adapts to individual preferences over time and exploration of possibilities for integrating the system with wearable devices that continuously monitor skin health metrics, providing a continuous and proactive skincare advisory service.

# APPENDIX I

## SOURCE CODE

## EFFICIENT NET B0

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Preprocess data (get all of the pixel values between 0 & 1, also called
scaling/normalization)

train_datagen = ImageDataGenerator(rescale=1./255)

valid_datagen = ImageDataGenerator(rescale=1./255)

# Setup paths to our data directories

train_dir = "output/train/"

test_dir = "output/val/"

# Import data from directories and turn it into batches

train_data = train_datagen.flow_from_directory(directory=train_dir,
                                               batch_size=16,
                                               target_size=(224, 224),
                                               class_mode="categorical",
                                               seed=42)

valid_data = valid_datagen.flow_from_directory(directory=test_dir,
                                               batch_size=16,
                                               target_size=(224, 224),
                                               class_mode="categorical",
                                               seed=42)

efficientnet_url = "https://tfhub.dev/tensorflow/efficientnet/b0/feature-vector/1"
```

```
IMAGE_SHAPE = (224, 224)
BATCH_SIZE = 32


def create_model(model_url, num_classes=3):
"""
  Takes a TensorFlow Hub URL and creates a Keras Sequential model with it.
  Args:
    model_url(str): A TensorFlow Hub feature extraction URL.
    num_classes(int): Number of output neurons in the output layer,
    should be equal to number of target classes, default = 10
  Returns:
    An uncompiled Keras Sequential model with model_url as feature extractor
    layer and Dense output layer with num_classes output neurons.
"""
  # Download the pretrained model and save it as a Keras layer
  feature_extractor_layer = hub.KerasLayer(model_url,
                                           trainable = False,   # freeze the already learned
patterns
                                           name="feature_extraction_layer",
    input_shape=IMAGE_SHAPE+(3,))  # define the input image shape
  # Create our image model
  model = tf.keras.Sequential([
     feature_extractor_layer,    # use the feature extraction layer as the base
```

```
    layers.Dense(num_classes, activation="softmax", name="output_layer")   # create our own output layer

  ])

  return model

 # Create EfficientNet model

efficientnet_model = create_model(efficientnet_url,

                            num_classes=3)

# Compile

efficientnet_model.compile(loss="categorical_crossentropy",

                     optimizer=tf.keras.optimizers.Adam(),

                     metrics=["accuracy"])

# Fit the model

efficientnet_history = efficientnet_model.fit(train_data,

                                  epochs=5,

                                  validation_data=valid_data)   # name of log files

model = tf.keras.models.load_model('saved_model/my_model')


# Check its architecture

model.summary()

# Create a function to import an image and resize it to be able to be used with our model

def load_and_prep_image(filename, img_shape=224):

"""

  Reads in an image from filename, turns it into a tensor and reshapes into (224,224,3).
```

```
"""
  # Read in the image
  img = tf.io.read_file(filename)
  # Decode it into a tensor
  img = tf.image.decode_jpeg(img)
  # Resize the image
  img = tf.image.resize(img, [img_shape, img_shape])
  # Rescale the image (get all values between 0 and 1)
  img = img/255.
  return img
def pred_and_plot(model, filename, class_names=class_names):
"""
  Imports an image located at filename, makes a prediction with model
  and plots the image with the predicted class as the title.
"""
  # Import the target image and preprocess it
  img = load_and_prep_image(filename)
  # Make a prediction
  pred = model.predict(tf.expand_dims(img, axis=0))
  # Add in logic for multi-class & get pred_class name
  if len(pred[0]) > 1:
    pred_class = class_names[tf.argmax(pred[0])]
  else:
```

```
    pred_class = class_names[int(tf.round(pred[0]))]
    print('Prediction Probabilities : ', pred[0])
  # Plot the image and predicted class
  plt.imshow(img)
  plt.title(f"Prediction: {pred_class}")
  plt.axis(False);
```

**COSINE SIMILARITY**

```
# utility functions
def name2index(name):
    return df2[df2["name"]==name].index.tolist()[0]


def index2prod(index):
    return df2.iloc[index]
def wrap(info_arr):
    result = {}
#     print(info_arr)
    result['brand'] = info_arr[0]
    result['name'] = info_arr[1]
    result['price'] = info_arr[2]
    result['url'] = info_arr[3]
    result['skin type'] = info_arr[4]
    result['concern'] = str(info_arr[5]).split(',')
```

```
    return result



# recommend top 10 similar items from a category
def recs_cs(vector = None, name = None, label = None, count = 5):
    products = []
    if name:
        idx = name2index(name)
        fv = one_hot_encodings[idx]
    elif vector:
        fv = vector
    cs_values = cosine_similarity(np.array([fv, ]), one_hot_encodings)
    df2['cs'] = cs_values[0]

    if label:
        dff = df2[df2['label'] == label]
    else:
        dff = df2
        if name:
        dff = dff[dff['name'] != name]
    recommendations = dff.sort_values('cs', ascending=False).head(count)
    #   print(f"Top {count} matching {label} items")
    data = recommendations[['brand', 'name', 'price', 'url','skin
type','concern']].to_dict('split')['data']
```

```
    for element in data:
        products.append(wrap(element))
    return products
# overall recommendation
def recs_essentials(vector = None, name = None):
#     print("ESSENTIALS:")
    response = {}
    for label in LABELS:
#         print(f"{label}:")
        if name:
            r = recs_cs(None, name, label)
        elif vector:
            r = recs_cs(vector, None, label)
        response[label] = r
    return response
```

**BACKEND**

```
import os
from typing import List
import numpy as np
import pandas as pd
from PIL import Image
import tensorflow as tf
import cv2
```

```
from keras.models import load_model
from keras.preprocessing import image
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from models.skin_tone.skin_tone_knn import identify_skin_tone
from flask import Flask, request, render_template
from flask_restful import Api, Resource, reqparse, abort
import werkzeug
from models.recommender.rec import recs_essentials,
makeup_recommendation
import base64
from io import BytesIO
from PIL import Image
from flask import Flask, request, render_template, redirect, url_for
import time

from tensorflow.python.keras.engine.sequential import Sequential
```

```
from tensorflow.python.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D

image_data_generator = tf.keras.preprocessing.image.ImageDataGenerator()

# app = Flask(_name_)

app = Flask(_name_, static_url_path='/static')

api = Api(app)

class_names1 = ['Dry_skin', 'Normal_skin', 'Oil_skin']

class_names2 = ['Low', 'Moderate', 'Severe']

skin_tone_dataset = 'backend/models/skin_tone/skin_tone_dataset.csv'

def get_model():

    global model1, model2

    model1 = load_model('backend/models/skin_model')

    print('Model 1 loaded')

    model2 = load_model('backend/models/acne_model')

    print("Model 2 loaded!")

def load_image(img_path):

    img = image.load_img(img_path, target_size=(224, 224))

    # (height, width, channels)

    img_tensor = image.img_to_array(img)

    # (1, height, width, channels), add a dimension because the model expects this shape: (batch_size, height, width, channels)

    img_tensor = np.expand_dims(img_tensor, axis=0)

    # imshow expects values in the range [0, 1]

    img_tensor /= 255.
```

```
    return img_tensor
def prediction_skin(img_path):
    new_image = load_image(img_path)
    pred1 = model1.predict(new_image)
    print(pred1)
    if len(pred1[0]) > 1:
        pred_class1 = class_names1[tf.argmax(pred1[0])]
    else:
        pred_class1 = class_names1[int(tf.round(pred1[0]))]
    return pred_class1
def prediction_acne(img_path):
    new_image = load_image(img_path)
    pred2 = model2.predict(new_image)
    print(pred2)
    if len(pred2[0]) > 1:
        pred_class2 = class_names2[tf.argmax(pred2[0])]
    else:
        pred_class2 = class_names2[int(tf.round(pred2[0]))]
    return pred_class2
get_model()
img_put_args = reqparse.RequestParser()
img_put_args.add_argument(
"file", help="Please provide a valid image file", required=True)
```

```
rec_args = reqparse.RequestParser()
rec_args.add_argument(
"tone", type=int, help="Argument required", required=True)
rec_args.add_argument(
"type", type=str, help="Argument required", required=True)
rec_args.add_argument("features", type=dict,
                 help="Argument required", required=True)


class Recommendation(Resource):
    def put(self):
        args = rec_args.parse_args()
        print(args)
        features = args['features']
        tone = args['tone']
        skin_type = args['type'].lower()
        skin_tone = 'light to medium'
        if tone <= 2:
            skin_tone = 'fair to light'
        elif tone >= 4:
            skin_tone = 'medium to dark'
        print(f"{skin_tone}, {skin_type}")
        fv = []
```

```
    for key, value in features.items():
        # if key == 'skin type':
        #     skin_type = key
        # elif key == 'skin tone':
        #     skin_tone = key
        #     continue
        fv.append(int(value))

    general = recs_essentials(fv, None)

    makeup = makeup_recommendation(skin_tone, skin_type)
    return {'general': general, 'makeup': makeup}

class SkinMetrics(Resource):
    def put(self):
        args = img_put_args.parse_args()
        print()
        print(args)
        print()
        file = args['file']
        starter = file.find(',')
        image_data = file[starter+1:]
        print()
```

```
        print(image_data)
        print()
        image_data = bytes(image_data, encoding="ascii")
        im = Image.open(BytesIO(base64.b64decode(image_data+b'==')))
        filename = 'image.png'
        file_path = os.path.join('backend/static', filename)
        im.save(file_path)
        skin_type = prediction_skin(file_path).split('_')[0]
        acne_type = prediction_acne(file_path)
        tone = identify_skin_tone(file_path, dataset=skin_tone_dataset)
        print(skin_type)
        print(acne_type)
        print(tone)

        return {'type': skin_type, 'tone': str(tone), 'acne': acne_type}, 200

api.add_resource(SkinMetrics, "/upload")
api.add_resource(Recommendation, "/recommend")

@app.route("/")
def index():
    return render_template('index.html')
```

```
@app.route("/quiz")
def quiz():
    return render_template('quiz.html')


@app.route("/home", methods=['GET', 'POST'])
def home():
    return render_template('home.html')




class DataStore():
    skin_r="hey"
    acne_r="b"


data = DataStore()


@app.route("/predict", methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        file = request.files['file']
        filename = file.filename
        file_path = os.path.join('backend/static',filename)                    #slashes
should be handeled properly
        file.save(file_path)
        skin_type = prediction_skin(file_path)
```

```
        data.skin_r=skin_type

        acne_type = prediction_acne(file_path)

        data.acne_r=acne_type

        print(skin_type)

        print(acne_type)


        #return {'type': skin_type, 'acne': acne_type}, 200 #skin_type, acne_type

        return render_template('prediction.html', skin_type=skin_type,
acne_type=acne_type)

    @app.route("/routine")

  def routine():

    # Assuming you have obtained skin_type and acne_type from prediction

    skin_type = data.skin_r #"Dry Skin"  # Example value, replace with actual
prediction result

    acne_type = data.acne_r #"Low"       # Example value, replace with actual
prediction result


    return render_template('recommendation.html', skin_type=skin_type,
acne_type=acne_type)


  if _name_ == "_main_":

    app.run(debug=False)
```

# APPENDIX II

## SCREENSHOT

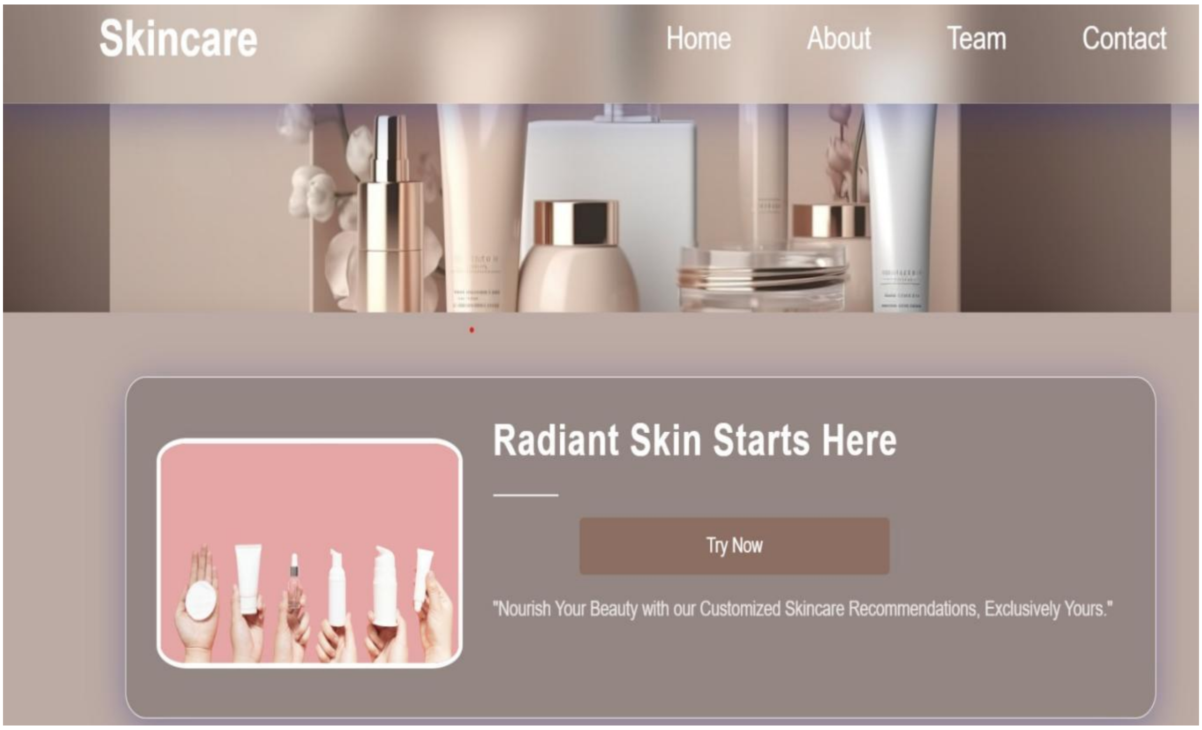


**Fig (a) Welcome Page**

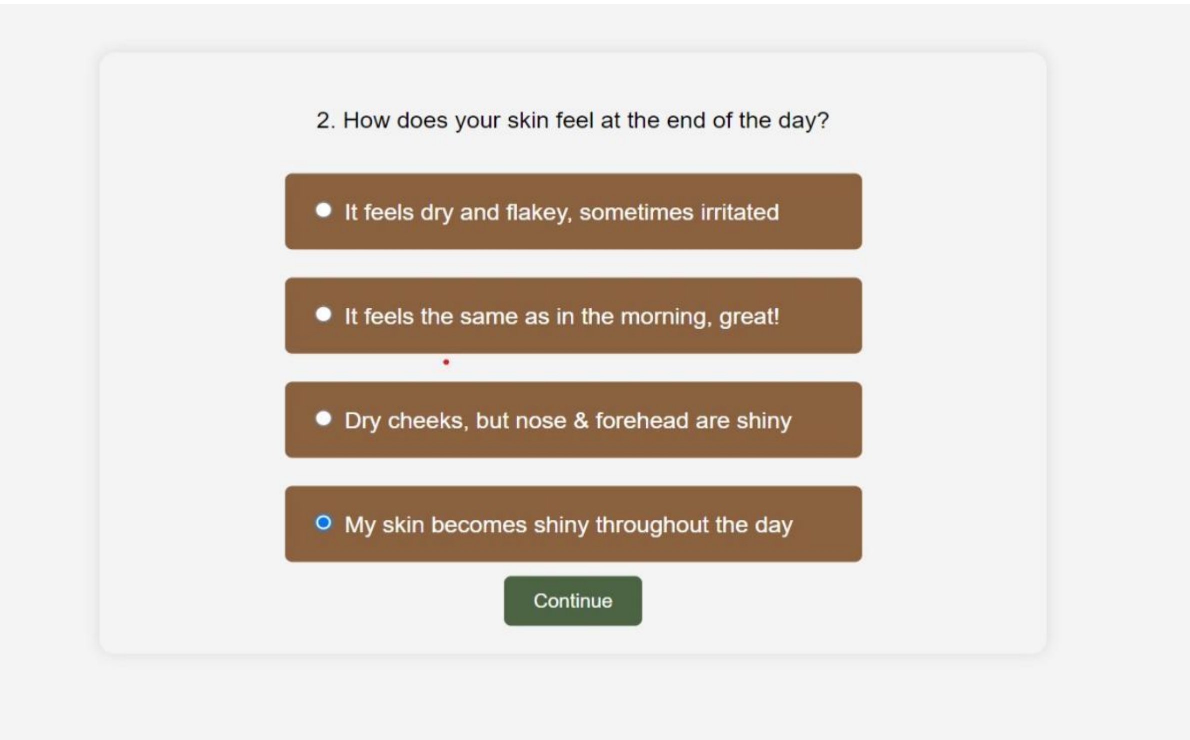


**Fig (b) Collection of User Data**

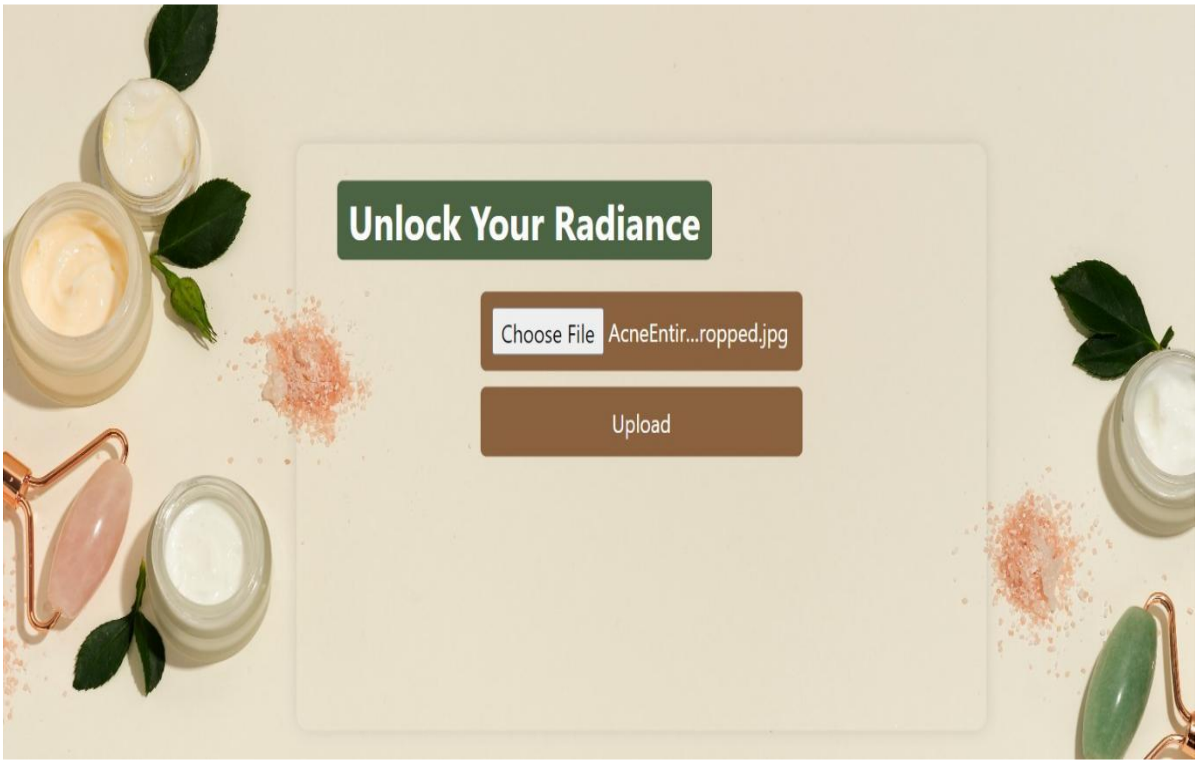


**Fig (c) Uploading the user image**

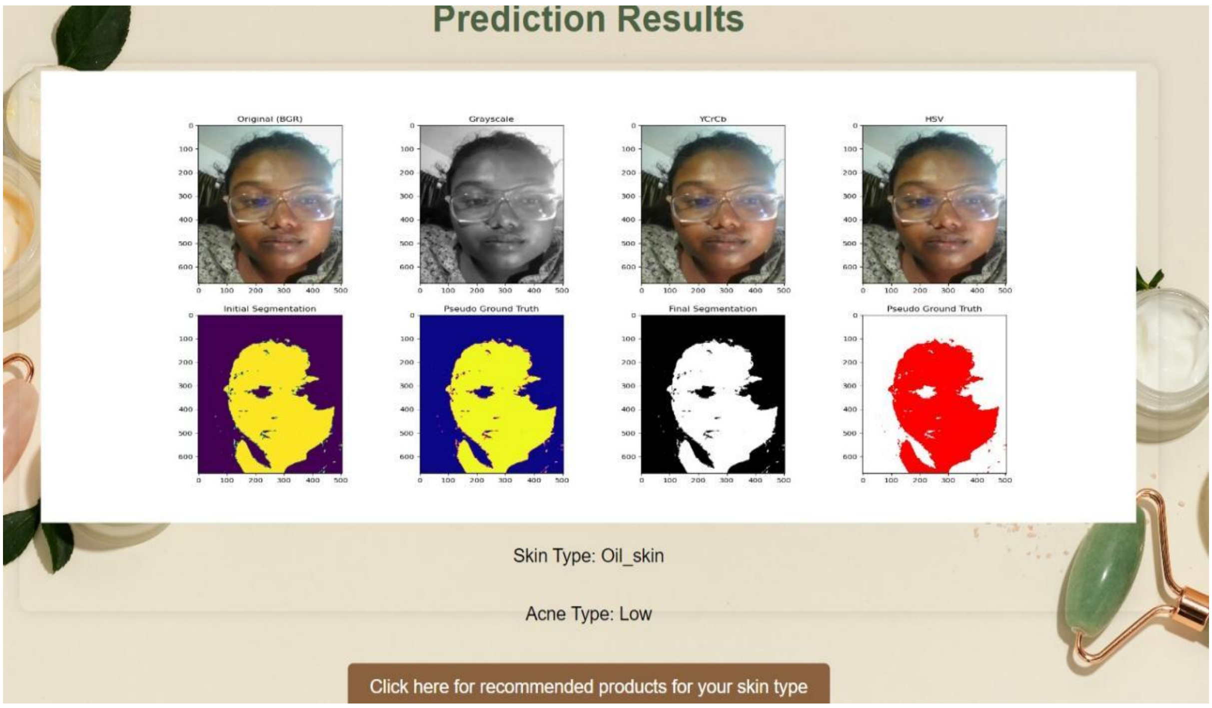


**Fig (d) Prediction Results**

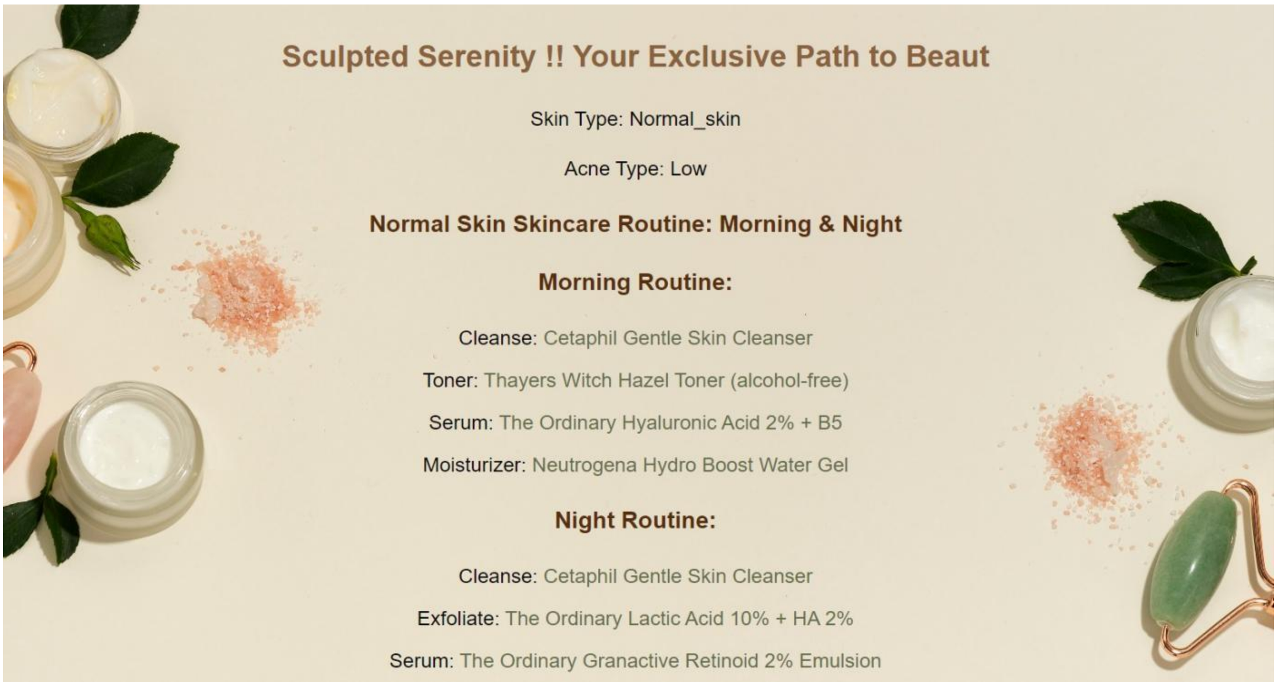


**Fig (e) Skincare Routine**

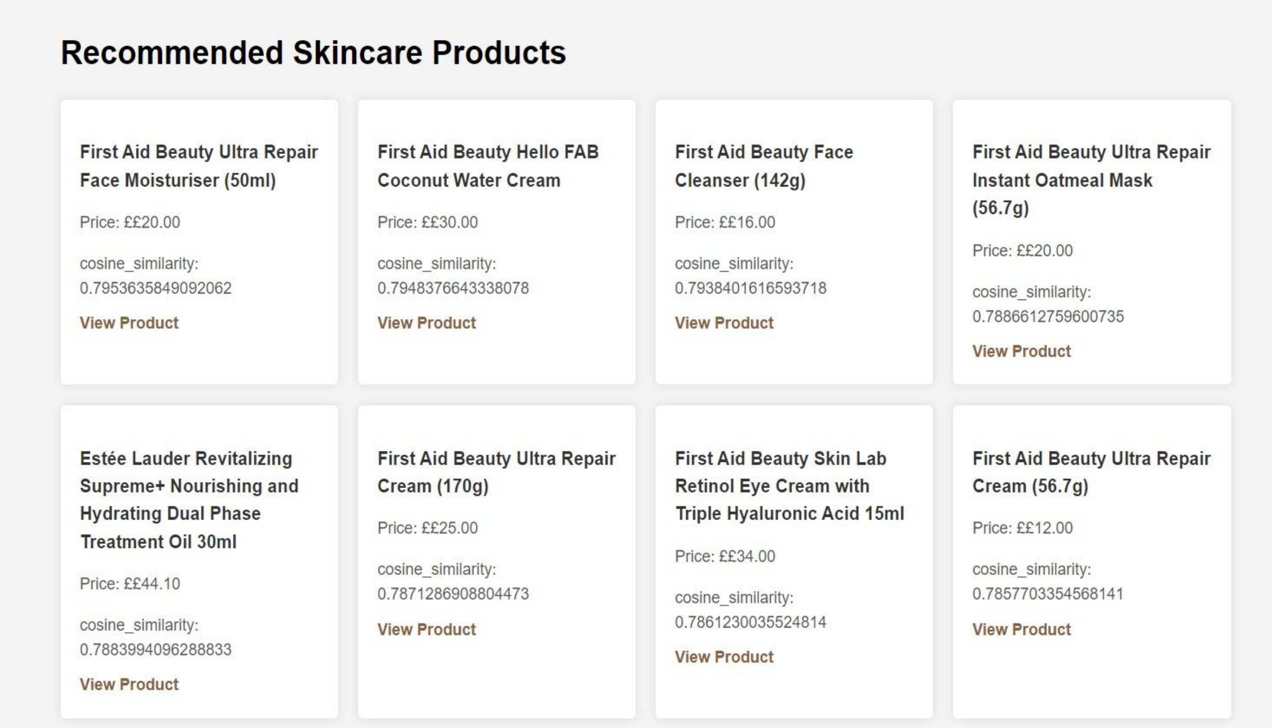


**Fig (f) Recommended Skincare Products**

## REFERENCES


[1] Dong x, Yan y, Ouyang w, (2021) "Style Aggregated Network For Facial Landmark Detection," Ieee Conference On Computer Vision And Pattern Recognition, Pp.379-388.

[2] Goindi, S., Thakur, K., & Kapoor, D. S (2023) "Skin Disease Classification and Detection by Deep Learning and Machine Learning Approaches."

[3] Iwabuchi, R., Nakajima, Y., et al. (2023) "Artificial Intelligence based Smart Cosmetics Suggestion System based on Skin Condition."

[4] Lee, Gyeongeun (2020) "A Content-based Skincare Product Recommendation System."

[5] Li, Yi, Song, Lingxiao, Wu, Xiang, He, Ran (2020) "AntiMakeup Learning A, B –Level Adversarial Network For Makeup Invariant Face Recognition," AAAI Conference, pp.79-388.

[6] Liu, Xudong, Taoli, Iris Chuoying Ouyang, and Ruizhe Wang (2023) "Understanding Beauty via Deep Facial Features," IEEE Conference, volume-246-256.

[7] Ma, X., Lu, J., Liu, X., & Kuang, H. (2022) "Skincare Recommender System Using Neural Collaborative Filtering with Implicit Rating."

[8] Rathgeb, Christian, Dantcheva, Antitza, Busch, Christoph (2022) "Impact Of Facial Beautification On Face Recognition," IEEE Access 2021,7,152667-152678.

[9] Saiwaeo, S., Arwatchananukul, S., Mungmai, L., Preedalikit, W., & Aunsri, N. (2023) "Human Skin Type Classification Using Image Processing and Deep Learning Approaches."

Heliyon 9 (2023) e21176

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

# Human skin type classification using image processing and deep learning approaches

Sirawit Saiwaeo [a], Sujitra Arwatchananukul [a,b], Lapatrada Mungmai [c,d], Weeraya Preedalikit [c,d], Nattapol Aunsri [a,b,*]

[a] School of Information Technology, Mae Fah Luang University, Chiang Rai, Thailand
[b] Integrated AgriTech Ecosystem Research Group (IATE), Mae Fah Luang University, Chiang Rai, Thailand
[c] Division of Cosmetic Science, School of Pharmaceutical Sciences, University of Phayao, Phayao, Thailand
[d] Research and Innovation Center in Cosmetic Sciences and Natural products, Division of Cosmetic Sciences, School of Pharmaceutical Sciences, University of Phayao, Phayao, Thailand

## ARTICLE INFO

*Keywords:*
Image enhancement
Image augmentation
Data preparation
Skin images
Contrast limited adaptive histogram equalization (CLAHE)
CNN

## ABSTRACT

Cosmetics consumers need to be aware of their skin type before purchasing products. Identifying skin types can be challenging, especially when they vary from oily to dry in different areas, with skin specialist providing more accurate results. In recent years, artificial intelligence and machine learning have been utilized across various fields, including medicine, to assist in identifying and predicting situations. This study developed a skin type classification model using a Convolutional Neural Networks (CNN) deep learning algorithms. The dataset consisted of normal, oily, and dry skin images, with 112 images for normal skin, 120 images for oily skin, and 97 images for dry skin. Image quality was enhanced using the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique, with data augmentation by rotation applied to increase dataset variety, resulting in a total of 1,316 images. CNN architectures including MobileNet-V2, EfficientNet-V2, InceptionV2, and ResNet-V1 were optimized and evaluated. Findings showed that the EfficientNet-V2 architecture performed the best, achieving an accuracy of 91.55% with average loss of 22.74%. To further improve the model, hyperparameter tuning was conducted, resulting in an accuracy of 94.57% and a loss of 13.77%. The Model performance was validated using 10-fold cross-validation and tested on unseen data, achieving an accuracy of 89.70% with a loss of 21.68%.

## 1. Introduction

Skin care is an essential component of a healthy lifestyle. During the past decade the average age of people using cosmetic skin care products has decreased, with higher demand among teenagers and working people. Skin care products promote a younger appearance, thereby boost confidence. Numerous manufacturers have responded to the rising demand for skin care products with many items available but skin type must be assessed before purchase. Skin specialists underline the necessity of understanding skin type before ordering products for direct dermal application. Some skin types can be identified by eyes, while the others cannot. Some

* Corresponding author at: School of Information Technology, Mae Fah Luang University, Chiang Rai, Thailand.
*E-mail address:* nattapol.aun@mfu.ac.th (N. Aunsri).

https://doi.org/10.1016/j.heliyon.2023.e21176
Received 14 May 2023; Received in revised form 11 October 2023; Accepted 17 October 2023

Available online 23 October 2023
2405-8440/

people have a mixture of skin types, with oily and dry skin in different regions of the body. The skin care industry has mushroomed over the past two decades, with Artificial Intelligence (AI) and Machine Learning (ML) utilized in many areas including healthcare, medicine, agriculture, industry and personal care and cosmetics. Deep learning (DL) is a machine learning technique that teaches computers to learn by example, and its use is rapidly expanding as technology and device specifications improve. Deep learning has advanced technology to the next level to solve real-world problems such as skin cancer classification [1–6].

Numerous studies have significant contributions to skin classification [7–18]. Research in 2020 supported dermatologists by providing fundamental information on deep learning and CNN, demonstrating their applications in the classification of skin diseases [2] and emphasizing the potential of CNN-based methods in automated diagnosis, while acknowledging the need for further research and advancements in image processing and pattern recognition for improved performance. The paper focused on two key areas: disease classification from medical images (such as dermoscopy and pathological images) and disease classification from digital photographs. The main concepts of deep learning now are for helping the dermatologists to understand and adopt computerized methods based on CNNs to showcase the state-of-the-art applications for lesion classification from medical images and colored photographs while also discussing their limitations. The paper also highlighted the lack of desktop applications for dermatological diseases other than skin cancer and suggested potential area for future development.

Many skin researches state that skin diseases pose significant challenges in terms of accessibility to medical care due to various factors such as physical disabilities, physiological issues, distance, limited medical expertise in rural areas, climate conditions, and employment constraints. The time consuming and expensive nature of diagnosis further discourages individuals from seeking dermatological care. However, widely available user-friendly smartphone applications are now prevalent in many fields including medicine. In the domain of dermatology, these applications have gained particular importance. Most skin diseases are visually identifiable, with diagnosis relying on examination of lesions and pattern recognition. The research conducted by [3] addressed whether skin diseases can be automatically diagnosed by leveraging advancements in mobile technologies and deep learning-based methodologies. A review of recent smartphone applications was conducted to explore their potential for empowering patients to actively manage their health and provide an assessment of their capabilities. In 2021, researchers successfully demonstrated teledermatology multiclass skin lesion detection and classification using a CNN model that generated images with 16 layers, and enhanced contrast using High Dimensional Color Transform (HDCT). The schema returned an RGB image as output with high contrast. Transfer learning was then applied to a pre-trained DenseNet201 model. The researchers extracted the features and integrated them into Extreme Learning Machine (ELM) classifiers with multiple classes. Results demonstrated that a CNN model could be pre-trained by a problem-related model and used for feature extraction. The output of the proposed model outperformed the four popular datasets ISBI2016, ISIC2017, PH2 and ISBI2018.

Later, in [4], researchers proposed a cosmetic skin type classification as four main categories including normal, dry, oily and combination utilizing a CNN to categorize skin type based on cosmetic items with the same label. Their classification model accuracy was 85% with a slight preference for oily images.

The field of deep learning requires a powerful processing unit to compute large amount of data with image augmentation used to artificially expand a small dataset [19–22]. In 2021, [19] suggested that image data augmentation improved deep learning-based model performance in pathological lung segmentation. The study used a variety of techniques by generating a random patch inside the lung and then blurring the area with a Gaussian filter. Contrast and brightness were then adjusted Experimental results revealed that this data augmentation strategy outperformed the standard approach in deep learning. Following these advancements, this research has three main objectives as follows, including:

- To contribute to deep learning in the medical field by developing an image classification system for human skin types.
- To provide a convenient alternative for dermatologists and patients who may lack time or resources for in-person consultations.
- To combine deep learning with specialist knowledge, to effectively classify different human skin types, thereby reducing the burden on dermatologists.

This paper is organized as follows. Section 2 provides an overview of various techniques including image augmentation, image enhancement, and classification model construction. Section 3 presents the proposed methodology and outlines the overall approach. The experimental results in Section4 which showcase the outcomes obtained from the proposed stages, while Section 5 highlights the study drawbacks and limitations. Finally, Section 6, draws conclusions, summarizes the outputs derived from the proposed method, presents a comparative analysis of related research, and discusses future study areas.

## 2. Literature reviews

### *2.1. Human skin types*

Human skin types differ [23], with each having its own unique characteristics and care needs. The three main skin types include normal, oily, and dry skin. Fig. 1 shows samples of image dataset for each type: (a) dry skin image, (b) normal skin image, and (c) oily skin image. Following describes the characteristic of each skin type.

- *Dry skin* lacks sufficient moisture, leading to tightness, roughness, and flakiness, often having a dull and dehydrated appearance. People with dry skin experience discomfort and increased sensitivity, sometimes leading to irritation. Dry skin can be influenced by various factors such as the environment, genetics, aging, and inadequate hydration. Regular moisturization is crucial to replenish moisture and maintain a healthy skin barrier, with severe dryness resulting in eczema or dermatitis.

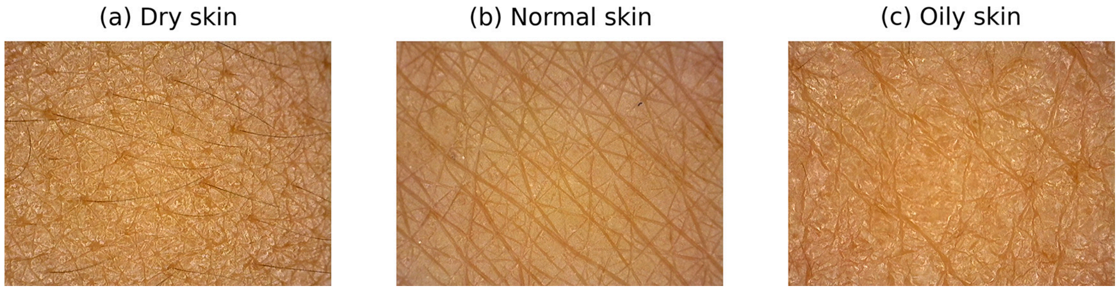


**Fig. 1.** The three types of skin including (a) dry skin, (b) normal skin, and (c) oily skin.

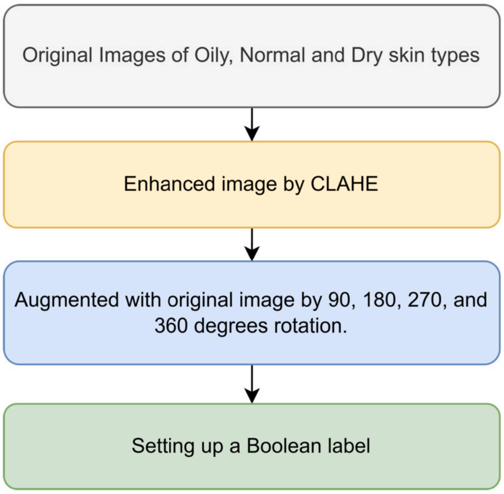


**Fig. 2.** Overall process of data preparation.

- *Normal skin* is considered healthy and well-balanced. It has a smooth texture, good elasticity, and maintains a natural moisture balance. People with normal skin have few skin issues, as their skin is neither excessively oily nor dry. The skin looks clear, with medium-sized pores and an even tone, producing a healthy amount of sebum which keeps it moisturized without being too oily.

- *Oily skin* is characterized by an overproduction of sebum. The skin appears shiny or greasy, particularly in the T-zone (forehead, nose, and chin). Oily skin often has enlarged pores and is prone to acne, blackheads, and whiteheads due to excess oil clogging the pores. However, oily skin tends to age at a slower rate and has natural protection against fine lines and wrinkles.

### *2.2. Image acquisition and preparation*

A DSLR Camera was used for capturing the human skin images. All images are face skin images taken under fluorescence light source. The original image files (640 × 480 pixels) were saved into JPEG format. It is the collection of a total of 60 subject's data. The subjects from the ages of 20 to 45 years old took part in our experiment. Image preparation increases the quality of the original image data used for training the computer vision models. Image enhancement and augmentation were performed during the preparation phases of skin datasets to improve the quality and increase data quantity to an acceptable level. Preparatory method techniques were compared for optimal model fit and performance. The model with the best score metrics was determined by combining data preparation using the original datasets, datasets with enhanced image quality, datasets with image augmentation to increase data, and labeling data into a Boolean array, as shown in Fig. 2.

#### *2.2.1. Contrast limited adaptive histogram*

Image enhancement is a modification process to ensure a more appropriate classification because some images may contain unclear features. Enhancing the image contrast improves skin texture, with marks more visible. CLAHE was used to enhance and increase image contrast depending on the near-constant image regions. CLAHE is a variation of adaptive histogram equalization (AHE) that is used to control excessive contrast amplification and compute multiple histograms each focusing on a specific region of the image to redistribute the luminance values [24]. By contrast, CLAHE is more complex to implement than regular adaptive methods. CLAHE was designed to avoid the problem of overamplification and is based on the algorithm shown in Fig. 3. When an image is received, the image is then split into 8x8 tiles (OpenCV's default size) and distortion correction is applied to each block. Then, the titles are checked to see if they exceed the contrast limit. If so, excess tiles are cut off and accumulated. The cumulative

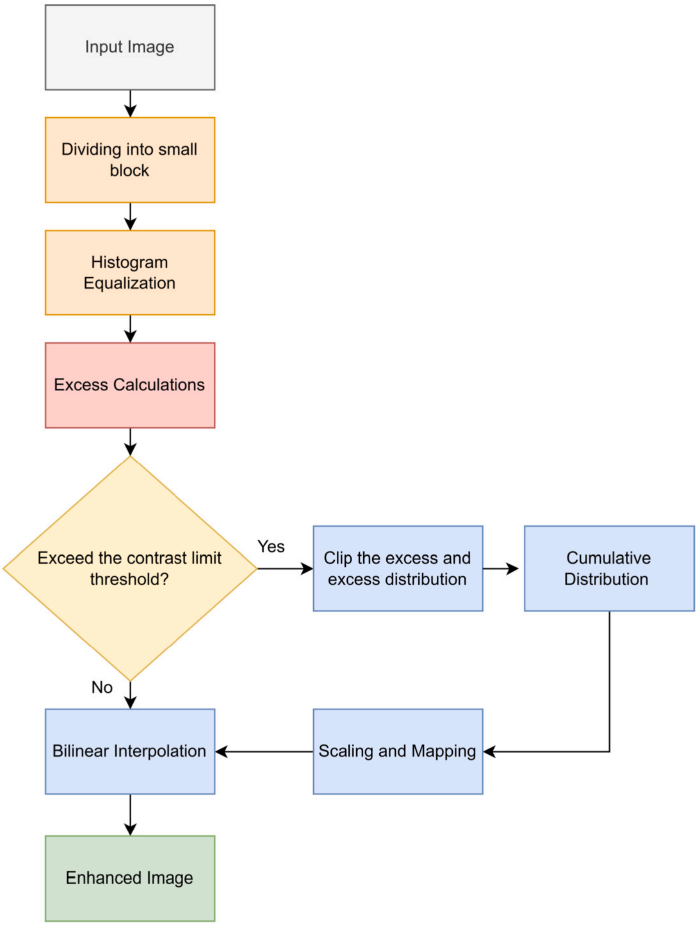


**Fig. 3.** A CLAHE workflow diagram.

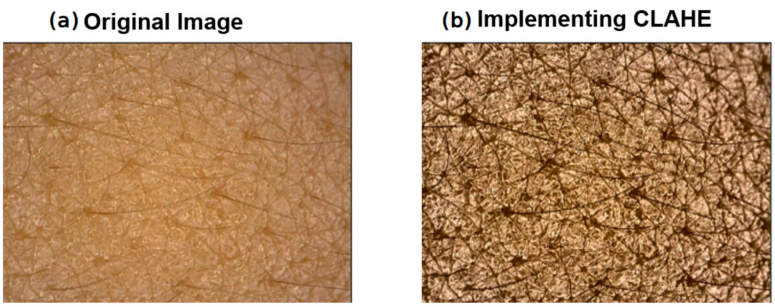


**Fig. 4.** Comparison between a) the original and b) enhanced image using CLAHE.

overrun is scaled and mapped to other blocks. Finally, a bilinear interpolation algorithm is applied to assemble all blocks into one image [25], [26].

Fig. 4 shows a comparison between the original image and a new image created by CLAHE method. The main effect of CLAHE is that it increases the visibility of details and textures in areas with low contrast by stretching the intensity range of each tile, making both darker and lighter regions more distinguishable. This technique is particularly useful for enhancing images that have non-uniform illumination or low contrast, resulting in an image with improved clarity and better visual representation of fine details.

Noise is an important factor affecting the performance of automated classifications. To overcome this issue, many denoising methods have been used according to the type of noises [27]. The proposed method has achieved high performance without any extra noise reduction step, which usually leads to increased computational costs. Various normalization algorithms with different types of images have been used to obtain high performances [28,29], however, they may increase computational costs. Therefore, we will not apply an extra intensity normalization step in our proposed approach to obtain efficient results.

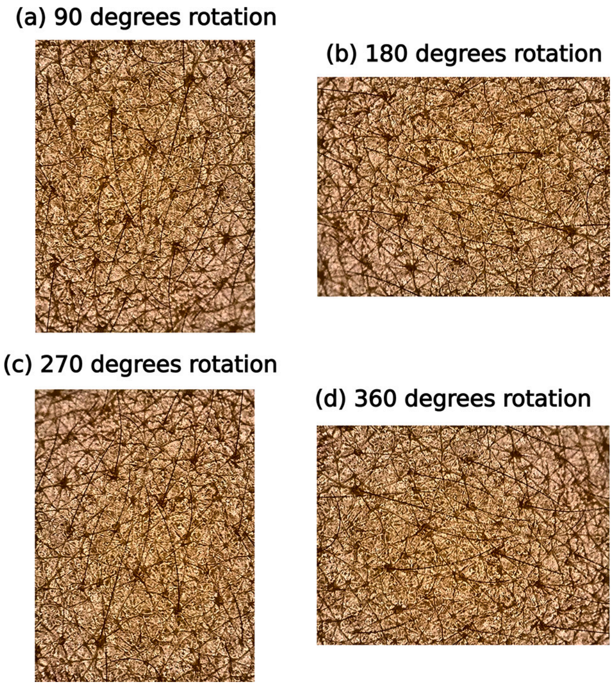


**Fig. 5.** Examples of rotating images: (a) 90 degrees rotation, (b) 180 degrees rotation, (c) 270 degrees rotation, and (d) 360 degrees rotation.

#### *2.2.2. Image augmentation by rotation*

Computer vision data augmentation is an effective technique for enhancing the performance of computer vision models without collecting extra data. One typical data augmentation technique is random rotation. As illustrated in Fig. 5, the position of the original image was transformed by 90, 180, 270 and 360 degrees of clockwise rotation and the number of skin image was increased 4-fold to 1,316 images from 329 images.

#### *2.2.3. Multi-label skin type data*

The string-type labels were converted into a Boolean array, with false and true representing 0 and 1, respectively to classify the three skin types. Data in row and columns are presented as Boolean values. If the array labels at the first two positions are false, the data relate to oily skin.

### *2.3. Convolutional neural network for skin classification*

Deep learning now complements artificial intelligence, with computers trained to perform classification tasks directly from images, text or voice. Machine learning is now used as a tool for sophisticated calculations.

A CNN is one of the most effective machine learning algorithms. Raw images can be fed into the model as the building blocks for model learning.

This study focused on the smallest model with the fastest training time. The four models MobileNet-V2, EfficientNetV2, Inception-V2 and ResNet are described below.

#### *2.3.1. MobileNetV2*

MobileNetV2 [30] is an innovative deep neural network architecture that specifically caters to the needs of efficient and lightweight image classification and feature extraction on mobile and embedded devices. Developed as an extension of the original MobileNet framework by Google, MobileNetV2 introduces a range of enhancements to improve performance and efficiency. Architecture of the MobileNet-V2 model is illustrated in Fig. 6.

The cornerstone of MobileNetV2 lies in its utilization of depth-wise separable convolutions, which divide the traditional convolution operation into distinct depth-wise and point-wise convolutions. By decomposing the process, MobileNetV2 significantly reduces computational complexity and parameter count while maintaining a commendable level of accuracy. Depth-wise convolutions independently perform spatial filtering for each input channel, while point-wise convolutions consolidate the filtered outputs across channels.

One of the key advancements in MobileNetV2 is the introduction of inverted residual blocks that amplify the representation power of the network while simultaneously minimizing computational costs. Inverted residuals employ a bottleneck structure that compresses input feature maps using 1x1 convolutions, followed by a depth-wise convolution. Subsequently, the dimensions are expanded again using 1x1 convolutions. This bottleneck design drastically reduces computation while facilitating seamless information flow through the network.

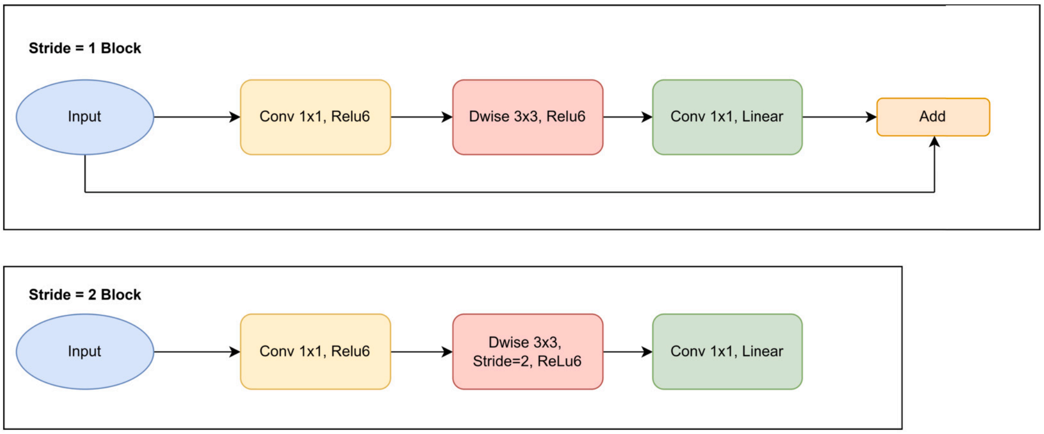


**Fig. 6.** Architecture of the MobileNet-V2 model.

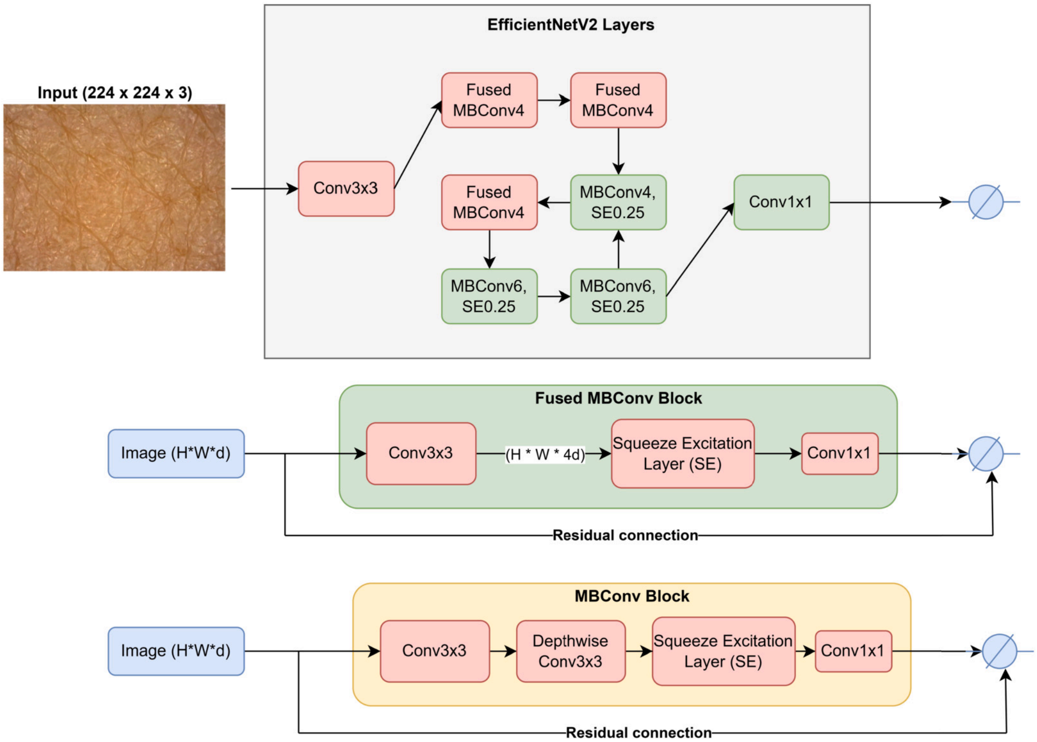


**Fig. 7.** Architecture of the EfficientNet-V2 model [31].

### 2.3.2. *EfficientNetV2*

The EfficientNet-V2 model design is shown in Fig. 7, with an inverted residual block or MBConv Block, both of which contribute to compact size and rapid training times. A residual block is used in image models to solve efficiency issues, and this technique has been widely implemented in mobile CNN models. The 1x1 convolution in the inverted residual block minimizes the parameters, while 3x3 convolution layers use depth-wise convolution to conserve processing resources. Depth-wise convolution is a separate operation from the standard convolution, which typically transforms the feature by multiplying it repeatedly by some factor. This approach is commonly used in CNN models tuned for use on mobile devices. Once the feature has been extracted from the 3x3 convolution layer, another round of field restriction with the 1x1 convolution layer is required. The MBConv building block struggles to deal with huge feature images or large overall image sizes. To speed up the traditional MBConv procedure, the Fused Inverted Residual or Fused MBconv was introduced in [31] by combining the 1x1 Conv and 3x3 Conv into a single 3x3 Conv to accelerate the model to the next level.

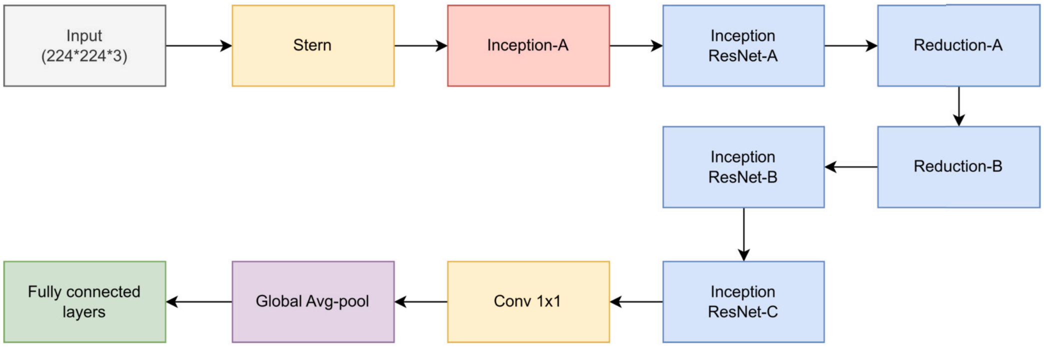


**Fig. 8.** Architecture of the Inception-V2 model.

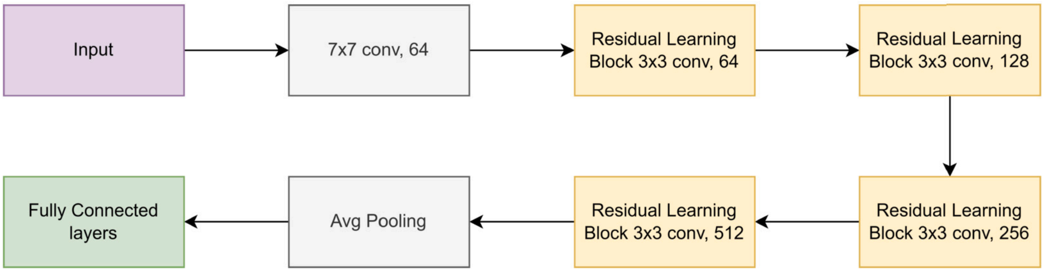


**Fig. 9.** Architecture of the ResNet-V1 model.

### *2.3.3. InceptionV2*

InceptionV2 [32], also known as GoogLeNet, is a deep CNN architecture introduced as an improvement over the original Inception network and developed by researchers at Google to address the challenges of training deeper networks while keeping computational requirements manageable. The key innovation in InceptionV2 is the introduction of the "Inception module" which is a stacked set of parallel convolutional layers of different sizes. Fig. 8 presents an architecture of the Inception-V2 model. In this model, the parallel layers perform different receptive field sizes and capture features at different scales, allowing the network to learn a diverse range of features. The Inception module replaces the traditional single-size convolutional layer and helps prevent information loss due to fixed filter sizes.

### *2.3.4. ResNet-V1*

ResNet [33] architectures typically consist of several stacked residual blocks. Each residual block contains a set of convolutional layers and shortcut connections. The convolutional layers within the block perform feature extraction and transformation, while the skip connections allow the network to propagate information from earlier layers directly to later layers. This facilitates the flow of gradients during training and enables the network to learn more efficiently. The architecture of the ResNet-V1 model is provided in Fig. 9. The key advantage of ResNet is its ability to train very deep networks by using residual connections, gradients can propagate more effectively through the network, alleviating the vanishing gradient problem and allowing successful training of deeper architectures. This enables the network to learn more complex and abstract features, leading to improved performance in tasks such as image classification, object detection, and image segmentation.

### *2.3.5. Feature extraction*

A CNN with deep learning has two layers. In the initial convolutional layer, a procedure occurs where a filter travels over the picture receptive fields and checks features in the image. To ensure that every pixel in the image has been checked for features, the image is run through several filters. After applying a filter to an image, the distance between the input pixels and the filter is determined. This layer produces a feature map, also known as convolved features map as collection of dots. The second layer is a pooling layer, which repeats a filter loop across the image. To reduce data collection size, the number of input parameters can be reduced to ensure enhanced functionality of the CNN while reducing its complexity [34].

### *2.3.6. Training a neural network*

Equation (1) presents the training in a neural network as follows:

$$\theta_{k+1} = \theta_k - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla_{\theta k} \ell \left( f_{\theta k}(x_i), y_i \right), \tag{1}$$

where $x$ contains the input features and $f(\bullet)$ to denote a neural network model. The model can be written as $\hat{y} = f(x)$ to input a prediction, with model parameters adjusted to reduce the loss function. The term $\theta$ denotes all the parameters, while $\hat{y} = f_\theta(x)$ describes the state of model prediction depending on the current state of the parameters. The goal of training a neural network is to minimize the loss function which describes the model failings. If $y$ is the goal to achieve and $\hat{y}$ is the prediction, the loss function can be denoted as $\ell(\hat{y}, y)$. Machine learning trains models using the training set as the input $x$, with $N$ as the number of examples. Neural networks use loops to minimize the loss function such as $\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i)$ using gradient descent. If $\theta_k$ is the current state of the model, an improvement as the next state can be represented as $\theta_{k+1}$ with minimized loss function. The solution can be written as equation (1) with $\triangledown$ as the gradient to adjust parameter $\theta$.

*"The new parameters $\theta_{k+1}$ are equal to the old parameters $\theta_k$ minus the gradient concerning the old parameters of the error/loss of the neural network's prediction against the correct predictions averaged over the entire dataset and down-weighted by the learning rate." [35]*

### 2.4. Evaluation metrics

The classification result can be identified as positive or negative. Positive indicated a correct classification result while negative does not. A combination of positive and negative makes the prediction metric more detailed using precision, recall and F1-score to define the concepts of true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Followings are the evaluation metrics used in this work.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- **Precision,** defined as Equation (2), measures how often accurate positive forecasts (true positives) are produced.
- **Recall,** defined as Equation (3), is a measure of how many of the positive samples in the data were correctly predicted by the classifier.
- **The F1-score,** defined as Equation (4), is a measure that combines recall and precision by calculating their harmonic mean. This average value is generally more suitable for ratios than the conventional arithmetic mean.
- **Accuracy,** defined as Equation (5), represents the proportion of correctly labeled subjects in comparison to the entire group of topics. It serves as a valuable metric for evaluating the overall correctness of a classification model.

#### 2.4.1. Confusion matrix

A confusion matrix [36] is a table that summarizes the performance of a classification model. It provides a detailed breakdown of the predicted and actual classes for a set of instances. The matrix is organized into rows and columns, where each row represents the predicted class, and each column represents the actual class. The entries in the confusion matrix reflect the number of instances that fall into specific categories based on their predicted and actual class labels. A prediction matrix evaluates of the model accuracy and helps to identify the types of errors. The patterns within the matrix allow insights into the model's strengths and weaknesses. A confusion matrix is a fundamental tool for performance evaluation in classification tasks, serving as a basis for calculating various evaluation metrics such as accuracy, precision and recall to provide a more comprehensive understanding of the model's performance. A confusion matrix provides a clear visual representation of the classification results, allowing for a thorough analysis of the model's predictions and identifying areas for improvement.

#### 2.4.2. Hyperparameter tuning

Hyperparameter tuning [37] refers to the process of selecting the optimal settings or configurations as the hyperparameters of a machine learning model. Hyperparameters are values that are set before the learning process begins and influences the behavior and performance of the model. Tuning these hyperparameters involves systematically exploring different combinations or ranges of values to find those that result in the best performance, such as accuracy or error rate. By fine-tuning the hyperparameters, we aim to enhance the model's ability to generalize and make accurate predictions on unseen data.

#### 2.4.3. K-fold validation

K-fold cross-validation [38] is a commonly used technique in machine learning for evaluating the performance of a model. It involves dividing the dataset into k subsets, or folds, of equal size. The model is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold being used once as the test set. The results are then averaged to give an overall performance measure of the model. K-fold cross-validation is used to assess the model's performance on new data and to prevent overfitting that occurs when a model is too complex and is fitted too closely to the training data, resulting in poor performance on new data. By using k-fold validation, the model is evaluated on multiple test sets, providing a more accurate estimate of its performance on unseen data.

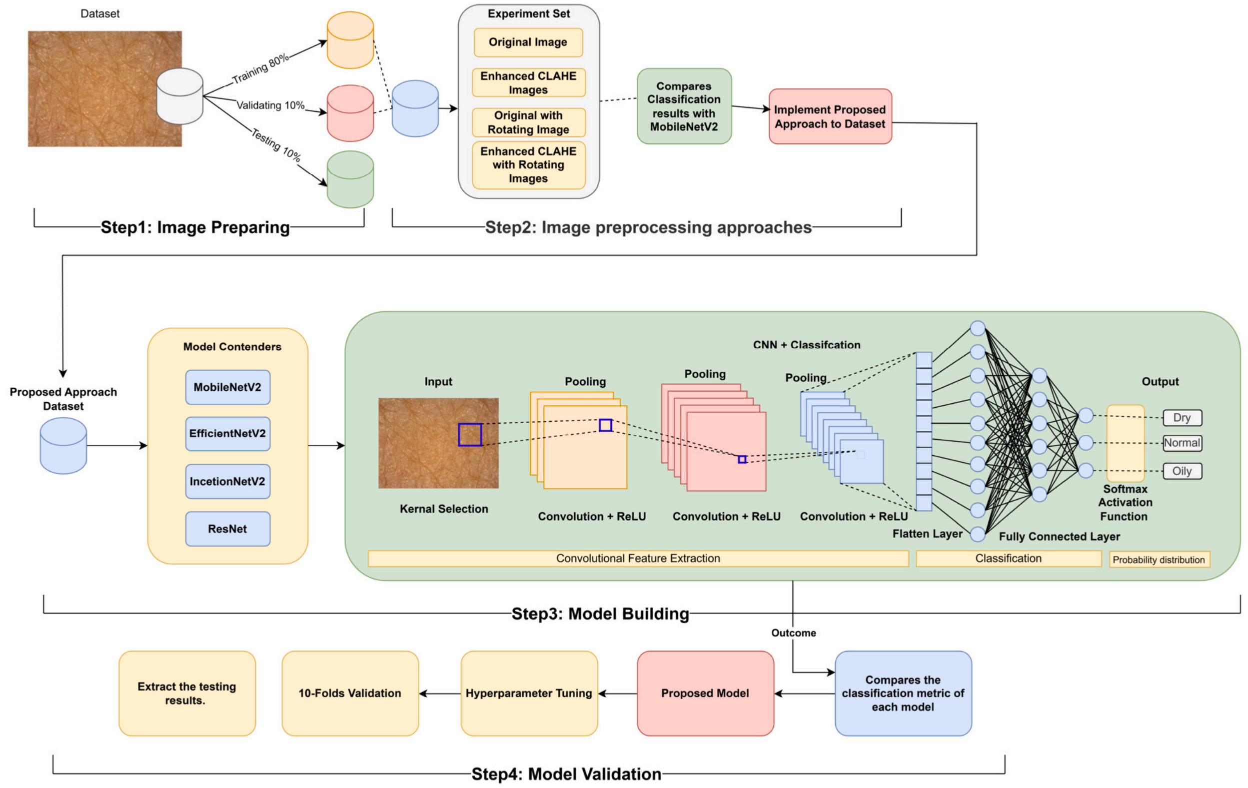


**Fig. 10.** The proposed methodology.

### *2.4.4. Receiver operating characteristic (ROC) curve*

The area under the Receiver Operating Characteristic curve [39], often referred to as AUC-ROC or simply AUC, is a popular evaluation metric used in binary classification tasks to measure the performance and discriminative power of a classification model. The ROC curve is a graphical plot that illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various classification thresholds. The AUC-ROC quantifies the overall model performance by calculating the area under the curve. The AUC-ROC metric ranges between 0 and 1, with a value of 0.5 indicating that model performance is no better than random guessing, while a value of 1 indicates a perfect classifier with no false positives or false negatives. A higher AUC-ROC score indicates better model discriminative ability to distinguish between positive and negative instances, with higher probabilities assigned to positive than to negative instances. The AUC-ROC metric is particularly useful when dealing with imbalanced datasets or when the cost of false positives and false negatives is not equal. The AUC-ROC score provides a comprehensive evaluation of model performance across various classification thresholds, allowing the selection of an optimal threshold that balances true positives and false positives.

### *2.4.5. Gradient-weighted class activation mapping (GRAD-CAM)*

GRAD-CAM [40] is a technique used in deep learning models to understand which parts of an input image contribute most to the predictions made by the model. GRAD-CAM helps to visualize and interpret the decision-making process by highlighting the image regions that strongly influence output values. GRAD-CAM achieves this by computing the gradients of the target class score with respect to the feature maps in the final convolutional layer. These gradients are then used to obtain a weighted combination of the feature maps, resulting in a heatmap that highlights the important areas of the image.

## 3. Methodology

The research protocols were authorized by the University of Phayao Human Ethic Committee, Thailand, approval number 3/019/58. The study was carried out on healthy volunteers who all have read and signed a written informed consent form.

The proposed approach involved four key steps and relied on deep learning to classify images of three different skin types, as shown in Fig. 10. Each step is explained below.

- **Step 1 Image Preparation:** Three skin types were divided into segments as 80% for training, 10% for dataset validation and 10% for dataset testing and categorized according to the kind of skin image, as shown in Fig. 1, with 112 images for normal skin, 120 for oily skin and 97 for dry skin types. The three skin types in this dataset were determined by dermatologists.

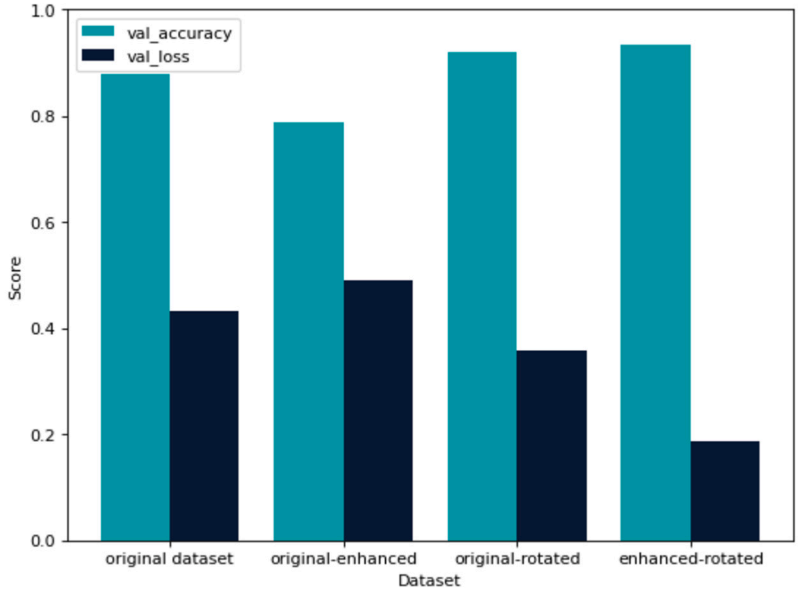


**Fig. 11.** Accuracy and loss validation of the different datasets.

- **Step 2: Image pre-processing approaches:** This step focused on identifying the optimal preprocessing method that aligned with our human skin type dataset. The experiment involved applying various preprocessing techniques to each dataset. The implemented datasets were then evaluated to determine which yielded higher accuracy, with MobileNetV2 employed as the decision-making model.
- **Step 3: Model building:** The datasets were used to test the transfer learning models MobileNet-V2, EfficientNet-V2, Inception-V2 and Resnet to determine which was the most effective at classifying human skin types.
- **Step 4: Model validation:** The proposed model was then tested to determine more reliable result using several tools such as cross-check validation, test with unseen data to extract the ROC curve and visualize the output image by implementing GRAD-CAM.

## 4. Experimental results

In the first experiment, four datasets were compared to classify the skin model as the original dataset, the dataset with enhancement, the dataset with augmentation, and the dataset with both augmentation and enhancement. MobileNet-V2 was used as the classifier to retrieve the optimal quality results. The training model was tested with a data set that included enhancements made to 329 images using common local area heterogeneous enhancement methods. Image augmentation was conducted by rotating the original 329 skin images by 90, 180, 270, and 360 degrees to give 1,316 new images for use in this work. Finally, the images were enhanced to train the model. The data were divided as 80% for the training dataset, and 10% to validate the classification model, and 10% to test the model as an unseen data. As a result, the dataset changed, while all other parameters remained constant.

Based on the results depicted in Fig. 11, the CLAHE enhancement and image augmentation by rotating the image provided optimal performance and robustness. This data preparation technique yielded a model with increased accuracy while minimizing the loss metric. A non-enhanced and rotated strategy can achieve the same accuracy as enhanced rotation but precision is lost in the process. The loss function describes the accuracy of the model prediction method. The distance between the algorithm output and the expected result was calculated with a high loss, indicating major errors in the data. The experiment utilized identical pain scores, size and quantity of data. The model with the highest accuracy and lowest loss function gave the optimal performance.

An imbalanced dataset was managed by adding rotating images to increase data quantity. The enhanced images also improved the algorithm extracted features, leading to more accurate outputs from the underlying learning model. Selecting the best model possible to maximize the learning process is essential.

Fig. 12 displays the confusion matrix for the validating phase. Findings demonstrated that the EfficientNet-V2 model had high level performance and reliability, with overall classification accuracy 91.55%. For normal skin, the model had the highest precision, 88.88% and recall 88.88%, whereas dry skin precision of 95.57% with highest recall 98.18%. For oily skin, precision was 89.28% and recall 86.20% because oily skins are difficult to define visually, making feature extraction harder.

Table 1 presents the performance results of different models specific task. MobileNet-V2 achieved an accuracy of 89.52%, with comparable precision and recall scores around 89%. EfficientNet-V2 performed slightly better with an accuracy of 91.55% and similar precision and recall scores. Inception-V2 exhibited the highest accuracy among the models at 92.22%, with precision, recall, and F1-score around 92%, while ResNet-V1 had the lowest performance with an accuracy of 72.94% and lower precision, recall, and F1-score compared to the other models. The loss values indicate that EfficientNet-V2 had the lowest loss at 22.74%, while Inception-V2

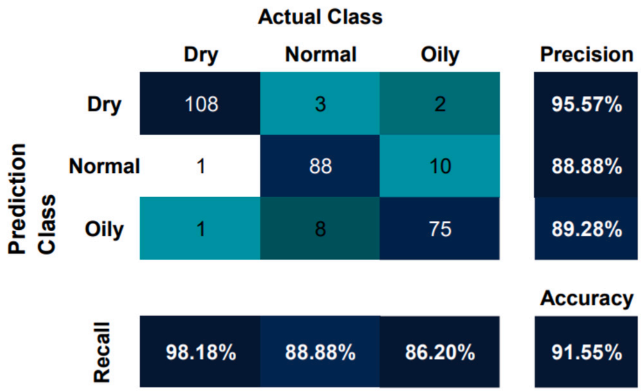


**Fig. 12.** Confusion Matrix of the Prediction Model.

**Table 1**
Experimental percentage results.

| Model | Accuracy | Precision | Recall | F1-score | Loss |
|---|---|---|---|---|---|
| MobileNet-V2 | 89.52 | 89.47 | 88.41 | 88.66 | 26.69 |
| **EfficientNet-V2** | **91.55** | **91.24** | **91.09** | **91.15** | **22.74** |
| Inception-V2 | 92.22 | 92.27 | 92.08 | 92.16 | 33.65 |
| ResNet-V1 | 72.94 | 72.31 | 69.59 | 70.89 | 56.27 |

**Table 2**
Hyperparameter search optimization results.

| Best epoch | Learning rate | Dense units | Accuracy % | Loss % |
|---|---|---|---|---|
| 47 | 0.0001 | 512 | 94.59 | 17.99 |
| **43** | **0.0001** | **416** | **94.59** | **17.78** |
| 42 | 0.0001 | 512 | 94.93 | 18.22 |
| 2 | 0.0001 | 512 | 94.25 | 17.48 |
| 15 | 0.001 | 256 | 94.59 | 18.02 |

had a slightly higher loss at 33.65%. Overall, results suggest that Inception-V2 performed the best in terms of accuracy and overall performance, while ResNet-V1 had the lowest performance among the evaluated models.

Results in Table 1 show that at 100 epochs with same data, InceptionV2 outputs the greatest accuracy but the loss was quite high. In part of EfficientNetV2, the output was outperformed in terms of accuracy and loss. The results were represented as classification matrix to validate loss and accuracy per epoch in each model.

Results also showed that the models gave the optimal classification for dry skin type because the characteristics of dry skin are easier to classify than the other types. Normal skin and oily skin, images look the same but had some characteristics and features that differed in computer vision. The EfficientNetV2 model outperformed the other three models in terms of accuracy and loss values, InceptionV2 gave the highest accuracy but also the highest loss metric, while the MobileNetV2 model had highest performance learning times.

### *4.1. Hyperparameter optimization results*

The Hyperband algorithm [41], which is a random search, was used employed to optimize resources efficiency for each possible configuration. This algorithm is fast and uses time efficiently by optimizing random sampling to achieve the best results. Traditional optimization involves looping or randomizing all parameters which can take a long time. Therefore, the Hyperband only runs 1-2 iterations. If the outcome is not satisfactory, proceed to the following sets. If they achieve a decent outcome in one or two iterations, then run them for a longer period of time. This method was followed to achieve optimal results.

Table 2 displays the results of tuning hyperparameters during the training process. The best epoch achieved was 47, with a learning rate of 0.0001 and 512 dense units. This configuration yielded an accuracy of 94.59% and a loss of 17.99%. At epoch 43 with the same learning rate and a slightly lower number of dense units (416), the model achieved the same accuracy and a slightly lower loss of 17.78%. On epoch 42, the model with 512 dense units achieved a slightly higher accuracy of 94.93% but had a higher loss of 18.22%. At epoch 2, the model with 512 dense units achieved an accuracy of 94.25% and a lower loss of 17.48%. Lastly, at epoch 15 with a learning rate of 0.001 and 256 dense units, the model achieved an accuracy of 94.59% and a loss of 18.02%. These results demonstrate the impact of different hyperparameter settings on the model performance, showcasing the trade-offs between accuracy and loss that can be achieved by tuning these parameters.

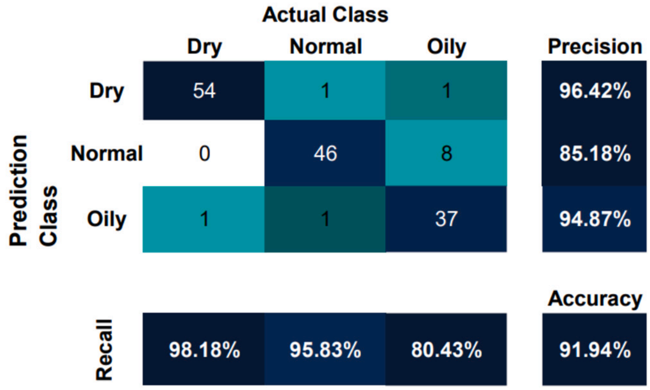


**Fig. 13.** Confusion Matrix of the Tunned Model.

**Table 3**
Accuracy and loss percentages per fold.

| Fold | Accuracy | Loss |
|---|---|---|
| 1 | 91.89 | 18.88 |
| 2 | 95.27 | 13.91 |
| 3 | 93.24 | 15.68 |
| 4 | 95.94 | 11.95 |
| 5 | 93.91 | 15.53 |
| 6 | 95.94 | 9.49 |
| 7 | 96.62 | 12.14 |
| 8 | 98.64 | 4.61 |
| 9 | 95.27 | 20.01 |
| 10 | 95.94 | 12.34 |
| **Average** | **95.27** | **13.45** |

From the tuning results in Table 2, the second will be the best of 30 attempts, because the top one unfortunately makes no difference compares to the second but for the loss metrics is impressive noticeable. Therefore, the second one will be the best parameters for human skin type classification approaches. At the end of searching parameters, the improvement from un-tunning model and tuning model has a large of gap in terms of loss metrics by went from 0.2274 to 0.177808 as 24.48%. For the accuracy is also improved but not significantly by went from 0.9155 to 0.9457 as 3.29%.

Fig. 13 summarizes the performance of a classification model in a simple and concise way. For dry skin type, there were 54 correct predictions (true positives), 1 instance misclassified as normal skin (false negative), and 1 instance misclassified as oily skin (false negative). For normal skin type, there were 46 correct predictions (true positives), 8 instances misclassified as oily skin (false positive), and no instances misclassified as other types. For oily skin type, there were 37 correct predictions (true positives), 1 instance misclassified as dry skin (false negative), and 1 instance misclassified as normal skin (false positive).

### *4.2. Model validation results*

Results focused on the evaluation and assessment of the developed model using appropriate validation techniques and metrics.

#### *4.2.1. K-fold cross-validation*

K-fold cross-validation can identify model stability, dependability and reliability by running the cross-validation multiple times with different random searches or shuffling to observe variations in performance metrics.

The Results of k-fold cross-validation with 10 attempts and 100 epochs for each fold. The model exhibited average results, with an accuracy metric of 95.27% and a loss metric of 13.45%. Compared to previous experiments involving hyperparameter tuning, accuracy showed a marginal increase rising from 94.59% to 95.27%. Conversely, the loss metric underwent a substantial improvement, decreasing from 17.77% to 13.45%, marking a significant enhancement compared to the previous iteration (please see Table 3).

#### *4.2.2. Validate model with unseen data*

Validation of a model using a test set guarantees a dependable and impartial evaluation, allowing well-informed decisions regarding its capability to generalize, suitability for deployment, and potential areas for improvement. By used the model that was highest performing among the 10 folds and using the test dataset which is the data that we split at first before any training, to ensure that the model will not unseen the dataset for accurate validation.

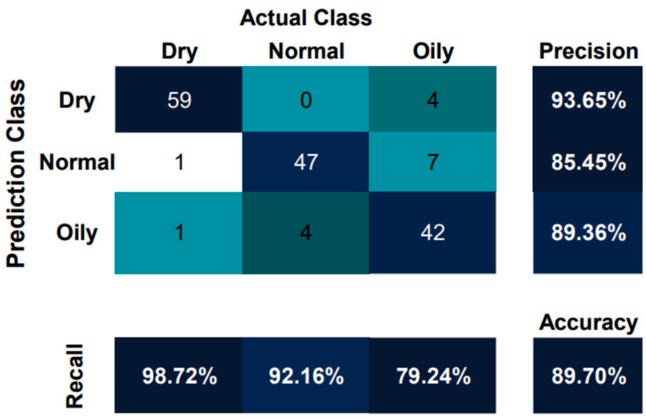


**Fig. 14.** Confusion Matrix from Testing result.

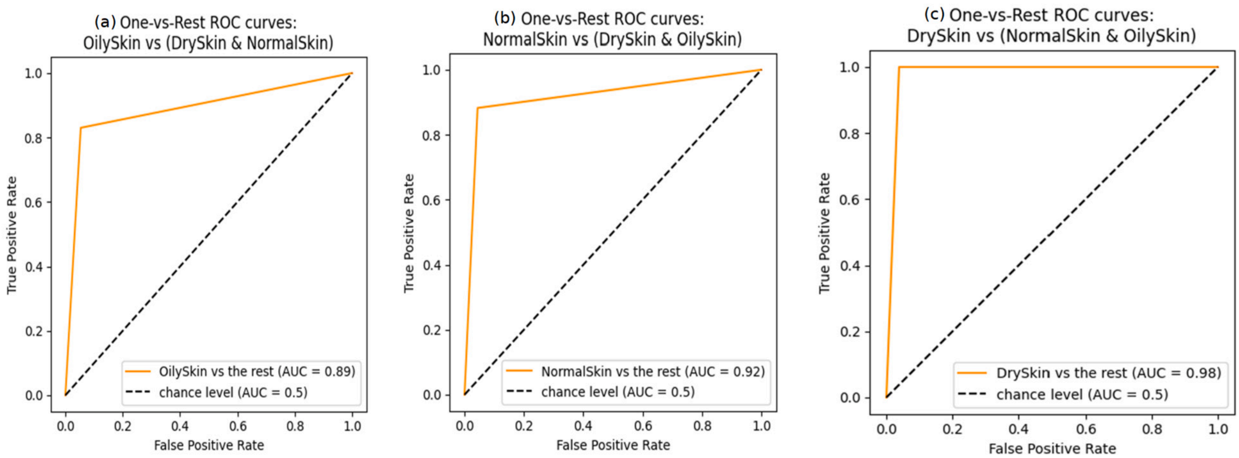


**Fig. 15.** ROC curves from the validation model: (a) oily skin type vs. others, (b) normal skin type vs. others, and (c) dry skin type vs. others.

The model exhibited exceptional predictive capabilities specifically tailored to the classification of dry skin images. From a dataset consisting of 63 images of dry skin, the model accurately identified approximately 59 images as true positives. In 4 images, the model incorrectly classified dry skin as oily skin. For classification outcomes of normal, the model demonstrated a similar level of performance, correctly predicting 47 images as normal skin, with an erroneous prediction in 7 instances by incorrectly classifying images as oily skin (Fig. 14).

From the classification metric, dry skin type was predicted with precision of 93.65% and recall of 98.72%, showing high accurate. For normal skin type, precision was 85.45% and the recall was 92.16% indicating highly accurate predictions. Oily skin type had precision of 89.36%, indicating accurate predictions, and a recall of 79.24%, suggesting that the model struggled to identify oily skin instances. Overall, the model achieved an accuracy of 89.70% and a loss of 21.68%. These classification metrics provided insights into model performance when classifying different skin types, with dry skin type showing the highest precision and recall values.

Overall, the model displays robust predictive abilities when classifying dry skin images, demonstrating remarkable accuracy. The model displayed a relatively consistent performance when classifying normal and oily skin, but made a few misclassifications, particularly when differentiating between oily skin and the other two skin types.

#### *4.2.3. ROC curves*

The model exhibited remarkable capabilities in classifying both oily skin and normal skin, showcasing nearly equal performance as indicated by the comparable area under the curve metrics. The AUC for oily skin was 0.89, while normal skin, reached 0.92. Despite the relatively smaller amount of available data for dry skin compared to the other two skin types, the model successfully leveraged the distinctive characteristics and features specific to dry skin, enabling accurate classification in this category that surpasses expectations, achieving the highest AUC metric of 0.98 across all skin types (Fig. 15).

#### *4.2.4. Comparison with baseline models*

Comparison with the baseline was important to evaluate the effectiveness and superiority of the proposed model, and whether the performance was better or worse than the baseline in terms of validation, accuracy, and validation loss metrics. This comparison

**Table 4**
Proposed model accuracy and loss percentages compared with the baseline model.

| Model | Accuracy | Loss |
|---|---|---|
| EfficientNet-B7 | 77.22 | 61.81 |
| EfficientNet-B5 | 80.64 | 61.21 |
| EfficientNet-B4 | 86.06 | 34.29 |
| EfficientNet-B3 | 86.08 | 35.15 |
| EfficientNet-B2 | 86.66 | 30.90 |
| EfficientNet-B0 | 86.66 | 28.25 |
| **Proposed Model** | **89.70** | **21.68** |

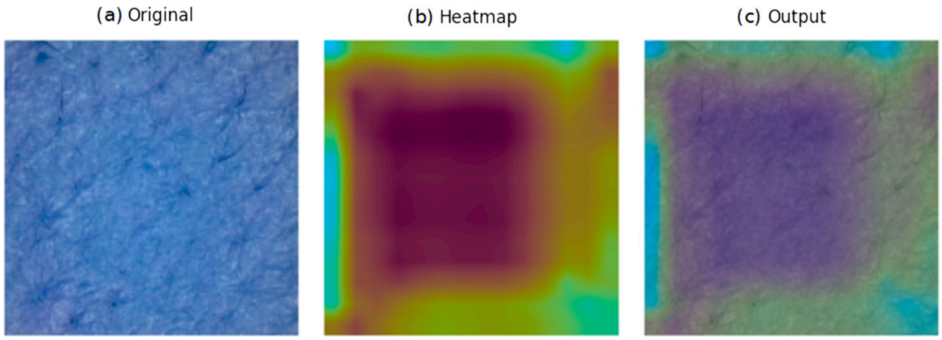


**Fig. 16.** Visualization of GRAD-CAM Interpretation Stages. The figure displays the sequential stages of image interpretation using GRAD-CAM: (a) the original image, (b) the heatmap generated by GRAD-CAM, and (c) the output from the classification model.

determined whether the proposed model offered any significant improvements over the baseline and was suitable for further development or deployment. Model comparison provided insights into the strengths and weaknesses of both models, enabling informed decisions about model selection and potential enhancements.

Table 4 presents validation accuracy and validation loss for EfficientNet-B7, EfficientNet-B5, EfficientNet-B4, EfficientNet-B3, EfficientNet-B2, EfficientNet-B0, and the proposed model. Validation accuracy represents the proportion of correctly classified instances, while validation loss measures the dissimilarity between the predicted and actual values, with lower values indicating better performance. Among the EfficientNet models, EfficientNet-B7 achieved an accuracy of 77.22% with a loss of 61.81%, followed by EfficientNet-B5 with an accuracy of 80.64% and a loss of 61.21%. The proposed model outperformed the EfficientNet models, achieving a higher validation accuracy of 89.70% with a lower validation loss of 21.68%. These results indicate that the proposed model demonstrated improved accuracy and reduced dissimilarity between predicted and actual values compared to the EfficientNet models.

### *4.2.5. GRAD-CAM results*

Results in Fig. 16 showed that the proposed model primarily focused on the center of the image, which contained numerous features such as skin marks, skin oiliness, and overall texture. The heatmap demonstrated red regions as the most crucial areas for analysis, while the blue regions were considered secondary.

## 5. Discussions

We presented in this paper a method for human skin type classification using image proecessing techniques with deep learning models. The strength of this work is that the performance of the proposed approach is highly acceptable and beneficial for commercial usage. The proposed method requires less computing resource which is feasible to be used in mobile application where dermatologists can cosmetologists can utilize this work in practice.

The proposed model showed promising results but it is important to acknowledge certain negative aspects related to bias and dataset imbalance. Despite extensive validations, the model consistently performed better in classifying dry skin compared to normal and oily skin. This performance discrepancy was attributed to biases inherent in the original datasets. The data collection process focused primarily on specific skin areas, which introduced challenges in feature extraction and classification for normal and oily skin types. These skin types share certain attributes, such as texture and the presence of acne, which further complicates the classification task. By contrast, the model demonstrated higher proficiency in classifying dry skin due to the dataset's emphasis on key attributes associated with dry skin, such as prominent skin marks. This uneven distribution of attributes within the dataset contributed to the differential performance of the model across three different skin types.

Imbalanced datasets present several drawbacks, leading to biased model performance that favors, the majority class, resulting in reduced accuracy for minority class predictions. Imbalanced datasets also limit the ability to generalize, making it less effective in handling unseen data. Underrepresentation of the minority class leads to misclassification and an increased number of false negatives or false positives. Evaluation metrics may be misleading, and training models become more challenging, requiring specialized techniques. To address these drawbacks, techniques like resampling, using appropriate evaluation metrics, and employing algorithmic approaches specifically designed for imbalanced datasets can help to improve model performance and mitigate the challenges posed by imbalanced data.

The data collection process is related to the limitations of CNNs which lack spatial understanding, with the inability to effectively capture long-range dependencies [42]. CNNs operate on fixed-size receptive fields and perform local operations. This limits their ability to understand spatial relationships between distant elements in an image or sequence and can be challenging for tasks that require global context such as understanding complex visual scenes or capturing long-term dependencies in sequential data. CNNs are also sensitive to variations in input scale, rotation, and translation which can affect their performance and may require additional techniques such as data augmentation or pre-processing. CNNs are used with high success rates in many computer vision tasks but their limited spatial understanding and inability to capture long-range dependencies are important considerations in certain applications.

A previous study [4] developed a model that classified oily, normal, dry, and combination skin types, with an accuracy of 85%. This accuracy was lower compared to our results but they were able to capture the entire facial area, which helped reduce biased classifications. In a future studies, it would be beneficial to compare the performance of our proposed method with a capsule-based network architecture. Capsule networks have been recently utilized for image classification and offer high potential to preserve spatial relationships of learned features [43,44]. In terms of data collection, it would be advantageous to focus on capturing the entire face to prevent biased classifications towards any specific class.

## 6. Conclusions

Deep learning models developed in this work to classify human skin types (dry, normal, and oily) gave promising results. Some issues were initially encountered such as imbalanced and biased quality in certain datasets but these problems were successfully addressed. Image pre-processing techniques increased the amount of data to tackle the imbalance issue, With enhancement techniques employed to improve the quality of problematic datasets. There were still some biases in the image quality that limited model's performance but overall results remained acceptable.

The original dataset contained fewer samples for dry skin compared to the two other skin types but surprisingly, the model achieved the most promising results when classifying dry skin, suggesting that dry skin has distinct features that can be effectively captured, with normal, and oily skin sharing some attributes that the images may did not fully capture. Our proposed model met all the acceptable criteria.

During evaluation using new, unseen data, the model achieved an accuracy of 89.70%, correctly classifying most skin types. The model showed a loss of 21.68% indicating that it successfully captured relevant patterns and features to distinguish between different skin types. These findings suggested that the model had the potential to accurately predict skin types based on the given characteristics, thereby offering a valuable tool for skincare professionals and individuals seeking personalized skincare solutions.

## Ethics statement

The research was approved by University of Phayao Human Ethic Committee, Thailand, approval number 3/019/58.

## CRediT authorship contribution statement

**Sirawit Saiwaeo:** Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing, Software. **Sujitra Arwatchananukul:** Formal analysis, Funding acquisition, Investigation, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Lapatrada Mungmai:** Data curation, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Weeraya Preedalikit:** Data curation, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Nattapol Aunsri:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## Acknowledgements

The authors are grateful for the grant from Mae Fah Luang University under the National Science, Research, and Innovation Fund (NSRF), Thailand: in part by Mae Fah Luang University under the grant Fundamental Fund/Basic Research Fund, Thailand: 662A03013, and in part by Integrated AgriTech Ecosystem Research Group (IATE) at Mae Fah Luang University, Thailand. The authors would like to thank the School of Information Technology at Mae Fah Luang University for providing facilities to this work.

## References

[1] E. Goceri, Automated skin cancer detection: where we are and the way to the future, in: 2021 44th International Conference on Telecommunications and Signal Processing (TSP), 2021, pp. 48–51.
[2] E. Göçeri, Convolutional neural network based desktop applications to classify dermatological diseases, in: 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), 2020, pp. 138–143.
[3] E. Göçeri, Impact of deep learning and smartphone technologies in dermatology: automated diagnosis, in: 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), 2020, pp. 1–6.
[4] A. Kothari, D. Shah, T. Soni, S. Dhage, Cosmetic skin type classification using CNN with product recommendation, in: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1–6.
[5] E. Göçeri, An application for automated diagnosis of facial dermatological diseases, İzmir Katip Çelebi Üniv. Sağlık Bilim. Fakült. Derg. 6 (2021) 91–99.
[6] R. Dandu, M.V. Murthy, Y.R. Kumar, Transfer learning for segmentation with hybrid classification to detect melanoma skin cancer, Heliyon 9 (2023) e15416, https://doi.org/10.1016/j.heliyon.2023.e15416.
[7] A.A. Kumar, M.V. Krishna Reddy, V.M. Sanjay, V. Kothuru, K.P. Shah, A. Kalucha, Comparative analysis of skin cancer prediction using neural networks and transfer learning, in: 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, 2022, pp. 1518–1521.
[8] R. Shaik, S.K. Bodhapati, A. Uddandam, L. Krupal, J. Sengupta, A deep learning model that diagnosis skin diseases and recommends medication, in: 2022 1st International Conference on the Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PCEMS), 2022, pp. 7–10.
[9] P. Xie, T. Li, J. Liu, F. Li, J. Zhou, K. Zuo, Analyze skin histopathology images using multiple deep learning methods, in: 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), 2021, pp. 374–377.
[10] A.H. Shahin, A. Kamal, M.A. Elattar, Deep ensemble learning for skin lesion classification from dermoscopic images, in: 2018 9th Cairo International Biomedical Engineering Conference (CIBEC), 2018, pp. 150–153.
[11] T. Polevaya, R. Ravodin, A. Filchenkov, Skin lesion primary morphology classification with end-to-end deep learning network, in: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 247–250.
[12] P. Yao, S. Shen, M. Xu, P. Liu, F. Zhang, J. Xing, P. Shao, B. Kaffenberger, R.X. Xu, Single model deep learning on imbalanced small datasets for skin lesion classification, IEEE Trans. Med. Imaging 41 (2022) 1242–1254, https://doi.org/10.1109/TMI.2021.3136682.
[13] S. Vinod, M.V. Thomas, A comparative analysis on deep learning techniques for skin cancer detection and skin lesion segmentation, in: 2021 International Conference on Communication, Control and Information Sciences (ICCISc), vol. 1, 2021, pp. 1–6.
[14] L. Song, J. Lin, Z.J. Wang, H. Wang, An end-to-end multi-task deep learning framework for skin lesion analysis, IEEE J. Biomed. Health Inform. 24 (2020) 2912–2921, https://doi.org/10.1109/JBHI.2020.2973614.
[15] M. Akay, Y. Du, C.L. Sershen, M. Wu, T.Y. Chen, S. Assassi, C. Mohan, Y.M. Akay, Deep learning classification of systemic sclerosis skin using the MobileNet-V2 model, IEEE Open J. Eng. Med. Biol. 2 (2021) 104–110, https://doi.org/10.1109/OJEMB.2021.3066097.
[16] Y. Jiang, J. Dong, Y. Zhang, T. Cheng, X. Lin, J. Liang, PFC-Net: position and context information fusion attention convolutional neural network for skin lesion segmentation, Heliyon 9 (2023) e13942, https://doi.org/10.1016/j.heliyon.2023.e13942.
[17] Z. Zhang, Y. Jiang, H. Qiao, M. Wang, W. Yan, J. Chen, SIL-Net: a semi-isotropic l-shaped network for dermoscopic image segmentation, Comput. Biol. Med. 150 (2022) 106146, https://doi.org/10.1016/j.compbiomed.2022.106146.
[18] E. Goceri, Comparative evaluations of CNN based networks for skin lesion classification, in: 14th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, 5th International Conference on Big Data Analytics, Data Mining and Computational Intelligence and 9th International Conference on Theory and Practice in Modern Computing, 2021.
[19] S. Saiwaeo, L. Mungmai, W. Preedalikit, S. Arwatchananukul, N. Aunsri, A comparative study of image enhancement methods for human skin image, in: 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, 2022, pp. 484–488.
[20] E. Goceri, Medical image data augmentation: techniques, comparisons and interpretations, Artif. Intell. Rev. (2023), https://doi.org/10.1007/s10462-023-10453-z.
[21] E. Goceri, Comparison of the impacts of dermoscopy image augmentation methods on skin cancer classification and a new augmentation method with wavelet packets, Int. J. Imaging Syst. Technol. (2023), https://doi.org/10.1002/ima.22890.
[22] E. Goceri, Image augmentation for deep learning based lesion classification from skin images, in: 2020 IEEE 4th International Conference on Image Processing, Applications and Systems IPAS, IEEE, 2020.
[23] G. Baki, K.S. Alexander, CHAPTER 3. Skin care products, in: Introduction to Cosmetic Formulation and Technology, John Wiley & Son, Inc., 2015, pp. 125–147.
[24] J. Saenpaen, S. Arwatchananukul, N. Aunsri, A comparison of image enhancement methods for lumbar spine X-ray image, in: 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2018, pp. 798–801.
[25] A.M. Reza, Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement, J. VLSI Signal Process. Syst. Signal Image Video Technol. 38 (2004) 35–44.
[26] P. Musa, F. Rafi, M. Lamsani, A review: contrast-limited adaptive histogram equalization (CLAHE) methods to help the application of face recognition, 2018, pp. 1–6, https://doi.org/10.1109/IAC.2018.8780492.
[27] E. Goceri, Evaluation of denoising techniques to remove speckle and Gaussian noise from dermoscopy images, Comput. Biol. Med. 152 (2023) 106474, https://doi.org/10.1016/j.compbiomed.2022.106474, https://www.sciencedirect.com/science/article/pii/S0010482522011829.
[28] E. Göçeri, Intensity normalization in brain MR images using spatially varying distribution matching, in: Proceedings of the International Conferences on Computer Graphics, Visualization, Computer Vision and Image Processing 2017 and Big Data Analytics, Data Mining and Computational Intelligence 2017 - Part of the Multi Conference on Computer Science and Information Systems 2017, 2017, pp. 300–304.
[29] E. Goceri, Fully automated and adaptive intensity normalization using statistical features for brain MR images, Celal Bayar Üniv. Fen Bilim. Derg. 14 (2018) 125–134, https://doi.org/10.18466/cbayarfbe.384729.
[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNet-V2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 4510–4520.
[31] M. Tan, Q.V. Le, EfficientNetV2: smaller models and faster training, https://arxiv.org/abs/2104.00298, 2021, https://doi.org/10.48550/ARXIV.2104.00298.
[32] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.
[34] E. Raff, CHAPTER 3. Convolutional Neural Networks, O'REILLY MEDIA, 2021, pp. 82–92.
[35] E. Raff, CHAPTER 2. Fully Connected Networks, O'REILLY MEDIA, 2021, pp. 31–34.
[36] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 42 (2012) 463–484, https://doi.org/10.1109/tsmcc.2011.2161285.
[37] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305, http://www.jmlr.org/papers/v13/bergstra12a.html.

[38] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second edition, Springer Series in Statistics, Springer, New York, 2009, https://books.google.co.th/books?id=tVIjmNS3Ob8C.

[39] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2006) 861–874, https://doi.org/10.1016/j.patrec.2005.10.010, https://www.sciencedirect.com/science/article/pii/S016786550500303X, ROC Analysis in Pattern Recognition.

[40] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2017, pp. 618–626.

[41] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: a novel bandit-based approach to hyperparameter optimization, J. Mach. Learn. Res. 18 (2017) 6765–6816.

[42] E. Goceri, Analysis of capsule networks for image classification, in: International Conferences Computer Graphics, Visualization, Computer Vision and Image Processing 2021, Connected Smart Cities 2021 and Big Data Analytics, Data Mining and Computational Intelligence 2021, 2021.

[43] E. Goceri, Classification of skin cancer using adjustable and fully convolutional capsule layers, Biomed. Signal Process. Control 85 (2023) 104949, https://doi.org/10.1016/j.bspc.2023.104949, https://www.sciencedirect.com/science/article/pii/S1746809423003828.

[44] E. Goceri, Capsule neural networks in classification of skin lesions, in: Proceedings of the 15th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP 2021), the 7th International Conference on Connected Smart Cities (CSC 2021) and 6th International Conference on Big Data Analytics, Data Mining and Computational Intel, 2021.

**Sirawit Saiwaeo** received B.S. degree in Software Engineering from Mae Fah Luang University, Thailand in 2021. Currently working in 1Moby as an Software development associate and studying for the master's degree in computer engineering at Mae Fah Luang University, Thailand. His research interests include Artificial Intelligence, Machine Learning, and Image Processing.

**Sujitra Arwatchananukul** received the B.S. and M.S. degrees in Computer Science from Chiang Mai University, Thailand, in 2004 and 2008 respectively and Ph.D. degree in Computer Engineering from Yunnan University, China. In 2016, she joined the School of Information Technology, Mae Fah Luang University, as a lecturer. Her current research interests include Data Science, Machine Learning, Data Analysis, Image Processing, Software engineering, Algorithms and Database Management system.

**Lapatrada Mungmai** obtained a B.Sc. degree in Microbiology, M.S. degree in Biology and Ph.D. degree in Pharmacy from Chiang Mai University, Thailand in 2003, 2006 and 2013, respectively. Since January 2014, she has been working as an Assistant Professor of Cosmetic Sciences with the Department of Cosmetic Sciences at the Faculty of Pharmacy, University of Phayao, Thailand. Her research focuses on Industrial Microbiology, Cosmetic Innovation, Clinical studies, and the assessment of biological activity in natural products.

**Weeraya Preedalikit** obtained a B.Sc. Cosmetic Science from Mae Fah Luang University, Thailand, in 2010. In 2013, she earned a M.S. degree in Pharmaceutical Sciences from Chiang Mai University, Thailand. From January 2014 onwards, she has been working as an Assistant Professor of Cosmetic Sciences with the Department of Cosmetic Sciences, Faculty of Pharmacy, University of Phayao, Thailand. Her research focuses on plant extracts, their biological activities, the development of pharmaceutical formulations and natural products, and conducting clinical studies.

**Nattapol Aunsri** received the B.Eng. degree and M.Eng. degree in Electrical Engineering from Khon Kaen University and Chulalongkorn University, Thailand in 1999 and 2003, respectively. He obtained M.Sc. degree in Applied Mathematics and Ph.D. degree in Mathematical Sciences from New Jersey Institute of Technology, Newark, NJ, in 2008 and 2014, respectively. Since October 2021, he has been working as an Associate Professor of Computer Engineering with the School of Information Technology, Mae Fah Luang University, Chiang Rai, Thailand. He is with the Integrated AgriTech Ecosystem Research Group (IATE), and is also with the Computer and Communication Engineering for Capacity Building Research Center, both at Mae Fah Luang University. His research interests include ocean acoustics, Bayesian estimation and filtering, signal processing, biomedical signal processing, machine learning and AI applications, and mathematical and statistical modeling and analysis including computational finance and economic. Dr. Aunsri is a member of IEEE, APSIPA, ECTI, and IAENG.

# Facial Skincare Product Recommendation Using Deep Learning Techniques

**Prof.Mr.Wasim Raja A[1] , Sri Hema N[2], Surya M R[3], Sanjay R[4]**
[1, 2, 3, 4] Dept of Artificial Intelligence and Data Science
[1, 2, 3, 4] Sri Krishna College of Engineering and Technology, Coimbatore

***Abstract-*** *Skincare products are essential cosmetics for women, especially in the modern era. To make the process easier and more effective, an innovative skincare product recommendation system has been developed. the system revolutionizes personalized skincare solutions by seamlessly integrating image processing and advanced deep learning techniques like Efficient net B0. The system goes beyond traditional classifications, accurately identifying diverse skin types, including normal, oily, dry, sensitive, or combination. In return, they receive intricate recommendations tailored not only to their specific skin types but also addressing individual concerns such as acne, pigmentation and dark circles. The suggestions provided are comprehensive, spanning a range of products including cleansers, moisturizers, serums, and more. Beyond being a mere product recommendation tool, the system says the routine to the user. By offering personalized recommendations and valuable skincare education, the innovative system becomes your trusted companion in achieving a skincare routine tailored to your unique needs, fostering a glowing and confident complexion with an over all is accuracy of 92.34%.*

***Keywords***- Skin care, Efficient net B0, Deep Learning, Recommendations, Acne, Skin Types

## I. INTRODUCTION

In the modern world, personalized skincare are more important. The integrated system utilizes cutting edge technology, leveraging Convolutional Neural Networks and Efficient-net B0 transfer learning. the powerful combination ensures the accurate classification of facial images into categories such as Dry, Oily, Normal, Sensitive or combination, while also adeptly identifying specific concerns like dark circles and needs. Remarkably, the model maintains heightened accuracy even when faced with challenges related to image quality.

To enhance its capabilities, the system employs a region-based skin detection method within the HSV and YCbCr color spaces. the innovative approach categorizes skin tones using the six Fitzpatrick scale categories, now extended to include detailed information on dark circles. Acne classification is seamlessly integrated into the system, utilizing a Convolutional Neural Network (CNN) structure with transfer learning. Powered by a specialized dataset, our recommended system employs cosine similarity to provide tailored product suggestions.

These suggestions are intricately aligned with diverse skin metrics, encompassing considerations for dark circles. the holistic framework aims to revolutionize skincare by offering a curated selection of products designed to effectively address specific individual needs. Through the amalgamation of advanced technology and personalized insights, our system aspires to redefine the skincare experience and empower users with targeted solutions for a radiant and healthy complexion.

## II. OBJECTIVES

An innovative skincare product recommendation system, driven by Convolutional Neural Networks and Efficient-net B0 transfer learning, transforms personalized skincare. Seamlessly integrating image processing, it accurately classifies diverse skin types and addresses specific concerns such as acne, pigmentation and dark circles. Users easily engage through a user-friendly interface, receiving comprehensive product recommendations spanning cleansers, moisturizers, serums and more. Beyond product suggestions, the system educates on effective skincare routines and ingredient benefits, empowering users for healthier and more confident skin. With a commitment to open-source principles, the practical system offers personalized recommendations, becoming a trusted companion in achieving a tailored skincare routine. The advanced technology, including region-based skin detection and acne classification, ensures heightened accuracy in assessing individual skincare needs. By revolutionizing the skincare experience, the holistic framework aims to redefine beauty routines, providing targeted solutions for a radiant and healthy complexion.

## III. METHODOLOGY

The primary objective the research t is to develop an advanced Skincare Product Recommendation System, integrating image processing and deep learning techniques.

The system aims to provide personalized skincare recommendations by accurately identifying users' diverse skin types, including normal, oily, dry, sensitive, or combination skin. Beyond the, the system evaluates skin tones and addresses specific skincare concerns like acne, pigmentation, dark circles. To achieve precise skin type identification, employing Convolutional Neural Network (CNN) models, incorporating transfer learning with state-of-the-art architectures such as EfficientNet B0. the ensures a robust and accurate analysis of individual skin characteristics. Additionally, implement a region-based skin detection method based on color segmentation and clustering to determine skin tone, adding a comprehensive layer to the system's understanding of users' unique skincare profiles. The heart of our recommender system lies in a specialized dataset that links skincare products to specific attributes. the dataset allows us to offer users a tailored selection of products aligned with their individual skincare needs.

A. Dataset Collection & Preprocessing

Collected 1800 skin issue images, enhancing quality through noise removal and exposure normalization. Data augmentation diversified the dataset for robust model training.

B. Feature Extraction

Customer demographics and product attributes inform feature extraction. Demographics like age and skin type, alongside product details, tailor recommendations. External data integration enhances personalization.

C. Skin Type Detection

Utilized a CNN with Efficient Net B0, extracting color, texture, and statistical features for skin type classification. The model categorizes skin types such as oily, normal, dry, and sensitive.

D. Product Recommendation Engine

A content-based recommendation system prioritizes ingredient similarity within product categories, ensuring personalized skincare suggestions aligned with individual skin types.

E. Cosine Similarity Of Products

Utilized t-SNE to visualize ingredient similarities, facilitating effective product comparisons. Cosine similarity aids in assessing ingredient alignment between products, enabling tailored skincare routines.

F. Matrix Factorization

Matrix Factorization simplifies the product recommendations considering the user input and skin concerns. The method suggests products based on similarities in brand, skin type and ingredients.

## IV. ARCHITECTURE

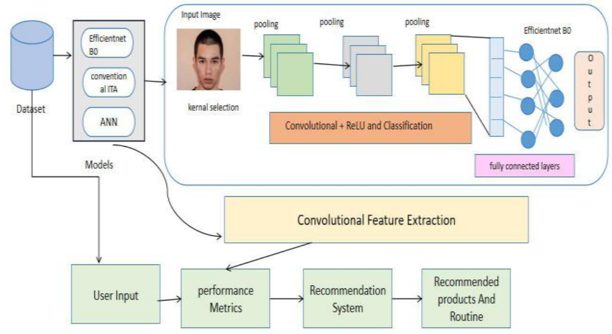


Fig.1. Proposed System Architecture

The architecture of the facial skincare products recommendation system with deep learning begins with user input, where a facial image is either captured by a camera or uploaded from a device. The heart of the system lies in the Efficient net BO convolutional neural network (CNN) architecture, specifically designed for efficient feature extraction from facial images. Convolutional layers within the architecture analyze the input image, identifying intricate facial features like the nose, eyes, mouth and the overall facial structure.

### A. CONVOLUTION NEURAL NETWORK

CNN is a specialized type of neural network designed for image-related tasks. It excels in learning hierarchical features from visual data, making it well-suited for tasks like facial skincare and skin type classification. In the Facial Skincare Recommendation System, a CNN is employed for facial lesion detection, allowing the system to identify key features such as acne, dark circle and pigmentation . the information is crucial for assessing specific areas of the face, understanding facial structure and providing targeted skincare recommendations.

### B. Efficient Net B0

Efficient Net is a family of neural network architectures that are known for their efficiency in terms of model size and computational resources while maintaining high accuracy. Efficient Net B0 is the baseline model of the family. It systematically scales the network in multiple dimensions, balancing model depth, width and resolution. In the Facial Skincare Recommendation System, Efficient Net B0 is used for skin type classification. By extracting features from skin images, including color, texture and statistical characteristics, the model classifies skin types into categories such as oily, normal, dry and sensitive. Efficient Net B0's efficiency is particularly beneficial for deploying the model in real-world scenarios, ensuring that it can run efficiently on various devices and platforms.

**C. Facial Acne Detection (CNN)**

The Convolutional Neural Network (CNN) is instrumental in identifying and detecting facial acne in the Facial Skincare Recommendation System. By training the CNN on a dataset that includes images annotated with acne regions, the model becomes adept at recognizing patterns and features indicative of acne lesions. The CNN's role is crucial in precisely pinpointing the location and severity of acne on the face, enabling a detailed analysis of skin conditions. the acne detection capability enhances the system's ability to tailor skincare recommendations by targeting specific areas affected by acne, providing users with more focused and effective skincare advice based on their individual needs.

**D. Skin Type Classification**

Efficient Net B0, on the other hand, contributes to the understanding of skin types. By leveraging color, texture and statistical features extracted from skin images, the model classifies users into different skin types. the classification forms the basis for recommending skincare products and routines tailored to the individual's specific skin characteristics and needs.

**E. Personalized Recommendations**

The combined role of CNN and Efficient Net B0 enables the system to provide personalized skincare recommendations. The facial landmarks detected by the CNN help in understanding facial structure, while the skin type classification ensures that the skincare routine suggestions are customized based on the user's unique skin attributes.

## V. DATA AUGMENTATION

Data augmentation in skin tone, type and acne predictions involves artificially diversifying the training dataset by applying transformations such as rotation, flipping and color adjustments. the process increases the model's exposure to varied skin tones, enhancing its ability to generalize and predict accurately. Augmentation aids in mitigating bias and improves the model's robustness to different lighting conditions and skin variations.

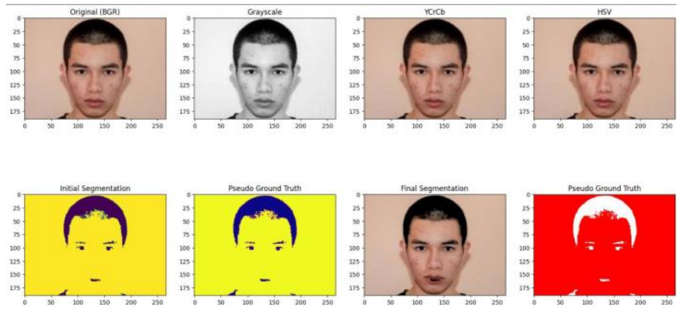


Fig.2. Data Augmentation

**1. Original (BGR) - Color Information**

The Original (BGR) representation allows for the analysis of different skin tones present in the image based on variations in the blue, green and red channels. the color information can be crucial for distinguishing between individuals with different complexions and identifying regions with specific tones associated with various skin types.

**2. Grayscale - Intensity Analysis**

The Grayscale version simplifies the image to a single intensity channel, providing insights into the overall brightness and darkness of different regions in the original image. the can be valuable for detecting subtle variations in skin tone, identifying areas with higher or lower pigmentation and potentially revealing patterns associated with certain skin conditions.

**3. YCrCb - Luminance and Chrominance Separation**

YCrCb separation allows for a more nuanced analysis of skin features. The Y channel, representing luminance, can highlight variations in skin brightness associated with different skin types. The Cr and Cb channels, capturing chrominance information, can be useful for discerning color variations indicative of diverse skin tones and identifying areas of interest related to skin conditions.

**4. HSV - Hue, Saturation and Value**

The HSV representation is particularly valuable for analyzing color-related information. Hue can help identify specific skin tones, saturation can reveal the intensity or vividness of those tones and value can provide insights into the brightness of the skin. the breakdown can be beneficial for distinguishing skin types, detecting anomalies in coloration and assessing the severity of skin conditions such as acne.

**5. Initial Segmentation - Skin Regions Identification**

The Initial Segmentation results demonstrate how the segmentation algorithms have initially divided the image into different regions. In the context of skin analysis, the step can help identify regions of interest related to different skin types, tones, or potential areas affected by acne.

**6. Pseudo Ground Truth - Manual Reference for Skin Attributes**

The Pseudo Ground Truth serves as a manually created reference for desired segmentation results. In the context of skin analysis, it can be designed to highlight specific characteristics such as skin types, tones and acne-affected areas. the serves as a benchmark for evaluating the accuracy and effectiveness of the segmentation algorithms in capturing relevant skin attributes.

**7. Final Segmentation - Refined Results**

The Final Segmentation represents the refined outcome of the segmentation algorithm after adjustments based on the Pseudo Ground Truth. the step is crucial for enhancing the accuracy of the segmentation process, ensuring that the algorithm aligns closely with the desired identification of skin tones, types and acne-affected regions.

## VI. IMAGE HISTOGRAM

The process of skin detection involves several key steps to accurately identify skin pixels within an image. Initially, segmentation is performed, followed by the prediction of skin pixels and k-means clustering.

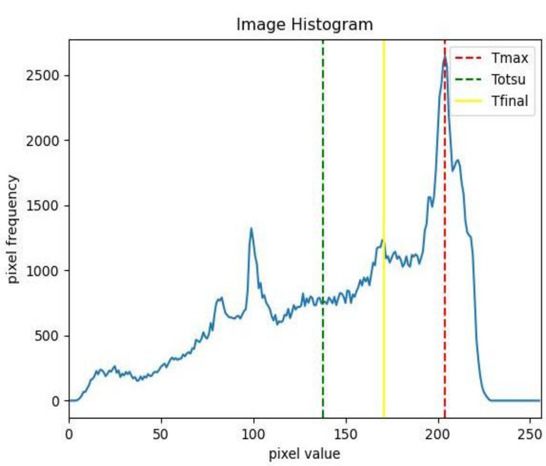


Fig.3. Image Augmentation

The initial segmentation begins with thresholding the grayscale image using a specific threshold value, which is calculated as the average of TOTSU, TFINAL and TMAX obtained from the image histogram. the thresholding process helps in separating the foreground (skin) from the background. For images with a resolution of 224 by 224 pixels, the segmentation ensures that each pixel within the image is individually evaluated based on its grayscale intensity.

The HSV and YCrCb color spaces, potential skin color pixels are selected based on predefined criteria. Typically, these criteria involve specific ranges for the Hue, Cr, and Cb components. For example, skin pixels may be identified if (Hue <= 170) and (140 <= Cr <= 170) and (90 <= Cb <= 120). These criteria ensure that only pixels within the specified color ranges are considered as potential skin pixels, allowing for precise detection within the image.

## VII. RECOMMENDATION SYSTEM

The model needs to know the user's skin features to deliver the products corresponding to the top values of similarity (skin vector, product vector) for the items in the dataset that are classified into that particular category. the can be seen in the figure, It would be an intelligent move to search for products with features compatible with the skin measurements and concerns of the consumer. The user's automated cosine similarity between the user skin attribute vector and the product feature vector is used to determine the similarity of ingredients between products, the t-SNE technique is employed, leading to the reduction of the dimensionality in the data. By preserving the similarities between instances, t-SNE effectively visualizes high-dimensional data on a two-dimensional plane. Similarities are calculated based on the distances between data points and cosine similarity is used to find similarities between non-zero vectors. Unlike distance-based measures, Cosine Similarity captures more information about vector direction. In the developed system, the technique is applied when the user selects a known brand on the recommended system. The system then analyses ingredient similarities based on skin type and skin concern and recommends a complete skincare routine with up to five products in each category, based on t-SNE and Cosine Similarity.

## VIII. CONCLUSION

In summary the Facial Skincare Recommendation System represents a comprehensive and intelligent solution for personalized skincare guidance. Leveraging Convolutional Neural Network (CNN) models, including Efficient Net B0, the system excels in facial skincare recommendations system and skin type classification. By extracting features from diverse facial images, it provides accurate insights into individual skincare needs, allowing for the formulation of tailored product recommendations. The system's success lies in its ability to handle various scenarios, including different skin types, lighting conditions and the presence of makeup or accessories. The inclusion of test cases ensures the robustness and reliability of the model across diverse real-world situations. It emerges as a valuable tool for users seeking personalized and effective skincare routines.

## REFERENCES

[1] Dong X, Yan Y, Ouyang W, "Style Aggregated Network For Facial Landmark Detection," IEEE Conference on Computer Vision and Pattern Recognition, 18 to 23 June-2021; pp.379-388.

[2] Goindi, S., Thakur, K., & Kapoor, D. S. (2023). "Skin Disease Classification and Detection by Deep Learning and Machine Learning Approaches."

[3] Iwabuchi, R., Nakajima, Y., et al. (2023). "Artificial Intelligence based Smart Cosmetics Suggestion System based on Skin Condition."

[4] Lee, Gyeongeun (2020). "A Content-based Skincare Product Recommendation System."

[5] Li, Yi, Song, Lingxiao, Wu, Xiang, He, Ran. "AntiMakeup Learning A, B –Level Adversarial Network For Makeup Invariant Face Recognition," AAAI Conference, 2 to 7 Feb-2020; pp.79-388.

[6] Liu, Xudong, Taoli, Iris Chuoying Ouyang, and Ruizhe Wang. "Understanding Beauty via Deep Facial Features," IEEE Conference, volume-246-256.

[7] Ma, X., Lu, J., Liu, X., & Kuang, H. (2022). "Skincare Recommender System Using Neural Collaborative Filtering with Implicit Rating."

[8] Rathgeb, Christian, Dantcheva, Antitza, Busch, Christoph. "Impact Of Facial Beautification On Face Recognition," IEEE Access 2021,7,152667-152678.

[9] Saiwaeo, S., Arwatchananukul, S., Mungmai, L., Preedalikit, W., & Aunsri, N. (2023). "Human Skin Type Classification Using Image Processing and Deep Learning Approaches."

Paper Acceptance Mail From IJSART.com External Inbox x

Ijsart Journal <editor.ijsart@gmail.com>
to me

Mon, Mar 18, 1:00 PM (3 days ago)

**Stay Safe !**

**Dear Author,**

We are thankful for your contribution to Research Community of IJSART.

We are pleased to inform you that your paper.

**Paper ID:** IJSARTV10I383627

**Title of the Paper:** Facial Skincare Product Recommendation Using Deep Learning Techniques

**Author's Name:** A1 : Sri Hema N A2 : Mr. Wasim Raja A A3 : Surya M R A4: Sanjay R

has been Accepted for publishing in IJSART, which will be published in **IJSART Volume 10,Issue 3, March 2024**

**Congratulations!**

ISSN [ONLINE] : 2395-1052

# THE BOARD OF

IJSART

INTERNATIONAL JOURNAL FOR SCIENCE AND ADVANCE RESEARCH IN TECHNOLOGY

is here by awarding this certificate to

SRI HEMA N

In recognition of publication of the paper entitled

FACIAL SKINCARE PRODUCT RECOMMENDATION USING DEEP LEARNING TECHNIQUES

Published in E-Journal

Volume 10,Issue 3 in March 2024

EDITOR IN CHIEF

PAPER ID : IJSARTV10I383627

Email id : editor@ijsart.com | website : www.ijsart.com

IJSART

ISSN [ONLINE] : 2395-1052

THE BOARD OF

IJSART

INTERNATIONAL JOURNAL FOR SCIENCE AND ADVANCE RESEARCH IN TECHNOLOGY

is here by awarding this certificate to

SURYA M R

In recognition of publication of the paper entitled

FACIAL SKINCARE PRODUCT RECOMMENDATION USING DEEP LEARNING TECHNIQUES

Published in E-Journal

Volume 10,Issue 3 in March 2024

PAPER ID : IJSARTV10I383627

EDITOR IN CHIEF

Email id : editor@ijsart.com | website : www.ijsart.com

IJSART
ISSN [ONLINE] : 2395-1052

# THE BOARD OF IJSART

INTERNATIONAL JOURNAL FOR SCIENCE AND ADVANCE RESEARCH IN TECHNOLOGY

is here by awarding this certificate to

SANJAY R

In recognition of publication of the paper entitled

FACIAL SKINCARE PRODUCT RECOMMENDATION USING DEEP LEARNING TECHNIQUES

Published in E-Journal

Volume 10,Issue 3 in March 2024

EDITOR IN CHIEF

PAPER ID : IJSARTV10I383627

Email id : editor@ijsart.com | website : www.ijsart.com