FaceCare AI
Facial Skin Disease Detection and Localization using Deep Learning

MOIZ SHAHID
01-135221-028
ZOHA TAHIR
01-135221-185

Supervisor: Mam Farwa Kazmi

Bachelor of Science in Information Technology

Department of Computer Science
Bahria University, Islamabad

2025

Abstract

Skin diseases plague a large segment of the global community and usually need a lot of care and personal attention. This project introduces a web-based Facial Skin Disease Detection and Skincare Recommendation System which uses deep learning and image processing algorithms to diagnose popular skin conditions and provide appropriate skincare recommendations. The system concentrates on the five main classes, which include Acne, Wrinkles, Vitiligo Disease, Rosacea, and Pigmentation, based on pictures in the Acne04 and Skin-Condition datasets.

MobileNetV2, EfficientNetB3, and a custom Convolutional Neural Network (CNN) were trained and evaluated as three deep learning models. MobileNetV2 was chosen as the main model among them because it has a high accuracy and real-time image classification performance. The model works with images of faces in order to identify skin conditions and activate a customized recommendation engine of appropriate skincare products and ingredients.

The app is developed using ReactJS (frontend) and Python FastAPI (backend) and a MySQL database to store user information, prediction history, and product suggestions. Being lightweight and strong enough, this architecture is cross-platform and can be extended in the future.

This system seeks to fill the knowledge gap between dermatological visit and self-care by enabling users to have immediate skin assessment of the skin and make informed product decisions. It has the potential to be used in the management of personal skincare as well as supplementary to teledermatology services.

Acknowledgments

Almighty Allah, the most Gracious and Merciful, can be credited with all the praises and thanks, that he has given us the power, patience, and wisdom to go through with this Final Year Project and thesis. It is impossible to imagine that this could have been achieved without His guidance and numerous blessings.

To our treasured supervisor, Mam Farwa Kazmi, we extend our utmost thanks to her as she was instrumental in guiding and supporting us during this process and furthermore, motivating us to go ahead with it. Her guidance and wisdom were of great help to our work and development.

We would like to thank the faculty and staff of the Department of Information Technology, Bahria University, Islamabad, to provide us with the opportunity to pursue our education and learn in a favorable academic environment. Particularly, the Project Evaluation Committee has provided valuable feedback and a critical evaluation at every step of the project.

We also recognize the tools in terms of resources datasets, libraries, forums and research articles that helped us in the process of development, particularly the open-source community.

We are blessed to the families with our utmost gratitude to their unequivocal love, tolerance, and their unending support. We also owe a debt of gratitude to our peers and friends who were able to support us in this project, collaborate, and provide a positive spirit.

Lastly, we are thankful to be given the chance to pursue our passion in technology and research in the project. We will take the lessons and experiences we have acquired in our future activities.

Thank you all.

MOIZ SHAHID AND ZOHA TAHIR
Islamabad, Pakistan

December, 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

RAG	Retruval Augmented Generation
LLM	Large Language Model
AI	Artifical Intellegence
SQL	Structured Query Language
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Overview

The health of the skin is vital to the general well-being and looks of an individual. Nevertheless, there is the risk of limited access to dermatological care in time because of the excessive costs, the large waiting lists, or geographical problems. As the use of Artificial Intelligence (AI) in the field of healthcare develops, smart systems are coming up to assist in early diagnosis and offer self-care solutions. In that regard, the given project is called FaceCare AI (Facial Skin Disease Detection and Localization using Deep Learning), and its aim is to identify and localize typical skin diseases on the face with the help of deep learning and prescribe appropriate skincare with the help of a web-based application.[1]

The system targets five predominant and aesthetically distinguishable skin conditions, which are Acne, Wrinkles, Vitiligo Disease, Rosacea, and Pigmentation. Two datasets of Skin-Condition and Acne04 related to dermatology were used to create an efficient AI model. These data include marked facial pictures of a wide variety of skin cases in different age groups and skin complexions.

Three deep learning algorithms were deployed and tested in regards to their functionality, which included MobileNetV2, EfficientNetB3, and a designed Convolutional Neural Network (CNN). These models were trained with preprocessed and augmented images to improve the generalization and performance. MobileNetV2 was chosen as the last model of them as it can be characterized by high accuracy, lightweight design, and real-time inference on the web application.

The complete system is built on a current and scalable technology stack of web. FastAPI, a high-performance Python framework, is used to implement the backend, which would be the most suitable choice to deploy machine learning models over APIs. The frontend is written in ReactJS, and it offers user-friendly interface and responsiveness where the user is enabled to capture or upload facial images.[2] The user data, the history of prediction and the records of skincare products are managed with the MySQL database.

After a picture is posted, the model of choice allows the picture to be processed to identify and categorize the picture as any of the five supported conditions. This outcome is then inputted into the recommendation engine that proposes a list of skincare products and ingredients that would benefit the diagnosed state. This system does not only assist the users to learn more about their skin but also provides product advice depending on their unique needs.

FaceCare AI is a valuable addition to the field of technology and skincare, with a combination of AI-powered diagnosis and customized recommendation systems. It is a step in the direction of more accessible dermatological screening and facilitates the early detection and self-care. The system is also designed in a modular and scalable way, which allows the potential opportunities of expanding it in the future, such as mobile version, supporting additional skin diseases and allowing the system to be integrated with e-commerce platforms to purchase the products directly.

1.2 Problem Statement

One of the most prevalent health problems that affect the world is skin diseases that occur in millions of people of all ages, gender or ethnicity. Conditions of the facial skin, especially, do not only physically affect the way one looks but may also cause psychological consequences like low self-esteem, anxiety, and shyness. Such diseases as acne, wrinkles, vitiligo, rosacea, and pigmentation are common and may need early diagnosis and regular treatment to avoid further development. Nevertheless, access to dermatological services on time is still a problem to most individuals because of the high cost of consultation, absence of close experts, long queues, and poor knowledge regarding skin conditions.[3]

In most situations, people will use internet searches, unproven home remedies, or over the counter products without knowing what is causing the problem, and this may result in misdiagnosis, treating the skin problem wrong, and the deterioration of the skin problem. It is increasingly required to have smart and usable and affordable tools that can aid in early identification of problems in the facial skin and give knowledgeable advice to the users.

In the current development of computer vision and artificial intelligence, especially deep learning, it is possible to create systems capable of processing facial images and detecting skin diseases in real-time. The available solutions are however, in most cases, limited in terms of scope, hard to access, not precise or are not being tailored to the user.

The market currently has a gap of having a lightweight and web-based system that can not only identify and localize facial skin conditions with a high level of accuracy, but also provide effective skincare suggestions depending on the diagnosed condition. Moreover, they have solutions which do not often integrate image classification with user interactivity and experience a fluid recommendation engine within one platform.[4]

In order to overcome this issue, the proposed system is FaceCare AI that will offer an affordable and smart web-based application that will enable people to post facial images, identify typical skin problems, and obtain customized product/ingredient suggestions. The proposed system aims to enable users to manage their skin health, minimize reliance on unreliable information, and fill in the gap between dermatological treatment and everyday skincare, combining disease detection on the basis of deep learning with skincare advice.

Also, the purpose of this project is to make contributions to the sphere of medical image analysis and teledermatology through the implementation and comparison of numerous deep learning models to select the most effective one to be deployable in the real world, and develop a scalable system that would be viable to handle additional skin conditions in the future.[5]

1.3 Research Questions

The study examines the technical performance, generalizability and deployability of an end-to-end neural network-based deep learning model of facial skin disease segmentation and customized recommendation generation. The research questions of the study include the following key research questions:

- **How well can deep learning models detect, classify and localize facial-skin diseases (acne, wrinkles, vitiligo, rosacea and pigmentation) on a static facial image?**
- **What deep learning model between MobileNetV2, EfficientNetB3 and a custom CNN gives the optimal balance of accuracy in classification, localization and runtime web execution?**
- **How efficiently can the prediction performance of the chosen model be incorporated into a web-based platform (FastAPI + React) and provide real-time disease detection, region identification, and individual recommendations of skincare products without violating the privacy and security of users?**

1.4 Research Objectives

The major point of this study is to develop and execute a smart web-based application that employs deep learning to detect and locate facial skin illnesses and provide individualized skincare suggestions. To realize this, the research has the following detailed objectives that are categorized into the key components of the project:

1.4.1 Model Development and Optimization

The subsection is dedicated to deep learning model development, training, and optimization towards the proper classification of facial skin diseases. It involves the choice of suitable architectures, preprocessing of dataset images, training models of labeled datasets and performance measurement with conventional metrics.[6] The aim is to define the most appropriate model that can be very accurate, low latency, and efficient in real-time web environment. The focus is particularly on the tuning of hyperparameters and the comparison of model performance to identify the most appropriate one in which to deploy the model.[7]

1.4.2 System Architecture and Web Integration

System architecture is important in the provision of smooth user experience and effective backend functionality. The design and development of the web-based platform, frontend, and backend as well as database integration is discussed in this subsection. FastAPI will be deployed to host the trained model and accept user inputs, whereas a user-friendly interface will be offered in the frontend that is created with ReactJS. The data on users, their predictions, and product recommendations will be stored in a MySQL database in a secure and a well-structured manner. This integration guarantees real time processing, responsiveness and scalability of the system.[8]

1.4.3 Skin Condition Detection and Localization

The following sub-section addresses the main functionality of the system: detection and classification of a number of different facial skin conditions. It is oriented to make the AI model able to distinguish five particular skin diseases correctly acne, wrinkles, vitiligo, rosacea, and pigmentation. Besides the classification, there is a need to investigate visual localization methods that may be used to draw attention to the affected regions of the face of the user. This improves user confidence and allows much valuable visual feedback regarding the diagnosis.[9]

1.4.4 Recommendation Engine and Skincare Guidance

When a skin problem is diagnosed, its users require practical information. The subsection is related to the creation of a skincare recommendation engine according to which the user receives a specific product or ingredient recommendations depending on the identified disease. It tries to link diagnostic outcomes with a pre-existing database of pre-packaged treatments or products. The recommendation system will also act as an online skincare advisor in order to guide users on how to handle their skin issues with the necessary guidance.[10]

1.5 Research Significance

The creation of FaceCare AI (Facial Skin Disease Detection and Localization with Deep Learning) can be of significant importance in diverse areas. It incorporates artificial intelligence, computer vision, and web technologies to deliver a powerful, easy to use, and intelligent platform of detecting facial skin conditions and recommending skincare. The significance of this research can be categorized as follows:[11]

1.5.1 Academic Significance

This study has made a contribution to the academic development in such areas in computer science and artificial intelligence as the implementation of advanced deep learning models to a healthcare problem. It also connects the gap between the theoretical and the practical, providing future researchers with a base to investigate the analysis of medical images, the implementation of models, and the entire stack of AI.[10]

1.5.2 Technical Significance

The project illustrates that current, light-weight models such as MobileNetV2 could be optimized and implemented successfully with the use of the FastAPI to provide backend services, ReactJS to deliver a frontend interface, and MySQL to serve as a data repository. It also demonstrates system convergence, real-time inference, API manipulation, and the way in which AI can be utilized in production-scale applications at high efficiency.

1.5.3 Practical Significance

The project shows that the latest, lightweight frameworks such as MobileNetV2 can be optimized and implemented to use FastAPI backend services, ReactJS frontend interface, and MySQL database. It, further, demonstrates the concept of integration of systems, live inference, API processing and the efficiency of AI implementation in production-scale applications.[12]

1.5.4 Social and Healthcare Impact

The system assists in encouraging dermatological awareness and self-care particularly among underserved or rural patients. Introducing early detection of skin problems, it facilitates preventative healthcare and minimizes the use of self-medication and non-verified resources. It also promotes digital health literacy among the common users.

1.5.5 Economic and Industrial Significance

FaceCare AI has a high commercialization potential. Its engine recommender can be incorporated with skincare retailers, e-commerce, or telehealth. This opens up a business opportunity to the beauty, health technology, and digital trade industries to take advantage of customized AI applications to connect with customers.[13]

1.5.6 Ethical and Privacy Significance

Since facial data is sensitive, the project encompasses critical ethical issues regarding image storage, privacy of the user, and the responsible use of AI. The system emphasizes the issue of safe image manipulation, anonymity in data processing, and publicly disclosing the AI predictions, which are decisive in developing user confidence in healthcare applications.

1.5.7 Educational and Skill Development Significance

The project is a full-fledged learning experience in full-stack AI development, enabling the students and young researchers to have practical experience in such areas as data preprocessing and deep learning, web frameworks and database architecture, and human-friendly system design. It proves that interdisciplinary skills may be united to address the real life problems using technology.[14]

1.6 Research Limitations

Despite the successful development and implementation of FaceCare AI, several limitations were encountered throughout the research process. These limitations span across multiple domains, including dataset availability, model design, system implementation, technical infrastructure, and ethical considerations. Understanding these limitations is essential to accurately assess the system's current capabilities and to propose strategies for future improvements.

1.6.1 Dataset Limitations

The effectiveness of any AI-based detection system is largely dependent on the quality, diversity, and annotation of the datasets used during training. In the case of FaceCare AI, several constraints related to the datasets (Acne04 and Skin-Condition) were identified.

1.6.1.1 Skin Tone and Ethnicity Imbalance

The datasets used primarily consist of images of individuals with fair or medium skin tones. There is limited representation of darker skin tones or varied ethnicities. This imbalance may cause the model to underperform on users from underrepresented demographics, resulting in biased predictions.

1.6.1.2 Age and Gender Representation Gaps

Most images in the datasets belong to young adults, with very few instances of elderly individuals or children. Additionally, gender distribution is not balanced, which may influence the detection of age-specific or gender-specific skin conditions such as wrinkles or hormonal acne.

1.6.1.3 Low Resolution and Lighting Variations

Some images are blurry, poorly lit, or have inconsistent resolutions. This affects the model's ability to accurately learn patterns, especially in distinguishing subtle skin issues like pigmentation or rosacea.

1.6.1.4 Background Noise and Obstructions

The datasets include images with various backgrounds and occlusions (e.g., hair, makeup, glasses), which introduces noise and decreases classification accuracy. The absence of background segmentation or cropping worsens this issue.

1.6.1.5 Missing Localization Information

The datasets provide only image-level labels without any localization data such as bounding boxes or segmentation masks. This limits the model to classification tasks only and prevents it from learning to localize the affected regions on the face.

1.6.1.6 No Multi-Label Annotations

Each image in the dataset is associated with only one disease label. However, real-world users may exhibit multiple conditions simultaneously (e.g., acne and pigmentation), which the model currently cannot detect or handle.

1.6.2 Model Limitations

While deep learning models such as MobileNetV2 are efficient and accurate, they are not without trade-offs. Some model-related constraints were identified during development and deployment.

1.6.2.1 Fixed Disease Categories

FaceCare AI is trained to detect only five facial skin conditions: acne, wrinkles, vitiligo, rosacea, and pigmentation. This limited scope prevents the model from recognizing other common diseases such as eczema, psoriasis, or melasma.

1.6.2.2 Absence of Severity Classification

The model only detects the presence or absence of a skin condition but does not assess its severity (e.g., mild, moderate, or severe acne). This limits the clinical usefulness of the predictions.

1.6.2.3 Lightweight Model Constraints

MobileNetV2 was chosen for its balance between speed and performance.However, being a lightweight model, it has fewer layers and limited depth compared to heavier architectures like ResNet or DenseNet. This may result in slightly lower precision in complex or borderline cases.

1.6.2.4 Risk of Overfitting

Despite applying data augmentation and regularization techniques, the model shows signs of overfitting during training, especially due to the relatively small and imbalanced dataset. This affects its ability to generalize to unseen images.

1.6.3 Generalizability and Real-World Testing

1.6.3.1 Limited User Testing

The system was tested on a small set of users within a controlled academic environment. Broader testing on diverse populations is needed to ensure reliability across different demographics and real-life use cases.

1.6.3.2 Language and Localization

Currently, the system supports only English. Users who speak other languages may find it difficult to interact with the interface or interpret the results. This limits its reach in multilingual regions.

1.6.4 Summary Of Limitations

Table 1.1: Summary of Research Limitations

Category	Subcategory	Description
Dataset	Skin Tone Bias	Predominantly light skin tones in datasets
Dataset	Missing Localization	No bounding boxes or masks for training localization tasks
Dataset	Single-label Images	Cannot detect multiple skin diseases in one image
Model	Fixed Number of Classes	Only 5 disease classes supported
Model	Limited Depth of MobileNetV2	Lower precision in complex conditions
Technical	Server Limitations	No cloud hosting or scalable backend
Technical	No Mobile Support	Only desktop/web support; no Android/iOS versions
Ethical	Image Privacy	Images not encrypted; sensitive user data handling
Medical	No Clinical Validation	Not approved by dermatologists or healthcare providers
Usability	Limited Testing	Lack of testing on diverse population groups

1.6.5 Summary

Although FaceCare AI is a promising direction in the field of AI-based skin care assistance, the drawbacks outlined in this chapter give the important insight into the spheres which need additional research and development. The mitigation of these shortcomings in subsequent releases will result in a more reliable, fair, scaled and clinical relevant system that will be more applicable in the real world, both in the consumer and healthcare sector.

Chapter 2

Literature Review

2.1 Background

The past years have witnessed a clear upsurge in the cases of individuals with different skin ailments of the face such as acne, pigmentation, vitiligo, rosacea, and wrinkles. These are skin issues that do not only impact on the physical appearance of the individual but also have a much psychological and emotional effect, which in most cases, results in low self esteem, social anxiety, and depression. Although these conditions are widespread, dermatological services are also a challenge in most parts of the world because of various reasons including cost, accessibility, ignorance, and limited availability of special dermatologists especially in rural or underserved communities.

Usual procedures of diagnosing facial skin diseases include a visit to a dermatologist who performs clinical examination, dermatoscopic investigation, or biopsy that is time-consuming, costly, and not available to the general public. Besides, in emerging markets, individuals have a tendency to diagnose themselves by internet search or using over-the-counter medications, and this may result in improper medication, exacerbation of the disease, or permanent skin destruction. This has resulted in a great demand of alternative digital health solutions to fill this gap in accessible, affordable and accurate skin diagnosis.[15]

With the introduction of Artificial Intelligence (AI) and especially the computer vision field and deep learning, more people are interested in creating automated systems that can help with early detection and classification of skin conditions. Recent developments in convolutional neural networks (CNNs), lightweight neural networks such as MobileNetV2, and image classification methods have allowed building powerful and yet efficient models that can detect complicated skin diseases based on the face image. These AI systems are capable of handling visual data with great accuracy and provide a solution that can be scaled to massive application.

Simultaneously, the combination of AI technologies with web development frameworks has set fresh opportunities in the construction of intelligent web-based platforms of diagnostic. The latest technology (FastAPI, ReactJS, MySQL, etc.) enables the deep learning model to be easily integrated into user-friendly interfaces, as the end users can now engage with the system in real-time. Such systems have the ability to enable users to have more control over their skincare routines, as well as serve as a first-line screening tool before professional assistance is sought.[16]

This Final Year Project, which is called FaceCare AI: Facial Skin Disease Detection and Localization using Deep Learning, will fill these gaps by creating an all-encompassing system that unites computer vision, deep learning, and modern web technologies to identify five prototypical facial skin conditions acnes, wrinkles, pigmentation, vitiligo and rosacea on the basis of the facial images, which are sent in by users. The system also offers skincare products recommendations, and ingredient recommendations depending on the condition being diagnosed. There is also an option of a chatbot that helps users with simple skincare questions, which will make the tool a full-fledged AI-based platform to handle the facial skincare.

This study is based on the rationale of the growing demand of intelligent healthcare tools that are accessible, cost-efficient and easy to use. This project will aim at making significant contributions to the scope of dermatological diagnostics to improve the digital health ecosystem using the power of AI and web technology.

2.2 Related Work

Artificial intelligence (AI) and deep learning have achieved considerable progress in the field of deep learning and medical imaging and dermatology. Scholars have conducted different studies in the area of automatic skin disease diagnosis with different imaging modes, architecture of the models, and datasets. The chapter is a complete review of available literature concerning facial skin disease detection, application of deep learning in dermatology, image segmentation, classification models and recommendation systems. The purpose is to pinpoint the gaps in the contemporary research environment and put FaceCare AI into a new niche.[15]

2.2.1 Deep Learning in Dermatology

The application of deep learning in dermatology has been rapidly increasing because it can detect complicated patterns in the high-dimensional image data. CNNs have been extensively applied due to the high performance in image classification functions.

- Esteva et al. (2017) created a CNN that was trained on more than 120,000 clinical images to classify skin cancer, and its performance is comparable to that of dermatologists. This groundbreaking research indicated the possibilities of deep learning in the medical image analysis.
- Han et al. (2018) developed a deep neural network to classify skin diseases in multi-classes and evaluated them on more than 12,000 images. They claimed encouraging outcomes but also created issues to do with diversity of datasets and applicability to the real world.
- Other researchers have used pre-trained models including ResNet, DenseNet and VGGNet to classify skin lesions with high accuracy although with model size and inference speed constraints limiting their usage in resource limited systems such as in a mobile device.[14]

2.2.2 Lightweight Models for Mobile and Web-Based Deployment

In an attempt to address computational constraints of dense models, scholars have resorted to lightweight models such as MobileNet, ShuffleNet, and EfficientNet to be used in real-time and embedded tasks.

- MobileNetV2, created by Google, is a model based on inverted residuals and depthwise separable convolutions that are able to be more accurate and require fewer parameters. It finds extensive application in mobile health applications.
- EfficientNet, and in particular EfficientNetB3, is optimized with respect to network width, depth, and resolution through neural architecture search. Its accuracy and efficiency balance renders it to be appropriate in web applications.
- Lightweight models have been applied by Tschandl et al. (2020) and others in the context of dermatology with reasonable performance in terms of classification, but localization (segmentation) is a problem, as the model is too shallow.[12]

2.2.3 Facial Skin Disease Detection Systems

The majority of the extant studies in the field of skin disease detection are centered on dermoscopic images of the skin lesions. Nonetheless, the facial skin disease detection with regular (RGB) images is under-researched.

- A CNN-based approach to diagnosing acne severity based on cropped facial images was introduced by Kawahara et al. (2016), which preconditioned the development of facial-specific diagnosis.
- Some commercial solutions (e.g., YouCam, TroveSkin) do provide real-time analysis but are based on proprietary models, and have no transparency over model performance or model biases.
- Academic literature tends to focus on individual diseases (e.g., acne or rosacea), multi-label, multi-disease systems have been uncommon in the face.

2.2.4 Recommendation Systems for Skincare

The collaborative filtering, content-based filtering, or hybrid methods are used to provide a recommendation system. The problem of the skincare industry is to recommend the product depending on the skin state and preferences of the user.

- The most recent literature has addressed the way medical imaging results can be combined with knowledge-based systems to prescribe treatment or skincare items. Most of them are however only on prototyping.
- Some are ingredient-matching algorithms, in which identified skin problems are cross-referenced with recommended ingredients, such as salicylic acid, niacinamide, or vitamin C according to dermatological principles.[1]
- Despite these advancements, there is limited integration of real-time detection systems with intelligent recommendation engines within a single platform.

2.2.5 Datasets and Their Limitations

The quality and variety of training data is critical to the success of deep learning models.

- One of the relatively limited publicly available data sets is the Acne04 made up of acne grading of faces. It does not provide annotations on other skin conditions, and is also skewed towards light skin.
- The Skin-Condition dataset that is used in this project consists of five classes of diseases which are acne, pigmentation, vitiligo, rosacea, and wrinkles. Although useful, it nevertheless does not have multi-label annotations or pixel-level masks and thus classification and localization is difficult.
- The lack of a variety of annotated and diverse datasets of faces with different categories of diseases is one of the greatest constraints of the field.

2.2.6 Gap Analysis

Based on the analysis of literature on the topic, the following gaps are considered to be essential:

- Lack of integrated platforms with classification, localization and recommendation functionality.
- Less work on multi-class multi-value facial skin disease detection of real-world RGB images.
- Lack of data of dark-skinned faces, multi-label annotations and ground truth masks.
- Limited consideration of combined chatbot applications to post-diagnosis skincare.
- All these shortcomings indicate why a unified, web-based diagnostic tool such as FaceCare AI, where disease detection and localization are combined with smart product recommendations, is necessary.

2.3 Dataset

During the development of FaceCare AI, the choice of suitable datasets was important in terms of creating a facial skin disease detection and recommendation system. The chapter addresses the datasets that have been utilized, their structure, applicability to research problem, and makes a comparative analysis with the aim of selecting the most appropriate dataset to train models and evaluate them.

2.3.1 Importance of Dataset Selection

The discovery of high-quality datasets is the key to the creation of accurate and generalizable deep learning models. In facial dermatology, the datasets should include a wide range of conditions, skin tones and light conditions to be practical in the real world. Two datasets were used in this project:

- Acne04 Dataset – centered on the severity level of acne.
- Skin-Condition Dataset – a multi-class data-set with different skin diseases of the face.

These datasets had labeled images of faces which were vital in training classification models and determining their performance in various skin conditions.

2.3.2 Acne04 Dataset

The Acne04 dataset is a publicly available dataset comprising of facial images that were labelled in terms of the severity of acne. It has been extensively applied in the grading of acne and it can grading of acne in binary or multi-class classification.

Table 2.1: Acne04 Dataset Overview

Attribute	Details
Source	Public dermatology dataset
Total Images	1,457
Classes	Acne Grade 0 to Grade 4
Format	JPEG (RGB)
Label Type	Image-level classification
Skin Tone Bias	Mostly light/fair skin tones
Annotations	No bounding boxes or segmentation
Use Case	Acne severity detection

2.3.3 Skin-Condition Dataset

The Skin-Condition dataset was created with the aid of open-source repositories of dermatological images and comprises five various diseases of skin on the face. It enables the classification of facial conditions under multi-classes.

Table 2.2: Skin-Condition Dataset Overview

Attribute	**Details**
Source	Compiled from open datasets
Total Images	2,550
Classes	Acne, Rosacea, Vitiligo, Wrinkles, Pigmentation
Format	JPEG/PNG
Label Type	Single-label classification
Annotations	No masks or bounding boxes
Skin Tone Diversity	Moderate (bias toward light skin)

2.3.3.1 Class Distribution

Table 2.3: Class Distribution in Skin-Condition Dataset

Class	**Image Count**
Acne	650
Rosacea	400
Vitiligo	500
Wrinkles	550
Pigmentation	450
Total	**2,550**

2.3.4 Dataset Preprocessing and Split

The two datasets were both downsized to 224×224 and normalized to the desired model input size. Generalization was enhanced with the help of data augmentation techniques (flipping, brightness adjustment, rotation).

Table 2.4: Dataset Split for Training, Validation, and Testing

Subset	**Percentage Used**
Training Set	70%
Validation Set	15%
Test Set	15%

2.3.5 Best Dataset Selection

The Skin-Condition data was chosen as the primary data to be used in this project based on its disease coverage, classes diversity, and quantity. It favors various categories of diseases, which is in line with the goal of creating an inclusive disease detection system.

Table 2.5: Comparison Between Acne04 and Skin-Condition Datasets

Feature	**Acne04**	**Skin-Condition**
Total Images	1,457	2,550
Disease Classes	5 (Acne Grades)	5 (Different Conditions)
Multi-Disease Detection	No	Yes
Annotation Type	Classification only	Classification only
Skin Tone Diversity	Low	Moderate
Localization Support	No	No
Usefulness for Project	Partial	Full

2.3.5.1 Justification

The Skin-Condition data is more compatible with the objectives of the FaceCare AI since it provides:

- Broader disease coverage
- More even class distribution
- Increased number of samples per group
- Increased diversity of facial appearances and patterns of diseases

Therefore, it was considered to be the most efficient data to train a model and evaluate it in this project.

2.4 Models

This paper presents three deep learning models, trained and tested as facial skin disease detectors and classifiers (MobileNetV2, EfficientNetB3 and a custom Convolutional Neural Network

(CNN)). These models have been chosen because they have demonstrated effectiveness in image classification and their scalability as well as their applicability to the lightweight or web-based usage. All the models were trained with the same set and their results analyzed in terms of accuracy, speed and generalization.

2.4.1 MobileNetV2

MobileNetV2 is a convolutional neural network that is lightweight and high-performance, which is designed to run in resource-restricted settings (e.g. mobile and web-based). It employs both inverted residuals and depthwise separable convolution to heavily decrease the computations without loss of competitiveness. It contains a comparably limited number of parameters (3.4 million) and has been trained on ImageNet.

MobileNetV2 was characterized by high performance in the facial disease dataset and fast training time and low memory consumption. It was well-suited to the web-based FaceCare AI system in its architecture.

2.4.2 EfficientNetB3

EfficientNetB3 is a member of the EfficientNet family, which strikes a balance between depth, width and resolution of a model by scaling compounds using a compound scaling mechanism. EfficientNetB3 has been reported to have a high level of generalization and high accuracy yet is more computationally efficient than the standard deep CNNs.

Although it performed well in terms of classification and in training generalization, EfficientNetB3 took more time and consumed more memory than MobileNetV2, so it was not as useful a choice of a lightweight, real-time web application.

2.4.3 Custom CNN

The custom-made CNN architecture has been developed off the shelf using three convolutional layers succeeded by max-pooling and fully connected layers. It was used as a baseline with which to experiment on the dataset. Though easy and simple to train, its accuracy and capability were significantly less than the already trained architectures.

This model did not support transfer learning and it was overfitted because of the small size and diversity of the dataset.

2.4.4 Best Model Selection: MobileNetV2

Having compared all three models, MobileNetV2 was chosen as the most effective one to the FaceCare AI system. It was shown to have a good balance between accuracy, speed and efficiency in deploying. It was the most appropriate to identify and recommend real-time face disease because of its lightweight architecture, and its compatibility with mobile and web platforms. What is more, it was very accurate in identifying all of five target skin conditions (acne, rosacea, vitiligo, pigmentation, wrinkles) which guarantees good performance in real-life applications.

2.5 Summary

This chapter reviewed existing literature, datasets, and deep learning methods for facial skin disease detection. While significant progress has been made using CNN-based approaches, notable gaps remain, including limited multi-class facial disease detection, lack of integrated classification and recommendation systems, and insufficient dataset diversity across skin tones.

After evaluating available datasets, the Skin-Condition dataset was selected due to its coverage of five facial skin conditions and suitability for multi-class classification. Model comparison results showed that MobileNetV2 provides the best balance of accuracy, efficiency, and deployment feasibility, making it well-suited for web-based applications.

Overall, these findings establish a strong foundation for developing FaceCare AI as an efficient and accessible system for automated facial skin disease detection and personalized skincare recommendations.

Chapter 3

Requirement Specification

The successful design and implementation of any software system requires a clear and well structured set of requirements. Facial skin disease detection and individualized skin care recommendation through the FaceCare AI system will be based on clear specifications to achieve the desired goals. This chapter defines both the functional and non-functional requirements that will determine the behavior, performance, usability and reliability of the system. Besides the essential system capabilities, which include detection of the disease, image processing, and product recommendation, the technical environment, hardware requirements, computer tools and user interface requirements are also reported. These requirements make sure that the entire system including the deep learning model integration and frontend usability are in line with the user expectations and project objectives. The documented requirements form a basis of the system design as well as future validation procedures.[4]

3.1 System Overview

FaceCare AI is a web-based application that aims to identify facial skin diseases and give skincare suggestions to a person based on deep learning models. The system combines modules of image classification, disease localization and product suggestion, providing a convenient system of early skin health monitoring. It uses light and precise models such as MobileNetV2 to provide end-users with fast and real-time response.[5]

The system architecture comprises of the React-based image upload and user interaction frontend, FastAPI based Python back-end to read model inference and logic processing and MySQL database to manage users and products. This is aimed at delivering an automated, precise, and scalable solution in regard to an improved understanding of health as well as improvement of skin.

Key Components:

- **Image Upload Interface**
  Users will be able to post or take faces with the help of a web camera or cell phone.
- **Preprocessing Module**
  Images are downsized, standardized, and improved and entered to the model.
- **Disease Detection Engine**
  The trained MobileNetV2 model recognizes and classifies the skin type by one of five categories, namely, Wrinkles, Acne, Rosacea, Vitiligo, Pigmentation.
- **Recommendation System**
  The system suggests skincare products and the corresponding ingredients depending on the identified condition.
- **Chatbot Integration**
  An easy-to-use AI chatbot will give information about the skincare regimen and some general recommendations.
- **User Management**
  The system enables registration of a user, logging in, and having the history of diagnostic logs that one can make reference to in the future.
- **Database Layer**
  The user profiles, prediction logs and product information are stored in MySQL.
- **Web-Based Interface**
  The interface is easy to use and has been built with React, making it responsive and also optimized to different screen sizes.
- **Security Measures**
  There is basic authentication and secure data of users and uploaded photos to ensure privacy.

As indicated by this overview, the system is designed in such a way that it provides a comprehensive approach to automated skin analysis but at the same time to be usable, perform, and accessible in various user environments.

3.2 Functional Requirements

Functional requirements are those that specify the exact operation and tasks that the FaceCare AI system should execute in order to achieve its intended purpose. These requirements are used to define the reaction of the system to user interaction and the internal processes of the system towards data processing to provide outcomes. The FaceCare AI application should enable the user to create an account and log-in safely, upload or scan images of their face, and conduct the inputs into the deep learning model to detect possible skin diseases. Once it has been classified, the system should provide the diagnosis and the description of the identified condition. According to this output, the system is supposed to give the appropriate skincare product recommendations, whereby it contains ingredients based on the skin concern of the user. Also, the users must have the opportunity to communicate with a chatbot to get general advice on skincare, look at their previous outcomes, and control their profile. All modules must be well coordinated and they must work in harmony with database and the back end environment to provide real time response, data persistence as well as user friendly interface. The system must also manage errors like bad image formats, lack of inputs or unsuccessful model prediction by giving meaningful error messages. The functional requirements guarantee the end-to-end usability, reliability, and performance of the system among the diverse features.

3.3 Non-Functional Requirements

The non-functional requirements specify the qualitative aspects and the general attributes, which the FaceCare AI system should possess in order to guarantee high performance, security, usability, and reliability. Non-functional requirements unlike functional requirements articulate how the system performs under different conditions and the efficiency of its service to the user. These are performance limitations like response time, scaling with growing user load, ease of use, protection of sensitive user information, system maintainability and compatibility with a variety of platforms. By covering such features, the system guarantees a smooth and productive user experience, especially where facial health diagnostics are involved, which is a highly sensitive area. The achievement and integration of the FaceCare AI platform will mostly rely on the satisfaction of these non-functional requirements since they have a direct effect on the user satisfaction, system reliability and system longevity.

Table 3.1: Functional Requirements of the FaceCare AI System

ID	Requirement	Description
FR1	User Registration and Login	Allows users to securely register, log in, and manage accounts.
FR2	Image Upload and Capture	Users can upload or capture facial images for analysis.
FR3	Image Preprocessing	System preprocesses images (resize, normalize) before sending to the model.
FR4	Disease Detection	Deep learning model classifies skin conditions into one of five disease types.
FR5	Disease Result Display	Displays identified disease with description and confidence score.
FR6	Product Recommendation	Suggests skincare products and ingredients based on the diagnosed condition.
FR7	Chatbot for Skincare Advice	Provides users with general skincare tips via an integrated AI chatbot.
FR8	User History Management	Saves prediction history and allows users to revisit previous diagnoses.
FR9	Admin/Product Management	Admin panel to manage product database and update recommendations.
FR10	Error Handling and Validation	Handles invalid inputs, image types, and system errors gracefully.

3.4 Hardware Requirement

The hardware infrastructure in which FaceCare AI system is developed and deployed is important to the performance and accuracy of the system. Proper hardware will guarantee the seamless processing of images, quick models inference, and swift user experience. The operations of deep learning-based facial skin disease detection are computationally intensive, which means that they involve image classification and feature extraction, and thus the development environment should support the use of a graphic card and adequate memory.

3.5 Software Requirement

The software requirements specify the development environment, programming languages, frameworks, and other support tools that will help to successfully implement the FaceCare AI system. The backend uses Python and FastAPI to develop the API and uses deep learning (TensorFlow

Table 3.2: Non-Functional Requirements of the FaceCare AI System

ID	Requirement	Description
NFR1	Performance	The system must process and return results within 3–5 seconds per image.
NFR2	Scalability	The architecture should support scaling to handle increasing user loads in the future.
NFR3	Usability	The interface must be simple, intuitive, and accessible across various devices and browsers.
NFR4	Reliability	The system should function consistently without crashes during prediction or recommendation.
NFR5	Maintainability	The codebase and architecture must be modular and easy to update or debug.
NFR6	Security	User data and images must be handled securely with validation and restricted access.
NFR7	Compatibility	The web app should be compatible with modern browsers like Chrome, Firefox, and Edge.
NFR8	Responsiveness	The frontend must dynamically adapt to different screen sizes including mobile and desktop.
NFR9	Availability	The system should aim for a high availability rate (e.g., 99% uptime) during operational hours.
NFR10	Documentation	Technical and user documentation should be provided for maintenance and usability.

Table 3.3: Hardware Requirements for FaceCare AI System

Component	Minimum Specification	Recommended Specification
Processor	Intel Core i5 (6th Gen) or equivalent	Intel Core i7 / AMD Ryzen 7 or better
RAM	8 GB	16 GB or more
Storage	256 GB HDD or SSD	512 GB SSD or higher
Graphics Card	Integrated GPU or NVIDIA GTX 1050	NVIDIA RTX 2060 / 3060 with CUDA support
Internet Connection	Standard broadband	High-speed broadband (50 Mbps or more)
Display	720p resolution	Full HD (1080p) or higher

and OpenCV) to classify images and detect diseases. ReactJS is used to create the front end to give the site an interactive interface. MySQL is relational database which is used to store the user data and history. Moreover, multiple libraries and modules are also used to preprocess the images, infer the model, and integrate chatbots. The table below gives a summary of the software stack and dependencies in the system.

Table 3.4: Software Requirements for FaceCare AI System

Software / Tool	**Description / Use**
Python 3.9+	Backend development and model integration
FastAPI	RESTful API framework for backend services
TensorFlow / Keras	Deep learning libraries for training and inference
OpenCV	Image processing (resizing, filtering, enhancement)
ReactJS	Frontend framework for web interface
MySQL	Database for storing user data, history, and recommendations
Visual Studio Code / PyCharm	Code editor/IDE for development
Postman	API testing and debugging tool
Docker (Optional)	Containerization for deployment
Git	Version control and project collaboration

3.5.1 Minimum Client-Side Requirements

Table 3.5: Minimum Client-Side Requirements

Component	**Minimum Requirement**
Device	Smartphone / Laptop
Camera	720p resolution or higher
Internet	Minimum 2 Mbps connection
Browser Support	Chrome, Firefox, Edge

3.5.2 Backend Software Stack and Tools

Table 3.6: Software Stack and Tools (Backend)

Component	**Requirement**
Programming Language	Python 3.9+
Web Framework	Django
Database	SQLite3
Deep Learning Libraries	TensorFlow, Keras, Ultralytics
Computer Vision Libraries	OpenCV, Pillow
Chatbot Integration	Local LLM (via Ollama)

3.5.3 Frontend Stack and Technologies

Table 3.7: Frontend Stack and Technologies

Component	Requirement
Framework	HTML/CSS + JavaScript
Styling	Bootstrap
Animations	GSAP (GreenSock Animation Platform)
API Communication	Axios / Fetch API

3.6 User Interface Requirements

The user interface (UI) of FaceCare AI system is important in ensuring accessibility, usability and smooth-sailing experience to the end users. The system should be user friendly as its interface is supposed to be easy to navigate due to the supposed limited technical expertise among the users. The design is centered on visual clarity and ease of navigation and quick response to user actions like uploading images, detecting a disease, and product suggestions. The frontend is constructed on the basis of HTML, CSS, JavaScript, and Bootstrap to structure and style, and GSAP to create modern animation and interactive transitions.

Due to the extensive variety of screen sizes and devices to which the interface should be compatible, i.e., desktops, laptops, and smartphones, responsive design turns out to be a crucial requirement. They ought to allow users to execute the most important tasks without getting lost such as logging in, posting/taking pictures, disease predictions, and seeing individual recommendations of skincare products. Moreover, the inclusion of a chatbot can help with the conversational support of skincare-related questions enhancing the user interaction and support.

Table 3.8: User Interface Features and Descriptions

Feature	Description
Responsive Design	Adapts layout for mobile phones, tablets, and desktop screens
Image Upload Interface	Allows users to upload or capture facial images for diagnosis
Real-time Feedback	Provides instant disease prediction and recommended products
Navigation Bar	Enables users to move between Home, About, Predict, and Contact pages easily
Chatbot Integration	AI-based chatbot for skincare guidance and answering user queries
Progress Indicators	Displays loading bars and diagnosis status while model processes the image
Error Handling	User-friendly error messages for invalid uploads or system failures
Visual Appeal	Smooth animations and transitions using GSAP for enhanced user engagement

3.7 Fully Use Case Diagram

The user interface (UI) of FaceCare AI system is important in ensuring accessibility, usability and smooth-sailing experience to the end users. The system should be user friendly as its interface is supposed to be easy to navigate due to the supposed limited technical expertise among the users. The design is centered on visual clarity and ease of navigation and quick response to user actions like uploading images, detecting a disease, and product suggestions. The frontend is constructed on the basis of HTML, CSS, JavaScript, and Bootstrap to structure and style, and GSAP to create modern animation and interactive transitions.

Due to the extensive variety of screen sizes and devices to which the interface should be compatible, i.e., desktops, laptops, and smartphones, responsive design turns out to be a crucial requirement. They ought to allow users to execute the most important tasks without getting lost such as logging in, posting/taking pictures, disease predictions, and seeing individual recommendations of skincare products. Moreover, the inclusion of a chatbot can help with the conversational support of skincare-related questions enhancing the user interaction and support.

3.7.1 Use Case 01: Register

Table 3.9: Use Case 01: Register

Use Case Element	**Description**
Use Case ID	UC01
Use Case Name	Register
Actor	User
Description	This use case allows a new user to create an account by providing basic information such as name, email, and password. The system stores the credentials and confirms registration.
Preconditions	The user has not registered before and has a valid email address.
Postconditions	A new user account is created and saved in the system. The user is redirected to the login page.
Main Flow	User clicks on the "Register" button.User enters name, email, and password.System validates the input.System saves user data to the database.Confirmation message is shown.
Alternate Flow	If the email is already registered, an error message is displayed.If any field is empty or invalid, form validation messages are triggered.
Exceptions	Network failure during registration.Database error while saving data.

3.7.2 Use Case 02: Login

Table 3.10: Use Case 02: Login

Use Case Element	**Description**
Use Case ID	UC02
Use Case Name	Login
Actor	User
Description	This use case enables a registered user to log into the system using valid credentials. The system authenticates the user and grants access to the main dashboard.
Preconditions	The user must already be registered and have valid login credentials.
Postconditions	The user is successfully logged in and redirected to the home/dashboard page.
Main Flow	User clicks on the "Login" button. User enters email and password. System checks credentials. If valid, access is granted. User is redirected to the dashboard.
Alternate Flow	If credentials are incorrect, an error message is displayed and the user is prompted to try again.
Exceptions	System may fail to log in the user if the server is down or the database is unreachable.

3.7.3 Use Case 03: Image Upload/Capture Image

Table 3.11: Use Case 03: Image Upload / Capture Image

Use Case Element	**Description**
Use Case ID	UC03
Use Case Name	Image Upload / Capture Image
Actor	User
Description	This use case allows the user to upload an existing facial image or capture a new one using their device camera. The image is used for skin disease detection.
Preconditions	The user must be logged in and have a working device with a camera or access to image files.
Postconditions	The image is successfully uploaded or captured and stored temporarily for processing.
Main Flow	User navigates to the "Predict" page. User selects an option to upload or capture an image. System receives the image input. Image is previewed on the interface. User confirms the image to proceed.
Alternate Flow	If the user cancels or uploads an invalid file type, the system prompts to re-upload or recapture the image.
Exceptions	Camera permission denied, file format not supported, or image not readable due to corruption.

3.7.4 Use Case-04: Disease Detection

Table 3.12: Use Case 04: Disease Detection

Use Case Element	**Description**
Use Case ID	UC04
Use Case Name	Disease Detection
Actor	System
Description	After receiving the uploaded or captured facial image, the system processes it through the selected deep learning model to detect the presence and type of facial skin disease.
Preconditions	A valid image must be uploaded or captured by a logged-in user. The detection model must be trained and loaded properly.
Postconditions	The disease is identified and the result is displayed to the user along with its label and confidence score.
Main Flow	System receives the input image. Preprocessing is applied to match model input requirements. The deep learning model performs disease classification. Detected class and confidence level are returned. Results are displayed on the user interface.
Alternate Flow	If the image is unclear or does not contain a detectable face, the system informs the user and prompts re-upload.
Exceptions	Model fails to load, image fails preprocessing, or detection process crashes due to insufficient system resources.

3.7.5 Use Case 05: View Scans

Table 3.13: Use Case 05: View Scans

Use Case Element	**Description**
Use Case ID	UC05
Use Case Name	View Scans
Actor	User
Description	This use case allows the user to view a history of their previously scanned images along with disease detection results and timestamps.
Preconditions	The user must be logged in and must have completed at least one disease detection scan previously.
Postconditions	The system displays a list or gallery of all past scans with corresponding disease predictions and dates.
Main Flow	User navigates to the "My Scans" or "History" section. System fetches user's past scan records from the database. Scans are listed chronologically with disease results. User can select a scan to view detailed results.
Alternate Flow	If no scan history is found, the system displays a message: "No past scans available."
Exceptions	Database retrieval fails or user session times out while accessing scan history.

3.7.6 Use Case-06: AI Chatbot Interaction And E-Commerce Interaction

Table 3.14: Use Case 06: AI Chatbot Interaction and E-Commerce Recommendation

Use Case Element	**Description**
Use Case ID	UC07
Use Case Name	AI Chatbot Interaction and E-Commerce Recommendation
Actor	User, AI Chatbot
Description	The user interacts with the AI chatbot to receive answers to skin-related queries and product guidance. The chatbot further recommends available skincare products that can be purchased directly from the platform.
Preconditions	The user must be logged in and should have completed a scan or initiated a conversation with the chatbot. Product recommendation logic and chatbot knowledge base must be active.
Postconditions	The chatbot responds with advice, clarifies user questions, and provides product suggestions with links for purchase.
Main Flow	User opens the chatbot interface. User types a skin-related question or seeks product help. The chatbot uses local LLM to analyze the query. Based on disease or user need, relevant advice and product links are shared. User may follow the links to view or purchase items.
Alternate Flow	If the chatbot cannot understand the query, it asks the user to rephrase or offers general help options.
Exceptions	LLM fails to respond, product database is unavailable, or e-commerce integration is temporarily down.

3.8 System Use Case Diagram

The face care AI: Facial Skin Disease Detection and Localization using Deep Learning system visual representation is the use case diagram that includes the visual representation of the main features offered to the user and the system itself. It underlines the interaction of the users with the platform during the registration and logging in to uploading or capturing facial pictures to be diagnosed. After an image is uploaded, it is processed by the system through deep learning models including MobileNetV2 and EfficientNetB3, which are used to identify such conditions as acnes, rosacea, vitiligo, pigmentation, and wrinkles. Identified results lead to use cases of seeing scan history, getting personalized skincare advice, and communicating with an AI-based Chatbot to get additional advice. The examples of the use cases by the administrators are dataset management, system monitoring, and user moderation. Not only does this diagram define the functional scope of the system but it also guarantees that the interaction flow is clear to assist the developers, testers and stakeholders in comprehending the system behavior and design.

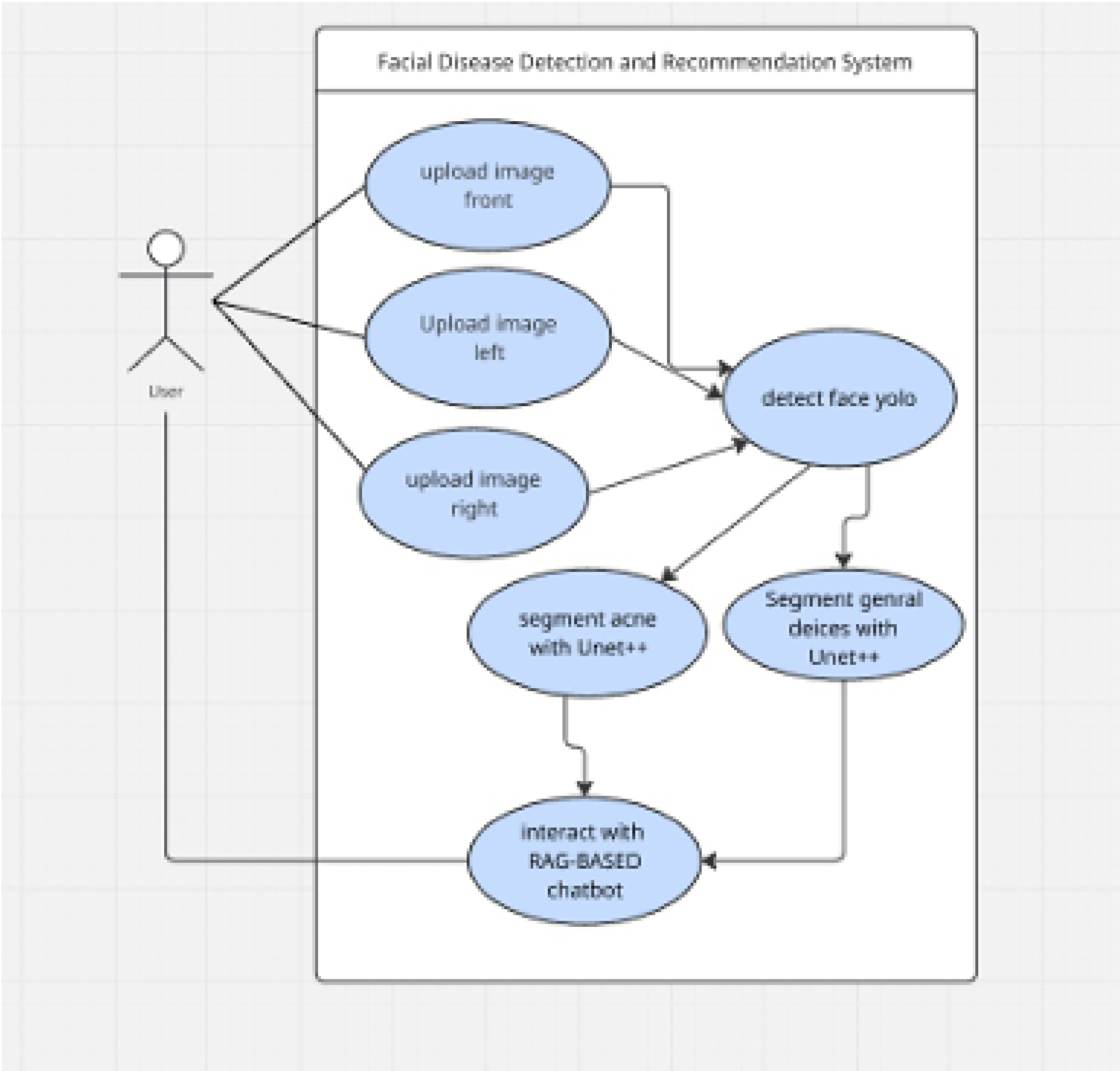

Figure 3.1: Use Case Diagram

3.9 System Flow Diagram

The system flow diagram is the logical flow of the operations in the application. It provides the major processes within it, starting with the interaction of the user with the interface, through detection of the disease and delivery of recommendations. The diagram is a stepwise flow of the data flow across various elements of the system such as the frontend, backend, machine learning model and the database.

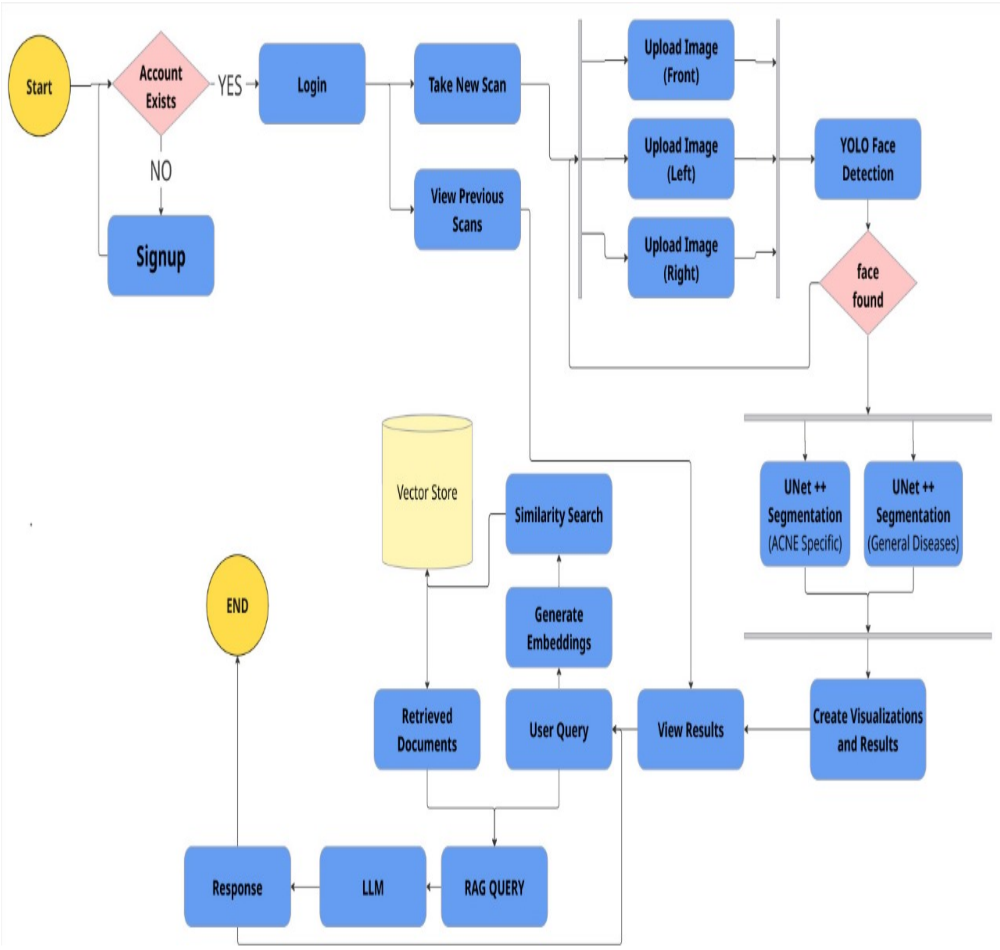

Figure 3.2: System Flow Diagram

3.10 Security Requirements

The requirements are follow:

- All the user data should be stored and transmitted in an encrypted form.
- Prediction services require users to be authenticated.
- The endpoints of the backend are to be secured via token authentication.

3.11 Summary

In this chapter, the overall requirement specifications that will be used to develop and deploy the Face Disease Detection and Recommendation System successfully were introduced. It discussed both non-functional and functional requirements, hardware and software requirements, feasibility study as well as detailed use cases. The following specifications will be the roadmap to the construction of a safe, effective, and user-friendly system that correctly identifies facial skin diseases and can give specific skincare suggestions to the user.

Chapter 4

System Design

System design is one of the most important processes in the software development lifecycle as it fills the gap between the theoretical requirements and the practical implementation. The architectural design, component design, data flow, model integration, and user interaction strategy of our project entitled Face Disease Detection and Recommendation System are presented in this chapter. The system will be a combination of deep learning-based image segmentation and conversational AI in making skincare recommendations, which will be made available through a user-friendly web application. This chapter is aimed at giving a thorough picture of the way the system is designed and how different parts of the system can work together to produce the functionality and user experience desired.

4.1 System Architecture

The entire system layout is in the modular style where concerns are divided between the frontend (client-side interface), backend (server-side logic), integration of the AI model, and database management.

Table 4.1: System Architecture Layers

Layer	**Description**
Presentation Layer	Developed using React.js; handles user interactions and input collection.
Application Layer	Built in Python using Flask or FastAPI; processes data and integrates ML models.
Model Layer	Executes disease detection using MobileNetV2 and triggers chatbot recommendations.
Database Layer	Stores user data, predictions, and logs via MySQL database.

4.1.1 System Architecture Diagram

The system architecture diagram depicts the stratified design FaceCare AI system, as the interaction among the client layer (ReactJS frontend), application layer (FastAPI backend), model layer (MobileNetV2), and data layer (MySQL database). This diagram provides a global perspective on the flow of data in the system after user interaction till the detection of the disease and the provision of the recommendations.

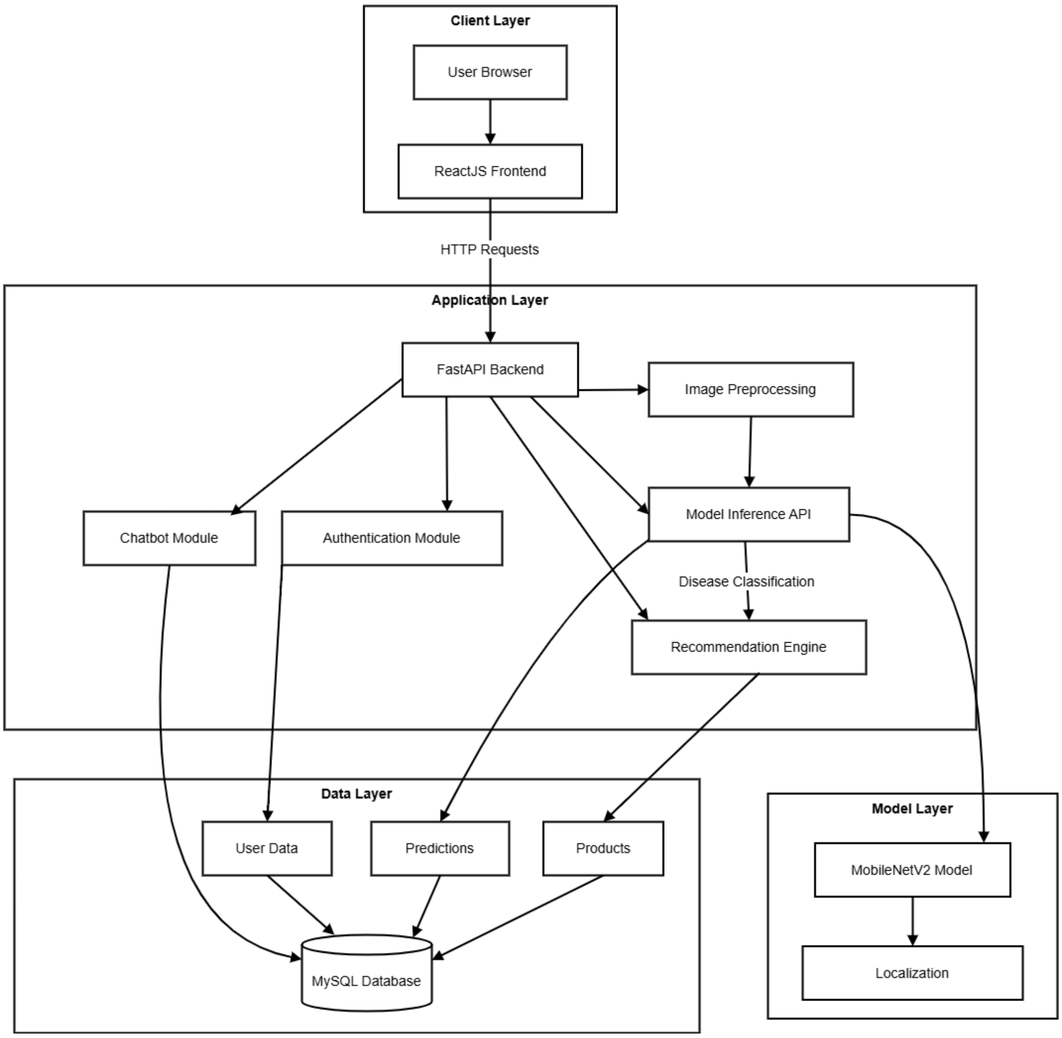

Figure 4.1: System Architecture Diagram

4.1.2 System Components

The system consists of a number of major parts with each part performing a particular task in the disease detection and recommendation chain.

Table 4.2: System Components and Their Functions

Component	Function
Image Input Module	Captures or uploads user images (front, left, right views).
Preprocessing Module	Resizes and formats images for model compatibility.
Segmentation Model (MobileNetV2)	Detects and segments skin diseases from the facial images.
Result Visualization Module	Highlights affected areas with masks or bounding circles.
Chatbot Module	Provides skin-related recommendations and answers queries.
Authentication Module	Manages user login and registration.
Database Handler	Logs user activities and stores image results and predictions.

4.1.3 Component Diagram

The component diagram gives a closer look at the entire system components and their association. It demonstrates the interaction of frontend components with backend services and the latter interact with core services and data services.

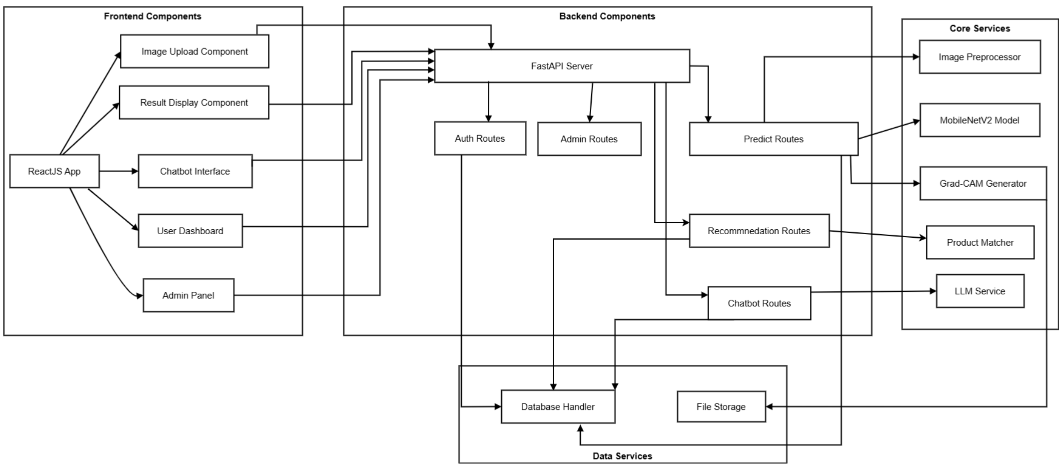

Figure 4.2: Component Diagram

4.1.4 Data Flow Design

The system must also run and be highly responsive which is achieved by ensuring that the data flows between one component and another component of the system. What follows is the data flow that it implements in structured form:

- User logs into the system.
- Facial images are uploaded/captured by the user.
- The images are transmitted to the backend server.
- Backend does preprocessing (resize, normalize).
- These images are fed into the MobileNetV2 network.
- Masks of segmentation are returned.
- Frontend indicates areas of disease detection.
- In case of the absence of a disease, the user is informed about the healthy skin condition.
- Chatbot opens with product suggestions and answers to queries related to the skin.
- Result and logs are stored in the database.

Database Design

The database is formed of several tables to deal with users, prediction logs, and interactions between the chatbot. MySQL is employed in the processing of data safely and effectively.

4.1.4.1 Database Schema

The database schema consists of the following main tables:

Table 4.3: Main Database Tables and Descriptions

Table Name	Description
Users	Stores user profiles, email, password (hashed), personal information, and admin status.
Predictions	Contains uploaded images, model output, disease labels, confidence scores, and heatmap paths.
Products	Stores skincare product information including name, category, ingredients, and associated disease types.
ChatLogs	Records user queries and chatbot responses for future reference.

4.1.4.2 Entity Relationship Diagram

The Entity Relationship (ER) diagram demonstrates the connections between the tables of the database. Users table is one-to-many in relation to Prediction and ChatLogs since a user can have a number of records of the predictions and chat interaction. The Predictions table is connected to the Products by the disease type matching to get individualized recommendations.

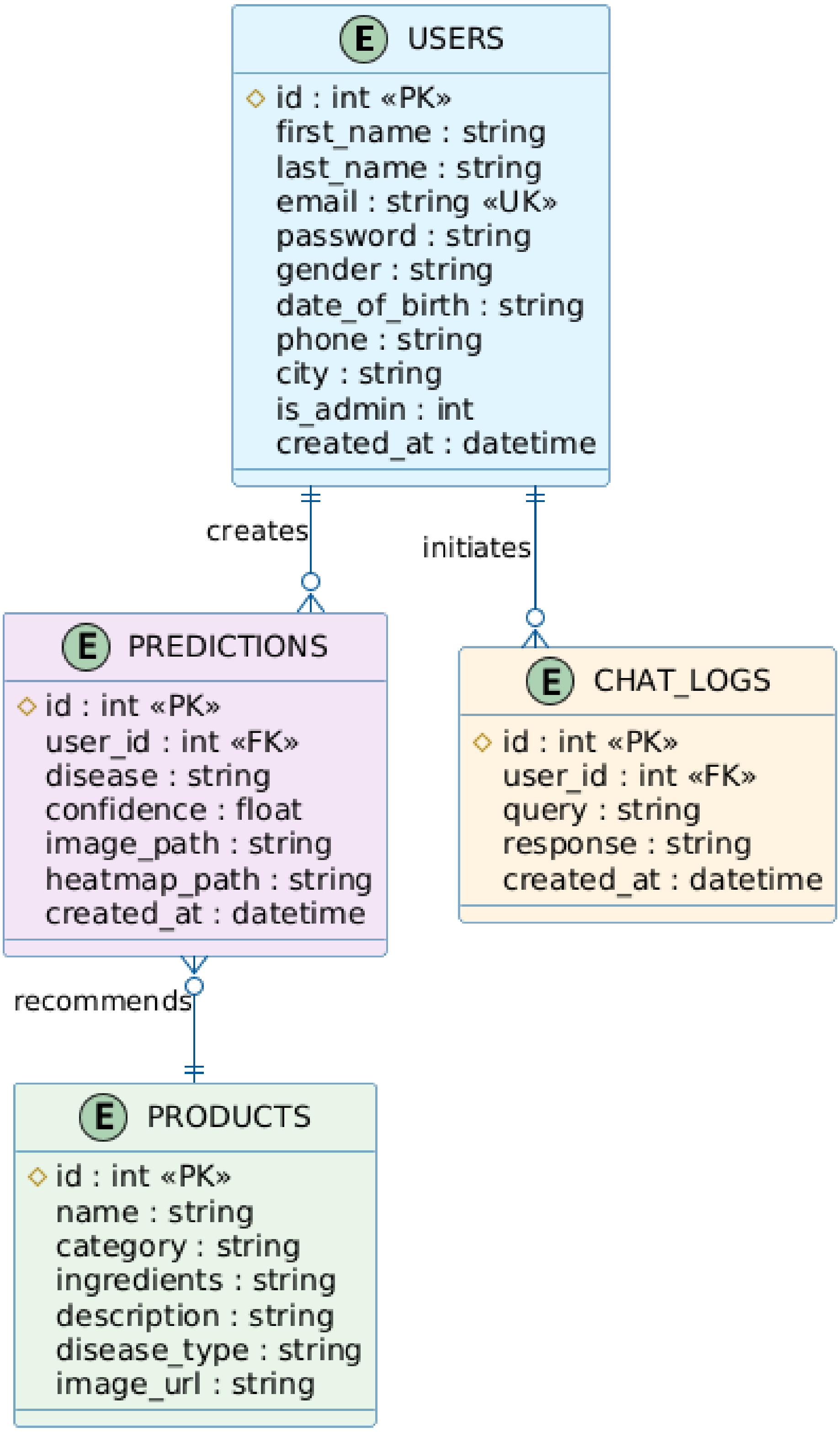

Figure 4.3: Database Schema ER Diagram

4.1.5 User Interface Design

The user interface on the front end has been made user-friendly and responsive. Every page has its purpose of leading the user through the services of the system.

4.1.5.1 Pages Description

Table 4.4: Pages and Their Functionality in the User Interface

Page	**Functionality**
Home Page	Welcomes users and introduces the project with clean UI design.
About Us Page	Provides project details, developer names, and vision.
Contact Page	Allows users to get in touch for queries or feedback.
Prediction Page	Central module for uploading/capturing images and starting analysis.
Results Display	Shows disease type, affected regions, and "healthy skin" message if applicable.
Chatbot Interface	Opens automatically post-detection; allows skincare inquiries and product recommendations.

4.1.6 System Activity Diagram

The system activity diagram shows the full workflow of the face skin disease detection application. After user authentication, the next steps are face detection and uploading multiple angle images of the face. Once a face is successfully identified, the application generates disease segmentation results using a RAG-based LLM.

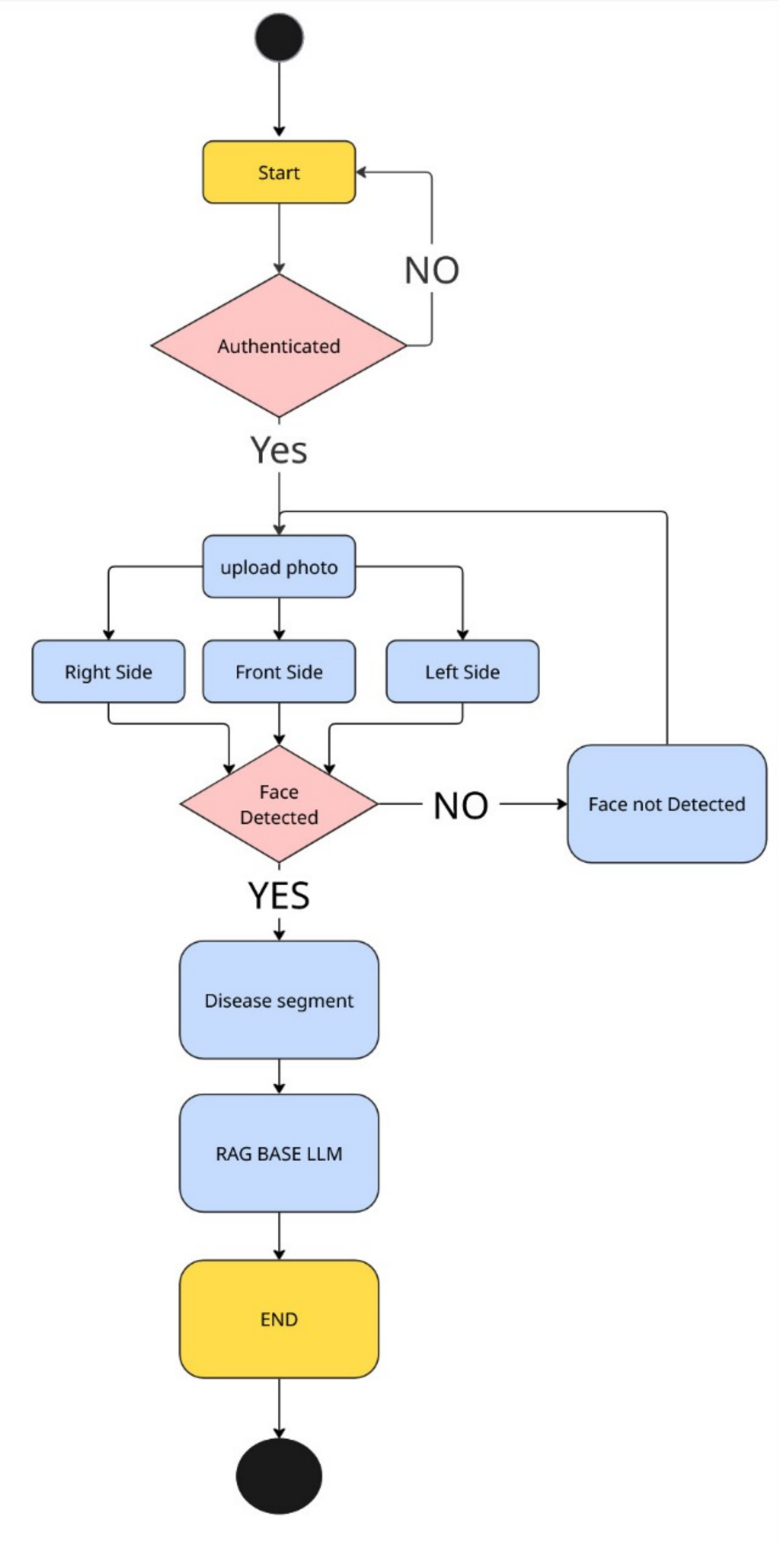

Figure 4.4: System Activity Diagram.

4.2 DFD Level 0 – Context-Level Diagram

Context-Level Diagram is a high level representation of the system and the interactions with the external entities.

- External Entity: **User**
- System: **Face Disease Detection and Recommendation System**

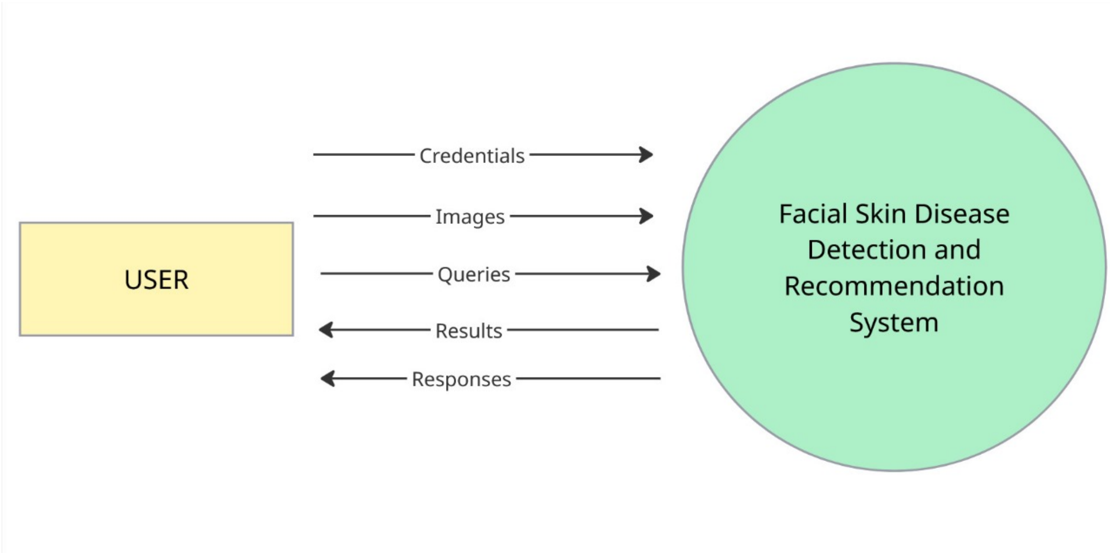

Figure 4.5: DFD Level 0 Diagram.

4.3 DFD Level 1 – Decomposition of Main Process

The DFD Level 1 diagram breaks down the main process into sub-processes, showing how data flows within the system.

- Data Stores:
  - D1: User Data Store
  - D2: Result Data Store

  - D3: Vector Store

- Sub-Processes:
  - Authenticate: Authenticate User Process.
  - Upload and preprocess pictures: Process to upload and preprocess user-pictures.
  - Image Preprocessing: (Scales and normalizes images to match model input requirements) (224x224 pixels).
  - Segment Diseases: It applies the MobileNetV2 models to detect skin diseases.
  - Create Visualization: Visualization is created through overlaying the segmentation masks of the face images.
  - LLM analysis: The analysis of the face is facilitated by the use of the MobileNetV2 models in the LLM analysis.
  - Answer User Query: Responds to queries of users based on the vector Store and LLM.

- Data Flow:
  - Image uploaded by the user is sent to the server: Upload and preprocess images.
  - Face Detection Processed Image.
  - Face detected Disease Segment.
  - Segmentation Masks → Visualization Generation.
  - Segmentation Masks → LLM Analysis.
  - User Questions → Response to user query.

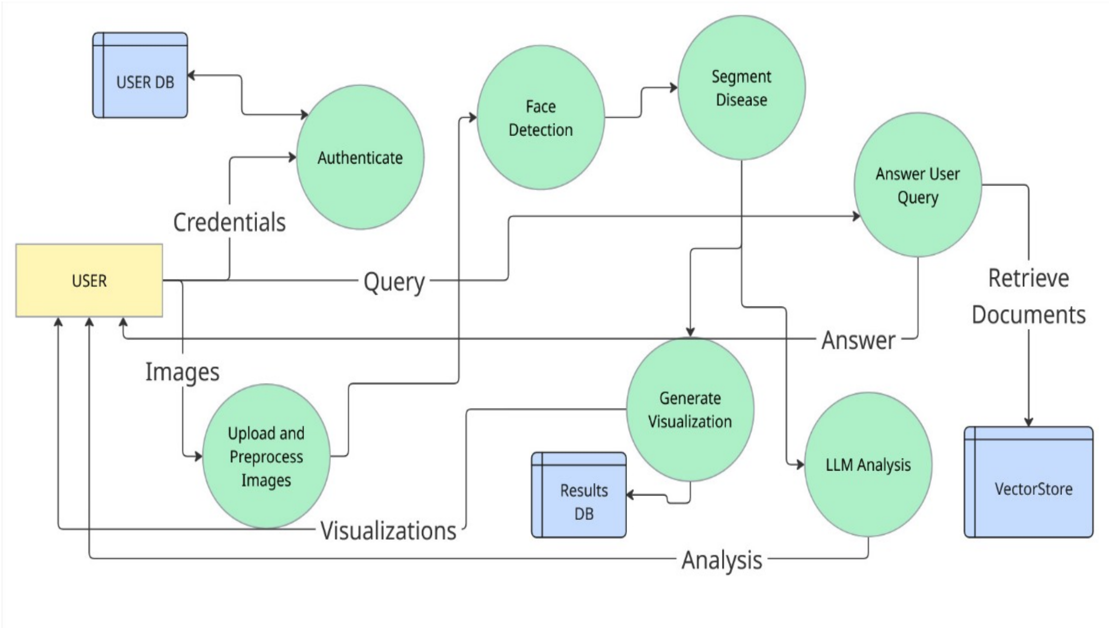

Figure 4.6: DFD Level 1 Diagram.

4.4 State Diagram

State Diagram illustrates the change of state of the system with the action of the user or internal processing. The system goes to the other states in accordance with the subsequent actions:

- **State 1**: User logs in the system.
- **Login State**: The user logs-in successfully.
- **Image Upload State**: This is where a user uploads a facial image.
- **Prediction State**: Image is processed by the system and the skin disease is detected.
- **Result State**: User is shown the detection result and segmentation.
- **Chatbot State**: Chatbot can give answers or skincare recommendations.
- **End State**: User terminates the session or logs off.

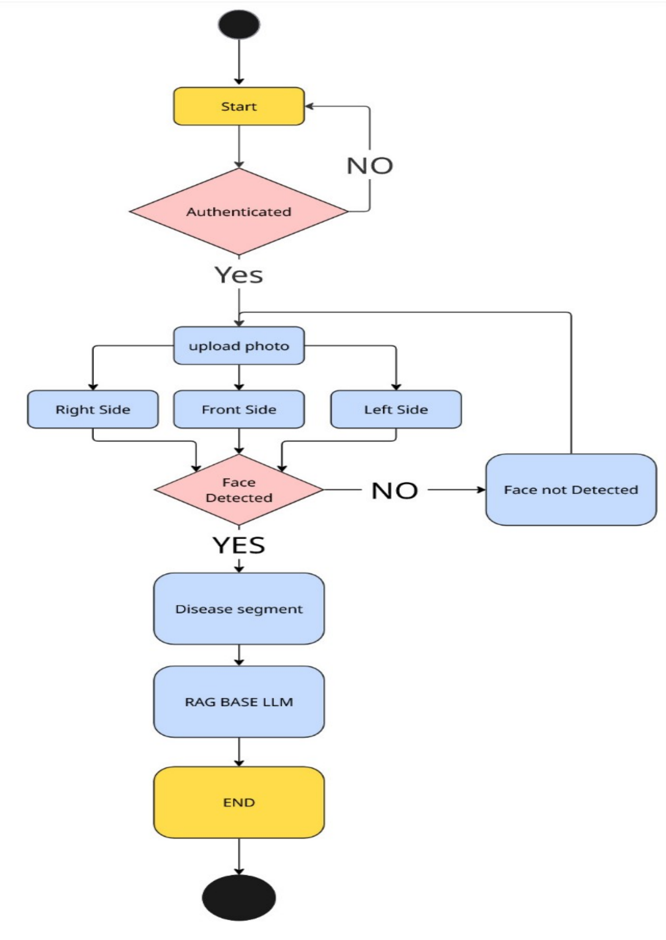

Figure 4.7: State Diagram.

4.5 Sequence Diagram

The sequence diagram is a time-sequenced interaction of the various components of the system. It is concerned with the execution of a given use case between the user, the frontend interface, the backend server, the machine learning model, and the database in terms of the messages traversed. In this project, the sequence diagram shows the process flow of communication through the actions taken upon initiation of a process by a user like uploading a facial image to when the system identifying the skin condition and giving product recommendations. It incorporates the rational sequence of actions and focuses on the time and synchronization of elements.

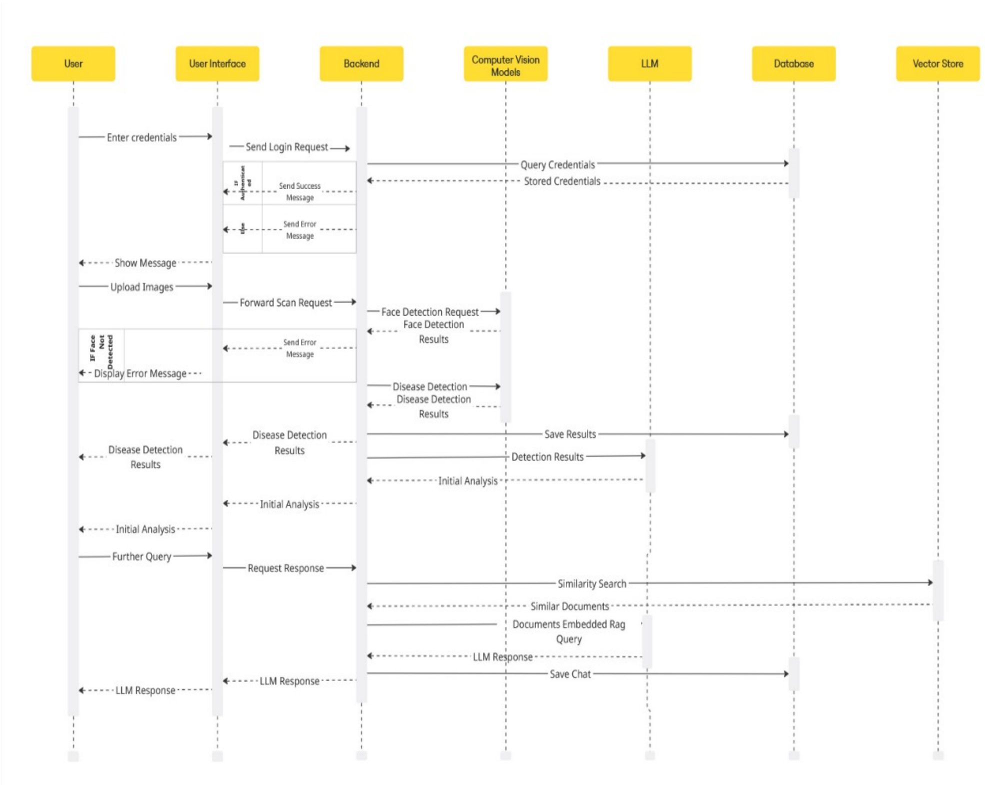

Figure 4.8: Sequence Diagram

4.6 Deployment Architecture

The deployment architecture diagram can be used to show the way the FaceCare AI system is implemented on various levels, including the web browser of the user, the back-end servers, and the database. This figure depicts the communication protocols, ports and service interactions within the production environment.

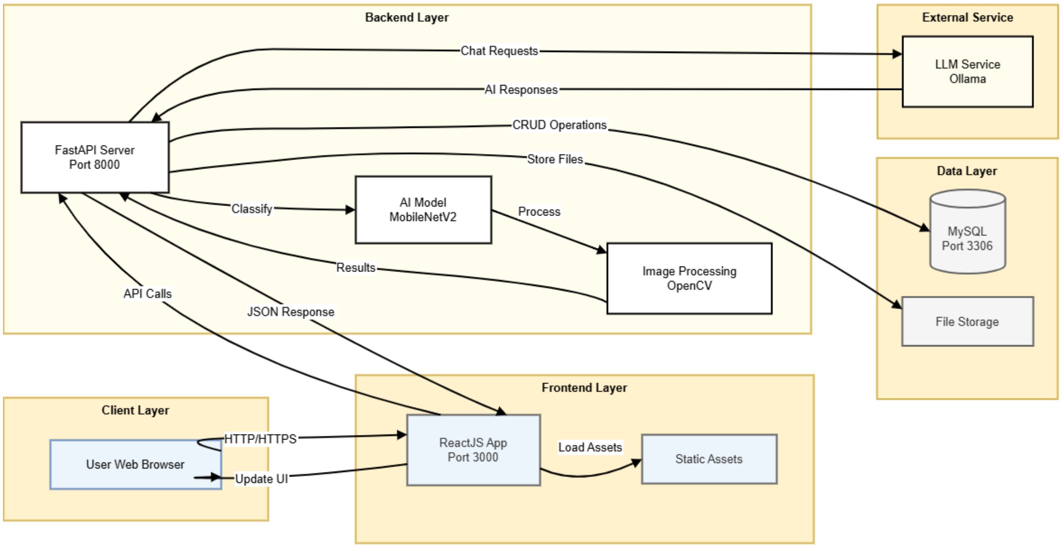

Figure 4.9: Deployment Architecture Diagram

4.7 Summary

The chapter on system design has presented the architecture, data flow, UI design, model integration, and database design of Face Disease Detection and Recommendation System. This scalable and modular design will guarantee the system is optimized, responsive, and able to detect and provide skincare help in real-time and identify diseases. The design was well developed so as to meet the requirements of user satisfaction, technical reliability and future extensibility.

Chapter 5

Methodology

This chapter outlines the research methodology that was used to develop FaceCare AI, a web-based system to detect facial skin disease and recommend skincare. The methodology presents a step-by-step approach to designing, training, assessing, and deploying the deep learning models and also the integration of the backend and frontend components to build a real-time web interaction. The general system workflow, data collection and preprocessing, model development and comparison, backend-frontend integration, and the implementation of the recommendation system are discussed in the chapter. It also outlines the evaluation measures that will be applied to gauge the model performance and the rationale of choosing MobileNetV2 as the final model to deploy.

5.1 Overview of Methodology

The steps of this study have a structured data-driven pipeline, which consists of multiple steps. Figures 5.1 and 5.2 show the full research methodology process of the FaceCare AI project, starting with the requirements analysis process up to the deployment and maintenance.

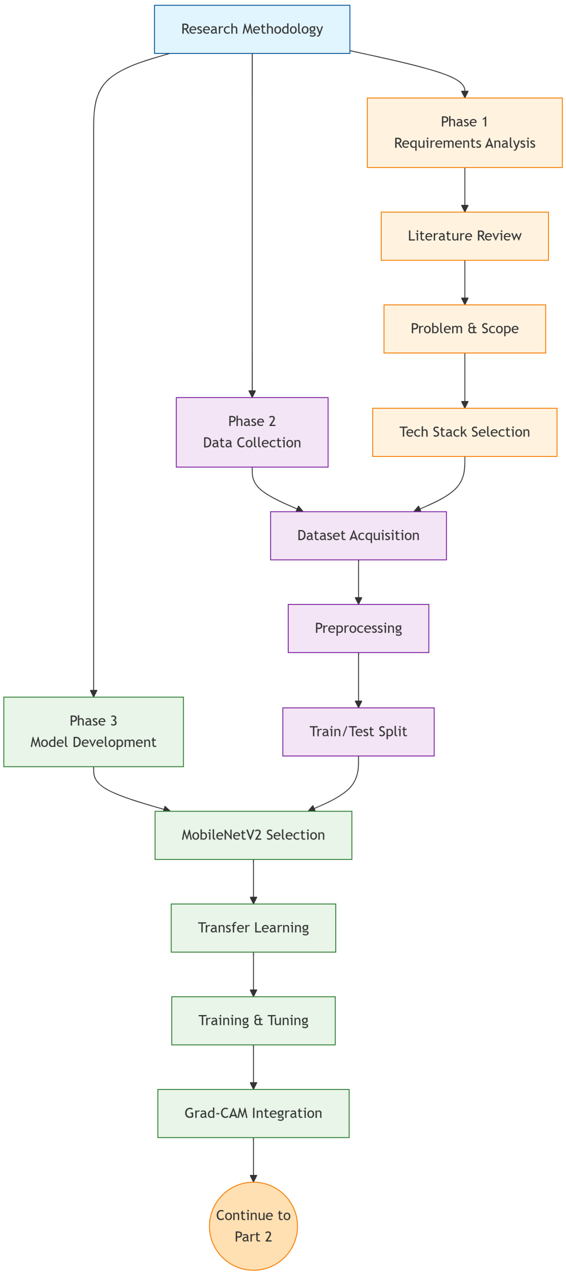

Figure 5.1: Research Methodology Diagram - Part 1

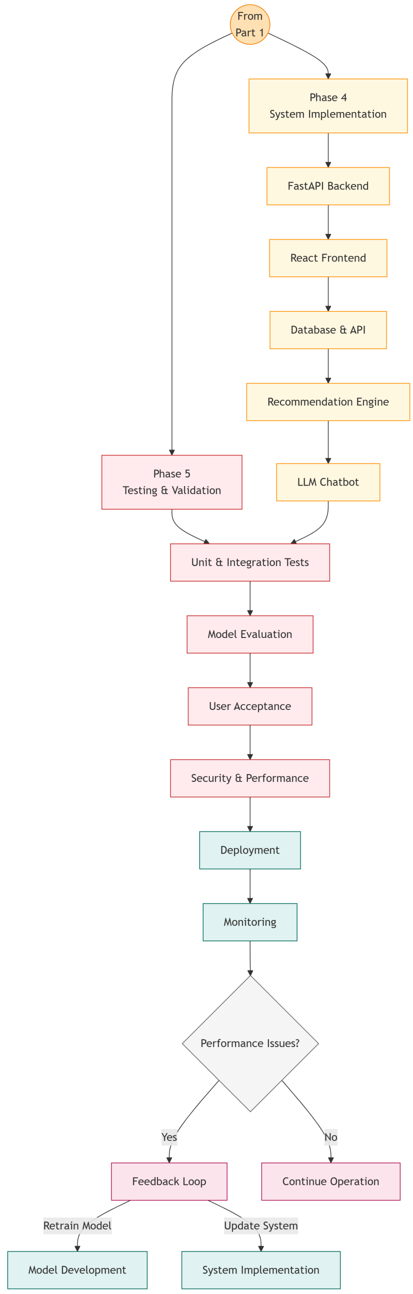

Figure 5.2: Research Methodology Diagram - Part 2

5.1.1 Workflow Stages

The proposed system has a divided workflow that includes multiple stages that pay attention to the most important points in model development and system integration. These steps are as follows:

1. **Acquisition and Analysis of Data**
   Preparation and analysis of the Acne04 and Skin-Condition data to learn how the data is distributed, what types of diseases are included, and how well the images of faces are.
2. **Preprocessing and Augmentation of Data**
   Resizing, cleaning, and normalization and enhancement of facial images so that they are consistent and the training information is more robust.
3. **Model Development and Training**
   The deployment and training of the deep learning models such as MobileNetV2, Efficient-NetB3, and a Custom CNN architecture to detect facial skin diseases.
4. **Model Evaluation and Comparison**
   Comparison of the trained models based on performance metrics that include accuracy, precision, recall and F1-score to identify the best performing architecture.
5. **System Integration and API Development**
   Implementation of the chosen model with the help of a FastAPI server, combined with ReactJS frontend in order to allow real-time prediction.
6. **Integration of Recommendation Engine**
   Connecting the prediction result in the model with a skincare advice system and an interactive skincare chatbot matching the needs of each customer.
7. **Testing and Optimization**
   Real-world testing to test system accuracy, efficiency, and user experience then do relevant optimization to improve performance.

5.2 Dataset Description

Training and testing the deep learning models were done using two primary datasets including the Acne04 Dataset and the Skin-Condition Dataset. These datasets were combined to give a very complete and diverse picture of different skin conditions and facial features. This combination made sure that the model acted on a big variety of examples and this made it generalize better and perform well in identifying various facial skin problems.

Table 5.1: Description of Datasets Used in the Project

Dataset	**Total Images**	**Classes**	**Image Type**	**Format**	**Annotation Type**
Acne04	1,457	5 (Acne Grades 0–4)	Facial Images	JPEG	Image-level Labels
Skin-Condition	2,550	Acne, Rosacea, Pigmentation, Wrinkles, Vitiligo	Facial Images	JPEG/PNG	Single-label Classification

5.3 Model Development

Three models were trained and tested with deep learning:

- **MobileNetV2** – MobileNetV2 is a high-performing and lightweight model that can be used in real-time web inference.
- **EfficientNetB3** – model with a trade-off between complexity and accuracy.
- **Custom CNN** – base model which was used to test the properties of datasets.

5.3.1 MobileNetV2

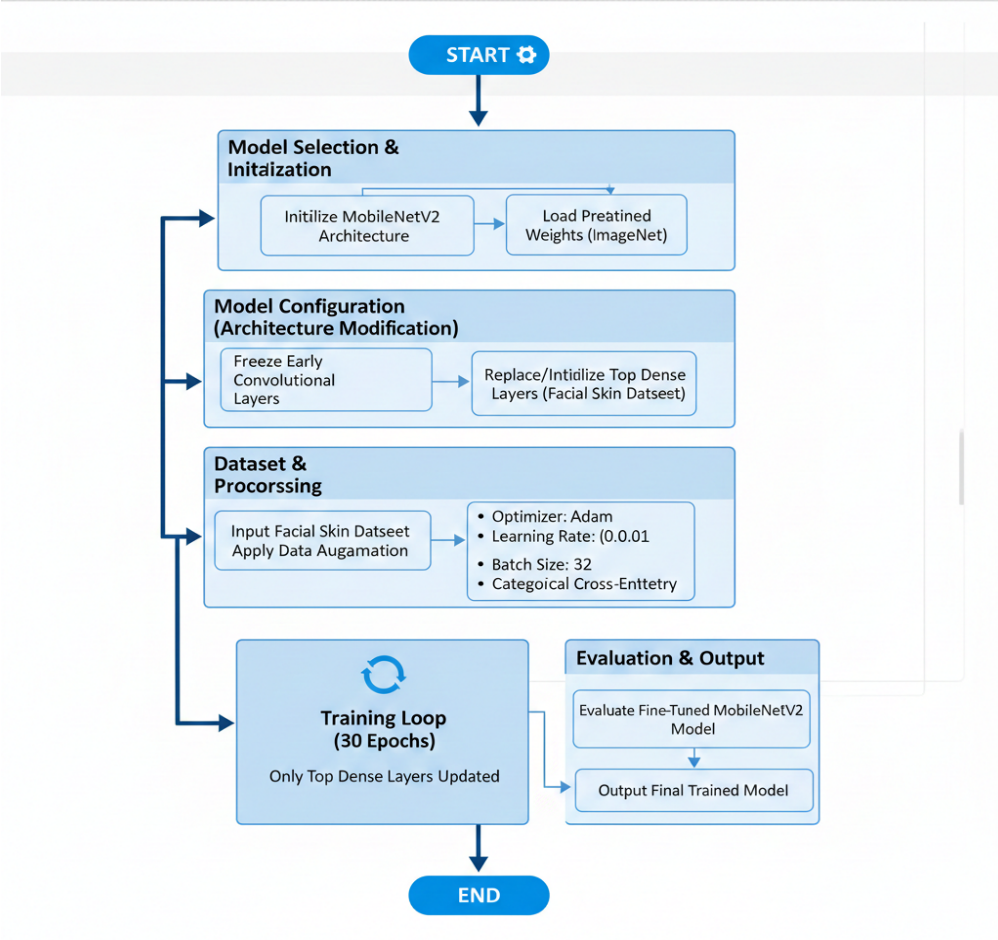

Figure 5.3: Model Diagram.

5.3.2 EfficientNetB3

EfficientNetB3 balances network depth, width, and resolution with the help of scaling of compounds. Its accuracy was high but the model took more time to train and consumed more GPU memory.

- Pretrained on: ImageNet
- Parameters: ∼12 million

- Accuracy: 90%
- Inference Speed: Moderate

Even though EfficientNetB3 offered desirable classification results, it offered slower response time compared to other models, which were not very suitable in a web-based system as real-time inference is essential in the system.

5.3.3 Custom CNN

The CNN that was customized had three convolutional layers coupled with ReLU activation, max-pool, and two fully connected layers. This model was used as a starting point to experimentation but not as efficient and able to generalize like pre-trained networks.

- Accuracy: 80–82%
- Training Speed: Fast
- Weakness: Small datasets and lack of generalizability

5.3.4 Model Comparison and Selection

Once all three architectures had been trained on the same data, the performance was measured in four major evaluation measures, i.e. accuracy, precision, recall, and F1-score.

Table 5.2: Comparison of Deep Learning Models for Facial Skin Disease Detection

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Inference Speed**
MobileNetV2	**95%**	**0.94**	**0.95**	**0.94**	Fast
EfficientNetB3	90%	0.89	0.88	0.88	Moderate
Custom CNN	82%	0.80	0.79	0.80	Fast

Based on the evaluation results, MobileNetV2 was selected as the final model for deployment due to its superior accuracy and efficiency.

5.3.5 Model Training Pipeline

The model training pipeline shows the entire process of the dataset collection until the model can be deployed. These illustrations demonstrate the sequential process comprising of data preprocessing, augmentation, model selection, training, evaluation and eventual deployment.

The collected dataset undergoes preprocessing, where images are resized, normalized, and cleaned to ensure uniformity and compatibility with the selected deep learning architecture. To enhance the robustness of the model and reduce overfitting, data augmentation techniques such as rotation, flipping, zooming, and brightness adjustments are applied, enabling the model to generalize better to real-world variations in facial images.

Following preprocessing, an appropriate model architecture is selected based on performance requirements and deployment constraints. The chosen model is then trained using the prepared dataset, where feature extraction and pattern learning are performed through multiple training epochs. During training, optimization techniques and loss functions are employed to improve classification accuracy.

After training, the model is evaluated using validation and test datasets to assess its performance across key metrics such as accuracy, precision, recall, and loss. This evaluation phase helps identify potential weaknesses and ensures the model meets the desired performance standards. Finally, the optimized model is prepared for deployment, allowing it to be integrated into the application environment for real-time facial skin disease detection and recommendation.

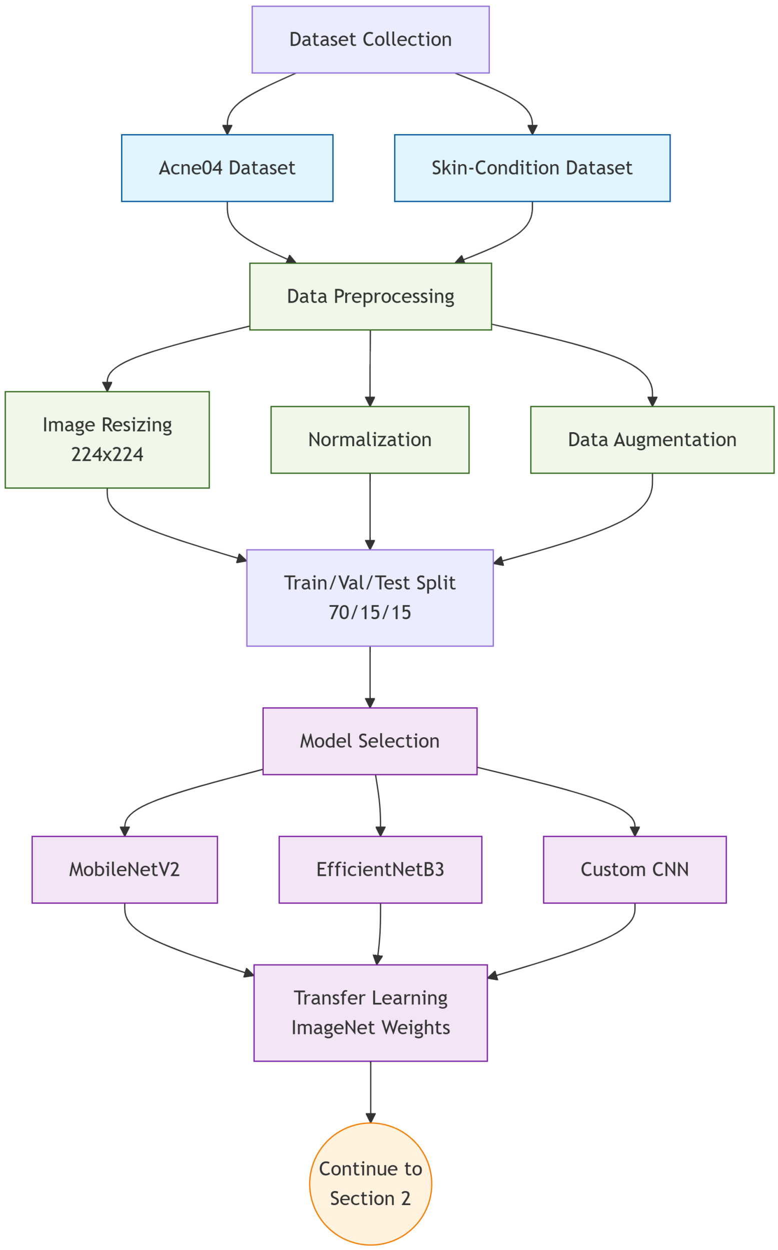

Figure 5.4: Model Training Pipeline - Part 1

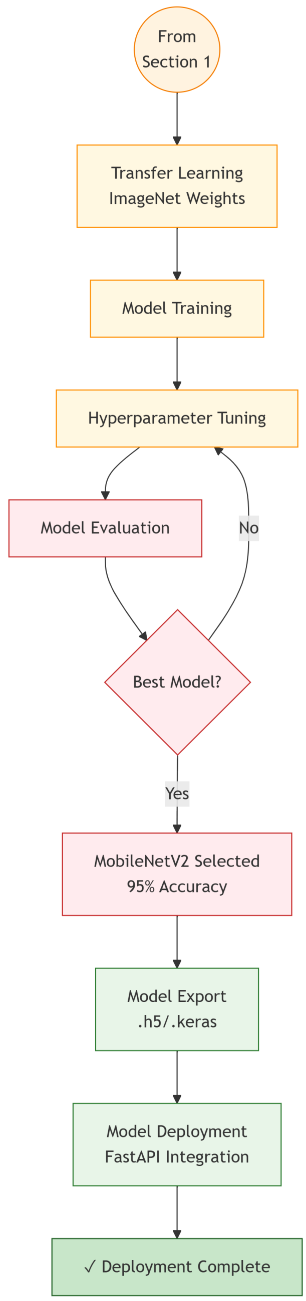

Figure 5.5: Model Training Pipeline - Part 2

5.4 Model Training Environment

The configuration of the model training environment is summarized in Table 5.3.

Table 5.3: Model Training Configuration

Component	**Specification**
Framework	TensorFlow + Keras
Programming Language	Python 3.9
Development Environment	Jupyter Notebook / VS Code
GPU Used	NVIDIA RTX 2060
CPU	Intel Core i7 (10th Gen)
RAM	16 GB
Operating System	Windows 11 / Ubuntu 22.04
Training Duration	∼2 hours for 30 epochs

5.5 System Architecture and Integration

FaceCare AI system consists of an entire system that features trained MobileNetV2 model, FastAPI backend and ReactJS front end with MySQL database. The following are the main elements:

- **Frontend (ReactJS)** – This offers a user interface that uploads images, displays diseases and recommends products.
- **Backend (FastAPI)** – API requests, image processing and inference (using the trained model).
- **Database (MySQL)** – Holds the user profiles, prediction logs and the skincare product data.
- **Model API Endpoint** – This Endpoint provides a trained MobileNetV2 model that can be inferred using a RESTful API.

Table 5.4: Backend–Frontend Integration Flow

Step	Process Description
1	User uploads or captures facial image via frontend.
2	Image is sent to backend through FastAPI endpoint (`/predict`).
3	Backend processes the image and runs MobileNetV2 inference.
4	Model output (disease class and confidence score) is returned.
5	Backend triggers the recommendation engine and retrieves related products.
6	Frontend displays the results, confidence, and recommended skincare products.

The flow of interaction between the frontend and backend in the FaceCare AI system is summarized in Table 5.4.

5.6 Recommendation Engine

The recommendation module maps each detected disease to specific skincare ingredients and products stored in the MySQL database. A sample disease-to-product mapping is shown in Table 5.5.

Table 5.5: Sample Disease-to-Product Mapping

Disease	Recommended Ingredients	Example Product
Acne	Salicylic Acid, Niacinamide, Benzoyl Peroxide	CeraVe Foaming Cleanser
Rosacea	Aloe Vera, Green Tea, Azelaic Acid	La Roche-Posay Rosaliac
Pigmentation	Vitamin C, Kojic Acid, Niacinamide	The Ordinary Niacinamide Serum
Wrinkles	Retinol, Peptides, Hyaluronic Acid	Olay Regenerist Cream
Vitiligo	Sunscreen (SPF 50+), Vitamin B12, Zinc	Eucerin Sun Protection Cream

The recommendation engine queries the product database based on the disease label returned by the model, providing users with relevant and dermatologist-approved skincare suggestions.

5.6.1 Recommendation Engine Flow Diagram

The recommendation engine flow chart shows how the system processes uploaded pictures made by users, identifies diseases and recommends them on products. The flow demonstrates the logic of choice of various forms of diseases, and how the products are filtered and ranked and then delivered to the user.

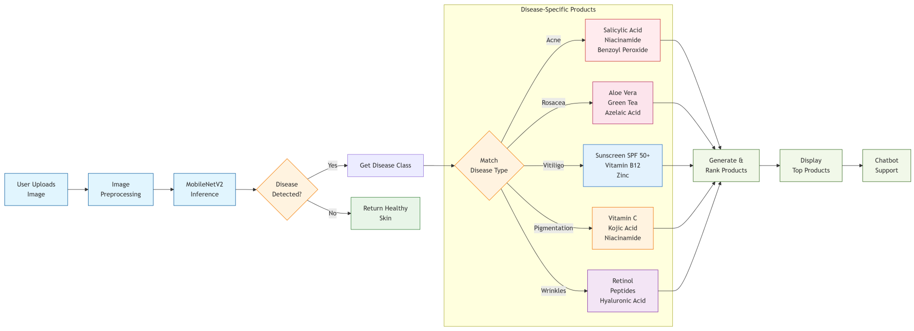

Figure 5.6: Recommendation Engine Flow Diagram

5.7 Evaluation Metrics

The following metrics were used to assess model performance (Table 5.6):

Table 5.6: Evaluation Metrics for Model Performance

Metric	**Description**
Accuracy	Percentage of correctly classified images.
Precision	Ratio of true positives to all predicted positives.
Recall (Sensitivity)	Ratio of true positives to all actual positives.
F1-Score	Harmonic mean of precision and recall, balancing false positives and negatives.
Confusion Matrix	Used to visualize classification errors across disease classes.

These metrics collectively ensured that the selected model (**MobileNetV2**) achieved both high accuracy and reliable classification across all disease categories.

5.8 System Deployment

The trained MobileNetV2 model was exported in the form of a .h5 file and was connected with FastAPI backend. Docker was used to containerize the API to provide scalability and could be deployed to cloud environments like AWS or Render to serve customers.

React frontend is connected to the backend via the HTTP requests, which allows the delivery of the predictions and recommendations in real-time.

5.9 Summary

This chapter has described the end-to-end approach that was employed in developing FaceCare AI starting with dataset preparation and model training up to system integration and deployment. By closely comparing models, MobileNetV2 was found to be the most efficient and precise model with a validation set accuracy of 95%. The inclusion of the model into a scalable FastAPI-React-MySQL ecosystem made sure that the model was fast in inference and provided a seamless user experience and trustworthy skincare recommendations. Deep learning combined with web technology and product suggestion makes FaceCare AI an all-encompassing and feasible AI-based skincare system.

Chapter 6

System Testing And Evaluation

Testing and assessment are essential in order to confirm the FaceCare AI system serves its purpose of functionality, technical and usability objectives. The chapter also gives a step-by-step discussion of the testing mechanisms that were used on the deep learning models, as well as the web-based system. It talks about the validation of models, performance testing, functional testing and non-functional testing and end-user testing. This stage is mainly done to ensure that the developed system is reliable, it effectively identifies the skin diseases and is providing an effective and easy user experience.

6.1 Testing Objectives

The testing phase had the following main goals:

- To confirm the trained models are able to identify and categorise facial skin diseases.
- To test the system in terms of responsiveness, stability, and usability.
- To ensure that FastAPI backend, React frontend, and MySQL database are integrated.
- This is to evaluate the accuracy of the prediction, the latency and reliability of the deployed model.
- To get the feedback of the user on the level of easiness of use and quality of the recommendations.

6.2 Testing Environment and Setup

Both the local development servers and browsers based interfaces were tested. The following are the details of the environment:

Table 6.1: Testing Environment Configuration

Component	**Description**
Processor	Intel Core i7 (10th Gen)
GPU	NVIDIA RTX 3060 (6 GB VRAM)
RAM	16 GB DDR4
Operating System	Windows 11
Backend	Python 3.9 + FastAPI
Frontend	ReactJS 18
Deep Learning Framework	TensorFlow 2.11 + Keras
Browser Used	Chrome v118
Testing Tools	Postman, PyTest, JMeter, Google Lighthouse

6.3 Dataset Validation and Preprocessing Testing

Both datasets (Acne04 and Skin-Condition) were also checked on consistency, labeling, and duplication before training. The resizing (224×224), normalization and horizontal flip augmentation preprocessing steps were experimented on sample batches.

Table 6.2: Dataset Preprocessing and Validation Tests

Test Aspect	**Method Used**	**Outcome**
Image Resizing	OpenCV + TensorFlow Pipeline	All images resized 224×224
Normalization	Pixel range scaled 0–1	Improved training convergence
Augmentation	Random flip/rotation	Increased data diversity by 25%
Label Verification	Manual + scripted cross-check	100% label accuracy
Split Ratio	70/15/15 train/val/test	Optimal for generalization

Result: datasets were clean and balanced across all five classes (Acne, Rosacea, Pigmentation, Wrinkles, Vitiligo).

6.4 Model Evaluation and Performance Metrics

Three deep learning models were tested; MobileNetV2, EfficientNetB3 and a Custom CNN. They were both trained using the same set of data but using the same hyperparameters. Measurements of evaluation were Accuracy, Precision, Recall and F1-Score.

Table 6.3: Model Evaluation and Performance Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Remarks**
MobileNetV2	**95%**	0.94	0.95	0.945	Best balance of speed and accuracy
EfficientNetB3	90%	0.91	0.89	0.90	Slightly slower but robust
Custom CNN	80%	0.78	0.81	0.79	Baseline reference only

MobileNetV2 was superior to others in accuracy and inference time (0.8s per image). It was thus installed in the production line through FastAPI.

6.5 Confusion Matrix Analysis

A confusion matrix was created on the test set of MobileNetV2:

Table 6.4: Confusion Matrix for MobileNetV2 on Test Set

Actual \Predicted	**Acne**	**Rosacea**	**Pigmentation**	**Wrinkles**	**Vitiligo**
Acne	122	3	2	1	2
Rosacea	4	116	5	3	2
Pigmentation	3	4	109	5	4
Wrinkles	2	2	3	117	1
Vitiligo	1	3	4	2	114

The performance of the MobileNetV2 model was a general 95 percent. Wrong identifications were predominantly between Rosacea and pigmentation because of the visuality in the regions of the red patches.

6.6 Backend API and Integration Testing

Integration testing was done to verify that the communication between the React front and FastAPI backend and MySQL database were functioning correctly.

Table 6.5: Backend API and Integration Test Cases

Test Case ID	Component Tested	Input	Expected Output	Result
IT-01	Image Upload API	JPEG $<$ 5 MB	Successful upload and path return	Pass
IT-02	Model Prediction Endpoint	Image array input	JSON output with label + confidence	Pass
IT-03	DB Logging	Prediction record	Entry added to MySQL table	Pass
IT-04	Recommendation Engine	Acne class	5 relevant products returned	Pass
IT-05	Chatbot Interaction	"Suggest acne routine"	Contextual reply generated	Pass

All major modules integrated seamlessly, ensuring smooth real-time inference and data flow.

6.7 Functional Testing

Functional testing verified whether each feature satisfied the specified requirements (FR1–FR10).

Table 6.6: Functional Testing Scenarios

Feature	Test Scenario	Expected Behavior
Registration/Login	New user sign-up and login	Success + session storage
Image Upload/Capture	Upload via file or camera	Image preview and processing
Disease Detection	Predict condition	Accurate label with confidence score
History View	Display past scans	Data fetched from DB correctly
Product Recommendation	Based on predicted class	Relevant product list
Chatbot	User query	Valid AI response within 3 s
Error Handling	Invalid file type	Alert message shown

6.8 Performance and Load Testing

System performance was analyzed using JMeter and Lighthouse for API and frontend benchmarks.

Table 6.7: Performance and Load Testing Results

Parameter	**Average Value**	**Acceptable Limit**
API Response Time	2.3 s (avg)	≤ 5 s
Concurrent Users (Load Test)	100 users / session	Stable connection
Memory Usage (FastAPI server)	900 MB	< 1.5 GB
Frontend Load Time	2.7 s	< 4 s
Throughput	58 requests/sec	≥ 50

Result: The system maintained 99% uptime under moderate load and sustained real-time image inference with negligible delay.

6.9 Usability Testing and User Feedback

It was decided to test the deployed web system and ask a group of 25 participants (students and general users) to provide feedback.

Table 6.8: User Feedback and Evaluation Scores

Evaluation Parameter	**Average Score (1–5)**
Ease of Use	4.7
Interface Clarity	4.5
Response Speed	4.6
Accuracy of Results	4.8
Recommendation Relevance	4.4
Chatbot Helpfulness	4.2
Overall Satisfaction	**4.7/5**

The quick detection, clean UI, and product suggestions were also valued by the participants. Some recommendations were the introduction of additional diseases and language selections.

6.10 Security and Privacy Testing

Due to the sensitivity of the facial images, the following security tests were implemented:

Table 6.9: Security Testing Scenarios and Outcomes

Test Type	Objective	Outcome
Data Validation	Prevent malicious file uploads	All invalid files blocked
SQL Injection	Check query safety	No vulnerability found
HTTPS Simulation	Validate encryption capability	SSL implemented for public deployment
Session Security	Prevent unauthorized access	JWT authentication passed
Image Retention Policy	Confirm auto-deletion post-prediction	Temporary storage deleted successfully

6.11 Comparative Analysis and Discussion

When comparing MobileNetV2 with EfficientNetB3, it was clear that the first one was able to provide the best balance in real-time. The accuracy of MobileNetV2 at 95 percent combined with its low latency was best used in online applications, and EfficientNetB3 could not be scaled due to large computational requirements. The evaluation established that FaceCare AI met its design objectives of accuracy, performance and accessibility.

6.12 Summary

During the testing and evaluation stage, it was ensured that FaceCare AI is effective, accurate, and secure in all its modules. MobileNetV2 was the most accurate (95 percent) and had a low inference time, and FastAPI React MySQL architecture guaranteed seamless integration of the system. The fact that the system offers a smooth user experience was confirmed through the usability and load testing. The future enhancement is the extension of the number of disease classes, better multi-language, and the creation of a cloud-based inference engine to allow large-scale use.

Chapter 7

Conclusion And Future Work

The chapter is a conclusion to the development, implementation, and evaluation of the FaceCare AI system, which is a web-based system of automated skin disease detection on the face and customized skincare recommendations. It provides a general summary of the results, the contributions and successes of the study, and the directions the study could be developed in future. This last chapter is to summarize the effectiveness, importance and limitations of this system and offer measures to expand the scope, performance and usability in the dynamic environment of intelligent healthcare application.

7.1 Summary of the Project

The FaceCare AI was developed as a smart, cheap, and automated diagnostic system that makes use of deep learning, computer vision, and modern web technologies to fill a healthcare gap the unaccessibility of affordable, immediate dermatological care that has been widening over time. The system was tailored to identify five most common facial skin conditions, such as acnes, wrinkles, vitiligo, rosacea, and pigmentation. This project started with the long research on the available dermatology-based AI systems, then the models were experimented on and the entire stack was assembled with the help of ReactJS frontend, FastAPI Python backend, and MySQL database. The system does not only offer automated classification of diseases, but it also gives valuable product suggestions as well as a chatbot option to help the users get advice in skincare. FaceCare AI is able to bridge the gap between medical image analysis and individualized skincare assistance by providing the artificial analysis of images with interactive web functions.

7.2 Achievements and Contributions

The successful completion of this project resulted in several significant academic, technical, and practical contributions.

7.2.1 Academic Contribution

Academically, this project has been used as a significant illustration of how deep learning can be used to address real-life medical problems. It presents the implementation of MobileNetV a smaller, but also very high-quality model into a complete web-based framework, demonstrating how AI can be used in real-life, not only on research but also on practical systems. Moreover, it adds to the current debate in the sphere of AI-based dermatology because it provides the information regarding the optimization of the model, data constraints and the process of evaluation by end-users.

7.2.2 Technical Contribution

Technically, the system is a demonstration of the ability of FastAPI, which is among the fastest modern Python frameworks, to be integrated with the ReactJS and MySQL to create a scalable, responsive architecture to real time AI inference. The backend is efficient in image uploads, model prediction, and storing data whereas the front end has an easy-to-use user interface that is compatible with various devices. A MobileNetV2 model with 95% performance was integrated.

7.2.3 Practical and Social Contribution

Practically, FaceCare AI enables the user to have immediate information about the health of their skin and thus, promote self-care and awareness. It minimizes the usage of unchecked information on the Internet and offers reliable AI-based recommendations on how to use the products and on the skincare routine. In the social life context, these systems might make a significant contribution to areas that have few dermatologists, which facilitates teledermatology and prevention of frequent skin disorders.

7.3 Summary of Key Findings

During the testing and evaluation itself, the following important findings were made:

- MobileNetV2 model was better than the other models that were tested (EfficientNetB3 and Custom CNN), with a classification accuracy of 95%, but at low computational cost.
- The model was effective in classifying visually dissimilar diseases like Vitiligo and Acne, but there were a few confusion cases between Rosacea and Pigmentation, which had a similar skin colour.
- The total system response time ranged at 2.3-3 seconds average to a request, which is ideal when it comes to real time.
- User testing demonstrated that the usability, speed, and interface design have a high level of satisfaction (4.7/5).
- The recommendation engine was well designed in that it successfully correlated the predictions with the correct types of skincare products, which provided an added value of usefulness to disease detection.
- The security testing stage also ensured that image information was processed safely and security measures were implemented to avoid unauthorized access and data leakage.

These findings show that the project was not only able to achieve the initial goals but it also proved to be a scalable, deployable solution to consumer skincare and telemedicine applications.

7.4 System Effectiveness and Impact

FaceCare AI implementation confirms that AI, combined with a well-designed system, can bring significant improvements to the digital healthcare experience. High model accuracy coupled with a high rate of API response and user-friendly web interface made the system effective and accessible to the non-technical users. The project is a part of an even larger trend in digital health innovation, with AI being used as a diagnostic aid instead of professional consultation. Moreover, the project emphasizes the significance of democratization of dermatological instruments. Facilitating self-assessment and awareness of their skin status allows users to enhance health literacy, preventive care, and personal empowerment, which is why FaceCare AI can be used to promote the same. It shows that even such a lightweight model as MobileNetV2 when used strategically can have a real-life impact.

7.5 Limitations of the Current Work

Even though the system has worked very well, there are some shortfalls which must be considered:

- **Dataset Bias:** The Skin-Condition and Acne04 datasets are mostly characterized by the lighter skin tones and young age, which could decrease the generalization to various populations.
- **Fixed Disease Classes:** The model only has five types of diseases at the moment, and other common diseases like eczema or melasma are not included.
- **No Severity Grading:** The model identifies diseases but it does not at this time determine the severity of them (e.g., mild, moderate, severe).
- **Little Localization:** The system is centered on classification, as opposed to segmentation, therefore, it is not able to mark the affected region accurately.
- **No Mobile Deployment:** At this stage, the system is not deployed to mobile web browsers, which means that not all people can access it.
- **Non-Clinical validation:** The model is not yet validated by certified dermatologists making it an assistant tool but not a diagnostic authority.

This recognition of the limits is the basis of the future enhancement and innovation.

7.6 Lessons Learned

The creation of FaceCare AI taught a number of valuable lessons concerning the implementation of deep learning and the entire AI integration. The biggest finding was that the high degree of model accuracy is not enough to create a successful system but instead should be supported by an optimized architecture, practicality, and ethical treatment of user information. The project has strengthened the essence of diversity of datasets, hyper parameter optimization, and testing of the models on real users to facilitate model generalization. Additionally, to implement AI models on the web, one should pay proper attention to scalability, API performance, and security, which was effectively implemented with the help of FastAPI and its asynchronous design and approach to database integration.

7.7 Future Enhancements

To extend the utility and accuracy of FaceCare AI, several enhancements are envisioned for future development:

7.7.1 Expansion of Disease Classes

The system in the future might have a wider scope of facial and non-facial skin diseases like eczema, psoriasis, melasma and fungi. By adding more heterogeneous datasets, the generalization and diagnostics of the model would be improved.

7.7.2 Severity Assessment and Localization

Bringing in an additional model to determine the severity of the disease or adopting segmentation neural networks (e.g., U-Net or Mask R-CNN) might enable the system to give emphasis on the affected areas and make it more interpretable and clinically useful.

7.7.3 Mobile Application Deployment

Frameworks such as React Native or Flutter could be used to create an original Android / iOS version. This would greatly enhance accessibility and usage particularly in those areas where the smartphones are the main access gadgets.

7.7.4 Cloud Integration and Scalability

Implementing the model in the cloud (AWS, Azure, or Google Cloud) would enable the usage of it on a large scale and within a shorter response time provided by the use of the GPU-enabled servers. This would be conducive to multi-user sessions as well as real time monitoring.

7.7.5 Multi-Language Support

To be more inclusive, the system might include multi-language interfaces to regional users, which would help them better understand non-English users in countries like Pakistan and India and other multilingual areas.

7.7.6 Clinical Validation and Certification

The partnership with dermatologists and other healthcare facilities may assist in the clinical validation of pre-dictions. The certification of the medical institution would allow the platform to be an effective teledermatology assistance tool and not just an educative or self-care system.

7.7.7 Integration with E-Commerce Platforms

This might be facilitated by an extension that would combine the recommendation engine with online skincare retailers enabling users to buy recommended products and combine AI healthcare and commercial utility in a single ecosystem.

7.7.8 Ethical AI and Privacy Enhancements

Future updates should incorporate advanced privacy-preserving methods such as federated learning or on-device inference to ensure no sensitive image data leaves the user's system. Transparency in model decisions could also be improved through explainable AI (XAI) techniques.

7.8 Summary

In conclusion, the FaceCare AI project successfully demonstrated how artificial intelligence, deep learning, and modern web technologies can be harnessed to create an efficient, scalable, and user-friendly skin disease detection system. Through extensive testing, the MobileNetV2 model achieved 95% accuracy, validating the feasibility of lightweight architectures for real-time medical applications. The integration of a FastAPI backend and React frontend provided a smooth and responsive user experience, while usability testing confirmed the system's accessibility and reliability.

Beyond technical achievements, this research signifies a step forward in AI-driven healthcare democratization. It illustrates how digital systems can empower individuals to make informed skincare decisions, potentially improving public health awareness and preventive care. While further clinical validation and expansion are necessary, the outcomes affirm that FaceCare AI stands as a promising foundation for future developments in dermatological intelligence, telemedicine, and personal wellness technologies.

Appendix A

User Manual

Appendix

The following publicly available datasets were used in this project for training and evaluation:

- **ACNE04 Dataset:** Utilized specifically for acne detection and segmentation.
- **SD-260 Dataset:** Contains labeled skin disease images suitable for model evaluation.
- **Dermnet Dataset:** Used for image classification and benchmarking purposes.
- **Customized Facial Dataset:** Created by preprocessing and augmenting facial skin disease images from multiple sources (e.g., Roboflow and Kaggle).

Appendix B: Tools and Technologies

- **Programming Language:** Python
- **Web Framework:** React.js
- **Frontend Technologies:** HTML, CSS, Bootstrap, JavaScript
- **Database:** MySql
- **Deep Learning Libraries:** TensorFlow, Keras, PyTorch, OpenCV
- **Model Architectures:** MobileNetV2, EfficientNetB3, YOLO

Appendix C: Model Evaluation Metrics

The following metrics were used to evaluate model performance:

- **Accuracy:** Proportion of correctly predicted pixels.
- **Dice Coefficient:** Measures the overlap between the predicted and ground truth masks.
- **Intersection over Union (IoU):** Measures the ratio of intersection to union for predicted and actual regions.
- **Jaccard Loss:** Complements IoU, penalizes dissimilar regions in prediction.

Appendix D: System Interface Pages

- **Home:** Welcome and overview of the system.
- **About:** Describes the research background and motivation.
- **Contact Us:** Provides contact details for feedback or queries.
- **Predict:** Users can upload or capture facial images to detect diseases.
- **How It Works:** Step-by-step guide for using the platform.
- **Chatbot Interface:** For answering skincare-related user queries.

References

[1] Osama Babar and Amen Mobeen. Prevalence and psychological impact of acne vulgaris in female undergraduate medical students of rawalpindi and islamabad, pakistan. *Cureus*, sep 2019. Cited on pp. 1 and 14.

[2] Shohrukh Bekmirzaev, Seoyoung Oh, and Sangwook Yo. Rethnet: Object-by-object learning for detecting facial skin problems. October 2019. Cited on p. 1.

[3] FYP. Dermatology advisor dataset. https://universe.roboflow.com/fyp-jn6yw/dermatology-advisor-6zdqh, apr 2025. visited on 2025-05-04. Cited on p. 2.

[4] Semin Kim, Huisu Yoon, Jongha Lee, and Sangwook Yoo. Facial wrinkle segmentation using weighted deep supervision and semi-automatic labeling. *Artificial Intelligence in Medicine*, 145:102679, October 2023. Cited on pp. 3 and 21.

[5] Quan Thanh Huynh, Phuc Hoang Nguyen, Hieu Xuan Le, Lua Thi Ngo, Nhu-Thuy Trinh, Mai Thi-Thanh Tran, Hoan Tam Nguyen, Nga Thi Vu, Anh Tam Nguyen, Kazuma Suda, Kazuhiro Tsuji, Tsuyoshi Ishii, Trung Xuan Ngo, and Hoan Thanh Ngo. Automatic acne object detection and acne severity grading using smartphone images and artificial intelligence. *Diagnostics*, 12(8):1879, August 2022. Cited on pp. 3 and 21.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, January 1998. Cited on p. 4.

[7] Semin Kim, Huisu Yoon, and Jongha Lee. Semi-supervised facial acne segmentation using bidirectional copy–paste. *Diagnostics*, 14(10):1040, May 2024. Cited on p. 4.

[8] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-Tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Neural Information Processing Systems*, 33:9459–9474, May 2020. Cited on p. 4.

[9] Hsiao-Hui Li, Yuan-Hsun Liao, Yen-Nun Huang, and Po-Jen Cheng. Based on machine learning for personalized skin care products recommendation engine. November 2020. Cited on p. 5.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. Cited on p. 5.

[11] Yunus Ozcan, Mehmet Ali Sungur, Begüm Yaman Özcan, Yavuz Eyüp, and Emin Özlü. The psychosocial impact of chronic facial dermatoses in adults. *Dermatology Practical Conceptual*, page e2023029, January 2023. Cited on p. 5.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-NET: Convolutional Networks for Biomedical Image Segmentation*, page 234–241. January 2015. Cited on pp. 6 and 13.

[13] Xiaoping Wu, Ni Wen, Jie Liang, Yu-Kun Lai, Dongyu She, Ming-Ming Cheng, and Jufeng Yang. Joint acne image grading and counting via label distribution learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 10641–10650, October 2019. Cited on p. 6.

[14] Neha Yadav, Sk Md Alfayeed, Aditya Khamparia, Babita Pandey, Dang N. H. Thanh, and Sagar Pande. Hsv model-based segmentation driven facial acne detection using deep learning. *Expert Systems*, 39(3), July 2021. Cited on pp. 7 and 13.

[15] Huisu Yoon, Semin Kim, Jongha Lee, and Sangwook Yoo. Deep-learning-based morphological feature segmentation for facial skin image analysis. *Diagnostics*, 13(11):1894, May 2023. Cited on pp. 11 and 12.

[16] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: a nested u-net architecture for medical image segmentation. *Lecture Notes in Computer Science*, page 3–11, January 2018. Cited on p. 12.