GENERATIVE AI FOR HEALTHCARE ASSISTANCE

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Dhruv Yadav (211403), Vanshika Jain (211167)

Under the guidance & supervision of

Mr. Faisal Firdous & Mr. Aayush Sharma

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

May 2025

Supervisor's Certificate

This is to certify that the major project report entitled **'Generative AI for Healthcare Assistance'**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is a Bonafide project work carried out under my supervision during the period from July 2024 to May 2025.

I have personally supervised the research work and confirm that it meets the standards required for submission. The project work has been conducted in accordance with ethical guidelines, and the matter embodied in the report has not been submitted elsewhere for the award of any other degree or diploma.

(Supervisor Signature)
Supervisor Name: Mr. Faisal Firdous
Designation: Assistant Professor
Department: Dept. of CSE & IT

Date:
Place: JUIT

(Supervisor Signature)
Supervisor Name: Mr. Aayush Sharma
Designation: Assistant Professor
Department: Dept. of CSE & IT

Date:
Place: JUIT

Candidate's Declaration

We hereby declare that the work presented in this major project report entitled **'Generative AI for Healthcare Assistance '**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of our own work carried out during the period from July 2024 to May 2025 under the supervision of **Mr. Faisal Firdous** and **Mr. Aayush Sharma.**

We further declare that the matter embodied in this report has not been submitted for the award of any other degree or diploma at any other university or institution.

(Student Signature)
Name: Dhruv Yadav
Roll No.: 211403
Date:

(Student Signature)
Name: Vanshika Jain
Roll No.: 211167
Date:

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

(Supervisor Signature)
Supervisor Name: Mr. Faisal Firdous
Designation: Assistant Professor
Department: Dept. of CSE & IT
Date:
Place: JUIT

(Supervisor Signature)
Supervisor Name: Mr. Aayush Sharma
Designation: Assistant Professor
Department: Dept. of CSE & IT
Date:
Place: JUIT

ACKNOWLEDGMENT

Firstly, we express our heartiest thanks and gratefulness to almighty God for his divine blessing makes it possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor **Mr. Faisal Firdous, Assistant Professor,** Department of CSE and **Mr. Aayush Sharma, Assistant Professor,** Department of CSE, Jaypee University of Information Technology, Waknaghat. Deep knowledge & keen interest of my supervisor in the field of **'Generative AI For Healthcare Assistance'** have been instrumental in carrying out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Mr. **Faisal Firdous,** Department of CSE, for his kind help to finish our project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

Dhruv Yadav
211403

Vanshika Jain
211167

TABLE OF CONTENTS

SR.NO	CONTENT	PAGE NO.
I	SUPERVISOR’S CERTIFICATE	I
II	CANDIDATE’S DECLARATION	II
III	ACKNOWLEDGEMENT	III
IV	TABLE OF CONTENTS	IV
V	LIST OF TABLES	VI
VI	LIST OF FIGURES	VII
VII	LIST OF ABBREVATIONS	VIII
VIII	ABSTRACT	IX
IX	CHAPTER 1: INTRODUCTION	1
X	• INTRODUCTION	1
XI	• PROBLEM STATEMENT	3
XII	• OBJECTIVES	4
XIII	• SIGNIFICANCE AND MOTIVATION	4
XIV	• ORGANIZATION OF PROJECT WORK	5
XV	CHAPTER 2: LITERATURE SURVEY	8
XVI	• OVERVIEW OF LITERATURE SURVEY	8
XVII	• KEY GAPS IN THYE LITERATURE	15
XVIII	CHAPTER 3: SYSTEM DEVLOPMENT	18
XIX	• REQUIRMENTS AND ANALYSIS	18
XX	• PROJECT DESIGN AND ARCHITECTURE	19
XXI	• DATA PREPRATION	21
XXII	• IMPLEMENTATION	22
XXIII	• KEY CHALLENGES	33
XXIV	CHAPTER 4: TESTING	34
XXV	• TESTING STRATEGY	34
XXVI	• TESTCASES AND OUTCOMES	35

XXVII	CHAPTER 5: RESULTS AND EVALUATION	36
XXVIII	• RESULTS	36
XXIX	• COMPARISION WITH EXISTING SOLUTION	44
XXX	CHAPTER 6: CONCLUSION AND FUTURE SCOPE	45
XXXI	• CONCLUSION	45
XXXII	• FUTURE SCOPE	46
XXXIII	REFERENCES	47

LIST OF TABLES

Sr. No	Table No.	Table Name	Page No.
I	2.1	Comparison between different papers	12
II	2.2	Result comparison between different research papers	14
III	2.3	Dataset used in different papers	15
IV	4.1	Testcases	35
V	5.1	Model result comparison	43

LIST OF FIGURES

Sr. No.	Figure no.	Description	Page no.
I	1.1	Different types of skin diseases	2
II	1.2	Different types of skin diseases	2
III	3.1	Disease detection pipeline	20
IV	3.2	Information generation phase	21
V	3.3	Epochs for model training	25
VI	3.4	Project Implementation	29
VII	3.5	Home page	30
VIII	3.6	Webpage to upload image	30
IX	3.7	Result webpage	31
X	3.8	Chat box	32
XI	5.1	Models’ performance comparison	38
XII	5.2	Confusion matrix for efficient net	38
XIII	5.3	Classification report for efficient net	39
XIV	5.4	Confusion matrix for mobile net	39
XV	5.5	Classification report for mobile net	40
XVI	5.6	Confusion matrix for dense net	40
XVII	5.7	Classification report for dense net	41
XVIII	5.8	Confusion matrix for ResNet 50	41
XIX	5.9	Classification report for ResNet 50	42
XX	5.10	Confusion matrix for ensemble	42
XXI	5.11	Classification report for ensemble	43

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
Acc	Accuracy
Mel	Melanoma
BKL	Being keratosis like lesions
SCC	Squamous cell carcinoma
GPU	Graphics Processing Unit
RAM	Random Access Memory
BCC	Basal cell carcinoma
ISIC	International skin image collaboration
HAM10000	Human against machine with 10000 training images
TP	True positive
TN	True negative
FP	False positive
FN	False negative
ML	Machine Learning
SVM	Support Vector Machine
DT	Decision Tree
RF	Random forest
ViT	Vision transformer
NV	Melanocytic Nevus
CNN	Convolutional neural network

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

PLAGIARISM VERIFICATION REPORT

Date: 7th, May, 2025.

Type of Document: B.Tech. (CSE / IT) Major Project Report

Name: DHRUV YADAV **Enrollment No.:** 211403

Contact No: 9810359404 **E-mail:** 211403@JUITSOLAN.IN

Name of the Supervisor (s): Mr. FAISAL FIRDOUS, Mr. AAYUSH SHARMA

Title of the Project Report (in capital letters): GENERATIVE AI FOR HEALTHCARE ASSISTANCE

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/regulations, if I found guilty of any plagiarism and copyright violations in the above major project report even after award of degree, the University reserves the rights to withdraw/revoke my major project report. Kindly allow me to avail plagiarism verification report for the document mentioned above.

- Total No. of Pages: 59
- Total No. of Preliminary Pages: 9
- Total No. of Pages including Bibliography/References: 3

DHRUV YADAV
Signature of Student

FOR DEPARTMENT USE

We have checked the major project report as per norms and found **Similarity Index** ...14...%. Therefore, we are forwarding the complete major project report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Faisal 09-5-2025
Signature of Supervisor **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received On	Excluded	Similarity Index (%)	Abstract & Chapters Details	
	• All Preliminary Pages • Bibliography/ Images/Quotes • 14 Words String		Word Count	
			Character Count	
Report Generated On		**Submission ID**	Page Count	
			File Size (in MB)	

Checked by

Name & Signature **Librarian**

ABSTRACT

This project is about training and comparing deep learning models for precise classification of skin disease from image data. Four commonly used convolutional neural network models—Efficient Net, MobileNet, Dense Net, and ResNet-50—were trained and validated to compare their performance. In addition to the above, a maximum confidence ensemble model was deployed to further increase prediction accuracy. Experimental outcomes indicated that each of the individual models attained very high training accuracies (more than 96%), yet testing accuracies were between 85% and 88%. The ensemble model was better than each of the individual models with a testing accuracy of 90.74%, illustrating enhanced generalization and resilience.

In addition to disease detection, the system also generates informative descriptions about the predicted disease, including its symptoms, causes, and possible treatments. This integration of generative AI aims to assist users by not only identifying the condition but also providing relevant medical information for better awareness and early intervention. The findings highlight the potential of combining classification models with information generation to build an intelligent, user-friendly healthcare assistance tool.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Skin diseases are a common and increasing global health problem that affects anyone from newborn babies to elderly individuals and represents a major burden on patients and healthcare systems. The skin, as the largest organ of the body, is susceptible to innumerable insults (ultraviolet light, infections, genetic tendencies, and environmental toxins), and can manifest a wide variety of skin diseases ranging from common dermatitis and acne to life-threatening skin cancers such as melanoma. Early detection and accurate diagnosis is essential: many skin diseases can be treated when they are in their early stages; however, late diagnosis can lead to increased morbidity and mortality, especially in low-resource environments where access to dermatology services is scarce [2].

The skin ailments statistic represents nearly one out of five patient visits in primary care settings, yet because of overlapping features in lesions, various skin colors, and variance in clinical exposure, the diagnostic accuracy of non-specialist clinicians can be very low (it can range from 24-70%). This discrepancy in skin ailments diagnosis leads to additional patient referrals (unnecessary) and increased health spending (in the US >USD 75 billion each year!). This causes delays to a timely diagnosis for serious conditions, such as malignant melanoma, which is predicted to rise by over 50% globally by 2040 [3].

Artificial Intelligence (AI), namely through new developments in deep learning and computer vision, has influenced change in the healthcare sector. AI has the capacity to analyze enormous amounts of medical data and with that, learn from it, so it is no surprise that AI systems can offer comparable diagnostic performance to that of human experts. In dermatology, AI models can analyze high-resolution images of skin lesions to detect statistical patterns and anomalies that may be difficult to visually detect. AI can provide rapid, consistent, and low-cost diagnostic support, which is especially useful where trained medical personnel are not available. AI is also improving clinical practices and workflows, which could limit diagnostic errors, aid in earlier disease detection, and provide unique, personalized patient care to improve health outcomes at the population level.

The "Generative AI for Healthcare Assistance" seeks to develop a Convolutional Neural Network (CNN)-based two-part model for identifying many skin diseases. After a diagnosis is reached, the system will use generative AI to give someone (patient or medical provider) a wealth of information about the condition: symptoms, treatment options, and prevention information. Rather than stopping with the early diagnosis of a condition, this holistic system's goal is to provide patients with the information they need to better manage their skin conditions.

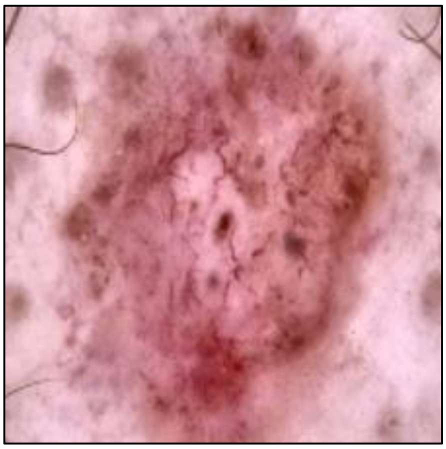

FIGURE 1.1: Different types of skin diseases

FIGURE 1.2: Different types of skin diseases

1.2 PROBLEM STATEMENT

Skin diseases are the most common type of disease affecting more than 1 billion people a year around the world. Rapid diagnosis and treatment are paramount to prevent complications; yet too often many people have to wait before seeking treatment for a skin problem. In areas with limited access to health care resources, it is virtually impossible to come up with the number of dermatological specialists needed to even begin to avoid, let alone prevent, misdiagnosis or delays in treatment. It is suffering longer durations of discomfort, diseases getting progressively worse, and escalating costs of care. Traditional methods of diagnosis can be slow and time-consuming, with the need of an expert practitioner being another bondage, thus many people do not have access to timely care.

One of the major causes of disability worldwide is skin disease. Given that human skin types are distinct from one another, microorganisms and germs can have many opportunities to thrive. Sometimes skin abnormalities can show clear signs of other underlying illnesses. For a skin disease, it is important to discover and properly diagnose it in order to prevent further disease propagation. Research indicates nearly one-half of the 52 percent of skin diseases worldwide require treatment.

Generative AI and effective machine learning frameworks based on convolutional neural networks (CNNs) can solve these challenges through a dual-system model. Not only is it possible to accurately detect and diagnosis skin disorders, but this dual-system model is able to generate patient-specific, comprehensive insights that guide prevention and treatment plans. However, there will be significant logistical, ethical and technical challenges to overcome for such a robust integrated system.

1.3 OBJECTIVES

In this project our main objective is to provide a system that detects and provides information about the detected disease by using CNN models and using generative AI for detailed information about skin disease.

1. **Develop a CNN model to detect skin disease**
   Using pre trained CNN models to create an model capable of detecting various skin disease from images.
2. **Integrate generative AI for information generation**
   Provide detailed, relevant information about the diagnosed skin condition, including symptoms, treatment options and preventive measures using generative AI.
3. **Improve healthcare accessibility**
   Provide a system that allows users to access timely a skin disease diagnosis in early stages.
4. **Enhance patient empowerment and education**
   Provide a tool with comprehensive information about skin conditions, improving their understanding and enabling better self-care and disease management.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

Many people all around the globe belonging to all age groups suffer from different types of skin diseases. Even though skin disease is these common it is very difficult to diagnose and cure them accurately. Especially in areas which are isolated from the world and proper healthcare has not yet reached. This results in the delayed treatment or no treatment at all in some cases. There is an urgent need to resolve and provide a solution for such issues. AI driven solution can help us resolve such issues.

This could be an effective resolution to overcome the previously mentioned challenges when combined with the new generation diagnostic techniques such as convolutional neural networks. The project is significant, in that it provides comprehensive, patient-centered information about diseases diagnosed, including signs, treatment options, and preventive strategies. In addition to enhancing diagnostic accuracy, this important two-pronged approach assures that patients are given both a diagnosis and all necessary information to be able to manage their illnesses effectively.

Moreover, the study is actually going to Democratize Health by giving all the communities, which have not seen high-quality diagnostic tests and other diagnostic tools, access to them. It would globally try to bridge this gap between the impoverished communities and the most advanced medical knowledge to tackle the resolution of healthcare disparities in the world. This is motivated by a vision for using AI in transforming healthcare delivery and inequities and imparting relevant and reliable information to the patients and clinical staff.

1.5 ORGANIZATION OF PROJECT REPORT

The project report is systematically structured into the following sections in order to have clarity and coherence in presenting the work undertaken.

Chapter 1: INTRODUCTION:

This chapter summarizes the project by describing the problem statement, its significance, and the motivation for working on it. This section describes the objective, scope and expected impact of the project and lays the basis for understanding the rationale and the goals of the study. It also introduces the target audience and potential applications of the solution.

Chapter 2: LITERATURE REVIEW:

It elaborates on the current research, tools, and solution in the place within the topic of the project. This chapter critically identifies and accesses the strengths and weakness of the existing systems as well as knowledge gap in the field that the project intends to address. This will establish the basic for the proposed methodologies for the project and ensure relevance and alignment with the industry standards and innovations.

The literature review focuses primarily on detailing the issues found within systems with regard to their performance, scalability, ease of use and accuracy. At the same time, it guarantees the reader that the project will not aimlessly execute a familiar task, instead, it derives purpose dealing with an historically unattended matter. The reader will appreciate that this project is well managed. Later parts of this report will explain how these central ideas frame the decisions on the selection of instruments, algorithms, and strategies for the design.

Chapter 3: SYSTEM DEVELOPMENT:

This section describes the system design and implementation by detailing the system architecture, design principle, detailing of technology used, algorithms used and development process. This includes flowcharts, systems architecture diagrams, system diagrams, and data flow describing the interactions among the different components of the system. The document also reveals major challenges and their respective solutions.

This part justifies the selection of particular technologies, tools, and methods for the system's development. It discusses system designing, integration, and implementation planning at a more granular level with respect to different components of the system. Flowchart and system architecture diagrams offer an integrative viewpoint of the multi-dimensional structure and function interplay. It describes the sequential construction of the system, version control, and the different levels of system functionality validation testing. In this section, all the important issues encountered during the system development are described along with their solutions.

Chapter 4: TESTING:

This chapter addresses the testing methodology to ensure that the system works, perform, and is reliable. This chapter provides descriptions of unit testing, integration testing, system testing, and user acceptance testing. It provides information about test cases, results, identification of issues, and resolution of issues. Performance metrics and benchmarks are also measured to ensure that the system has achieved its objectives. This chapter also includes the tools and settings that were used during testing, and shows how each part of the system was tested on its own as well as by the whole to verify that all works properly. In addition, the chapter shows user feedback to the development team while testing the system and how this helped to further refine the design and functionality of the system. All bugs and defects that we encountered are documented, and we list what was done to resolve them, so that the system is not only working correctly but is also stable and secure enough to be used in real-world situations as well.

Chapter 5: RESULT AND EVALUATION:

This section presents the project outcome with an analysis of performance. It includes results obtained from performance, feedback gathered from users, and comparison with existing systems This section will present that how the project has worked, which will mark the end by how far the objective were achieved by discussing the system's strong and week points along with potential areas for improvement. It looks at whether the system met the goals set at the beginning of the project and where it performed better or worse than expected. Real user experience is shared to give a clear picture of how useful and effective the system is in practical use. This part also talks about any limitations that were faced and gives suggestions for future improvements or features that can be added to make the system even better.

Chapter 6: CONCLUSION AND FUTURE SCOPE:

Core features implemented include security during user registration, logging, and leaderboard management. The project has laid the solid foundation for future scaling: it includes problem solving, real-time leaderboard refreshing, and enhancements of user's profiles by tracking progress through gamification. Other features being implemented are mobile app creation, third-party authentication services, and AI-driven recommendations so as to make it reach the target audience and further boost engagement. With these developments, the platform is expected to be highly competitive feature-rich solution for coding enthusiasts who will feel included and part of a collaborative, friendly, and engaging environment.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVENT LITERATURE

[1] The efficacy of automatic text generation using resources such as the Book Corpus dataset is investigated in the research “Generative AI-Based Text Generation Methods Using Pre-Trained GPT-2 Model” by Rohit Pandey and Hetvi Waghela. The study assesses different approaches with an emphasis on striking a balance between computing efficiency, diversity, and coherence. The study points out enduring issues such sporadic coherence breakdowns, decreased diversity in repeating situations, and high computational costs, even if GPT-2 demonstrates improvements in producing coherent and contextually relevant text. These drawbacks highlight the necessity of more creativity in model architecture, training methods, and assessment criteria. The study lays the groundwork for enhancing generative AI systems in upcoming applications and provides insightful information about existing capabilities.

[3] According to Anna Escalé-Besa et al.'s and colleagues’ paper "The Use of Artificial Intelligence for Skin Disease Diagnosis in Primary Care Settings: A Systematic Review," AI may improve the precision of primary care dermatological disease diagnosis. They reviewed 15 studies (2019–2022) using PRISMA criteria and systematic analytic resources such as PubMed, Scopus, and Web of Science. AI showed accuracy ranging from 0.41 to 0.93 and sensitivity between 58% and 96.1%. Despite its potential for diagnosis and triage, AI has drawbacks, such as study heterogeneity, small datasets, and algorithm dependability across a range of demographics. These deficiencies should be filled by future studies.

[4] Raghav Agarwal and Deepthi Godavarthi's study "Skin Disease Classification Using CNN Algorithms" evaluates 11 CNN algorithms using a dataset of 25,000 photos of eight prevalent skin conditions, along with 1,930 test images. Following pre-processing and image scaling, ResNet152 produced the best accuracy, recall, and precision. The study shows how CNNs can improve clinical decision support systems by performing diagnostic tasks better than dermatologists. Dataset biases and difficulties in practical implementation are among the limitations. Enhancing model interpretability and dataset variety should be the main goals of future research.

[5] Mudassir Saeed et al study "The Power of Generative AI to Augment for Enhanced Skin Cancer Classification" classifies skin cancer using CNN models such as VGG16, VGG19, and a hybrid VGG19+SVM in addition to GANs. The hybrid VGG19+SVM obtained 96% accuracy on ISIC 2019 and 92% accuracy on ISIC 2020 using the ISIC 2019 (25,000 images) and ISIC 2020 (33,000 images) datasets. Model performance and data variety were enhanced via generative AI. High computing needs and dataset imbalance are among the limitations, indicating the need for more generalization and model efficiency research.

[6] A proprietary dataset of 18,550 photos from nine different skin disease classes was utilized in the publication "Multiple Skin-Disease Classification Based on Machine Vision Using Transfer Learning Approach" by Md. Abrar Hamim, Shahadat Hossain Sajim, Fahim Ur Rahman, F.M. Tanmoy. After testing three CNN models—VGG16, VGG19, and MobileNet-v2—MobileNet-v2 had the highest accuracy of 83%. The study highlights how well MobileNet-v2 works for quick and easy skin disease identification. A lack of image variety and dataset diversity are among the limitations. Enhancing dataset heterogeneity and investigating mobile applications for practical applications are the goals of future research.

[7] The study "Innovative approaches for skin disease identification in machine learning: A comprehensive Study" by K.Vayadande studies and compares different machine learning methods that are capable of detecting various types of skin diseases and also focuses on the performance of algorithms like KNN, SVM and complex CNN. It also studies the performance of GAN for generating data that helps to detect disease in an image.

[8] The study "A Deep CNN Transformer Hybrid Model for Skin Lesion Classification of Dermoscopic Images Using Focal Loss" by Yali Nie, Paolo Sommella, Marco Carratù, Mattias O'Nils, and Jan Lundgren uses the SIC 2018 dataset, which consists of 10,015 images in seven categories, to investigate sophisticated techniques for skin lesion classification. CNN models such as ResNet-50 and a hybrid CNN-Transformer model with Focal Loss are used in the study. The hybrid model with Focal Loss outperformed ResNet-50, which had an accuracy of 74%, by achieving 89%. Nevertheless, the study points out a significant drawback: it offers no proof of the model's efficacy in actual clinical situations, highlighting the necessity for additional validation in real-world healthcare settings.

[9] The research titled "Skin Disease Detection Using Deep Learning" authored by S. Inthiyaz et al. uses the Xiangya-Derm dataset (a large dataset made up of more than 150000 clinical images of 543 skin disease types.) to develop an automated system to classify skin disease. The authors use a Convolutional Neural Network (CNN) architecture, pre-processing (with scikit-image) on images made into a predefined pixel resolution of 227 x 227 set by the authors. After that, the authors extract features using a pre-trained ResNet-50 model, with some noteworthy aspects of deep residual learning, and relying on a Softmax layer and Support Vector Machine classifier). The authors attempted to show that the model would be able to discriminate (classify) common skin conditions, like eczema, melanoma, and psoriasis, in addition to healthy skin. Despite the possible ways the proposed method can increase speed or accuracy of diagnostic performance--particularly in an educational or limited-resource setting--the paper lacks several specific and meaningful quantitative evaluation metrics (sensitivity, specificity, and F1-score) and does not provide any assessment of any real-world clinical validation, which is why additional testing and studies on the deployment feasibility of the methods suggested in the paper, is necessary.

[10] A mobile application for skin disease diagnosis is the main goal of MWP Maduranga's project, "Mobile-Based Skin Disease Diagnosis System Using Convolutional Neural Networks (CNN)". Using the HAM10000 dataset, which comprises seven different kinds of skin conditions, the study applies Mobile Net with Transfer Learning to attain an accuracy of 85%. This method demonstrates how lightweight CNN models can be used for effective diagnosis on mobile devices. The study does note one important drawback, though: when the model is transformed to the. Flite format for mobile deployment, its accuracy declines. This emphasizes how optimization strategies are necessary to preserve performance during model conversion and guarantee usefulness in mobile healthcare applications.

[11] The research paper "A Machine Learning Approach for Skin Disease Detection and Classification Using Image Segmentation" authored by Mostafiz Ahammed. proposes a novel approach for skin disease detection based on image segmentation and machine learning classifiers. The authors apply the ISIC 2019 Challenge dataset and the HAM10000 dataset which have a large collection of dermatoscopic images that are divided into 8 classes of skin disease including melanoma (MEL), basal cell carcinoma (BCC), and actinic keratosis (AK). This involves using a series of preprocessing methods there is a removal of digital hair (DHR), which applies a morphological Black-Hat transformation and inpainting algorithm followed by Gaussian filtering to remove noise. Then the authors apply the Grabcut segmentation algorithm which applies k-means clustering and uses the HSV color space, as outlined in the paper, to segment the impacted lesions from the surrounding skin. The machine learning classifiers used include Decision Tree (DT), Support Vector Machine (SVM) and K-nearest neighbor (KNN) and SVM is shown to perform the best overall out of the provided classifiers. However, the authors do not fully describe how they are overcoming the data imbalance challenges between the classes of skin diseases. The authors address the overall data imbalance by applying random oversampling to adjust the number of pixels and account for imbalance which is a great start for further balance techniques to explore through the data to mitigate some of the overfitting. The authors show some promise in their research, but there is a need for further validation, in regard to training, testing, and validating, the models with a more real-world external dataset.

TABLE 2.1: COMPARISION BETWEEN DIFFERENT PAPERS

Author & Year	**Objectives**	**Methods**	**Dataset**	**Results**
Dip Kumar Saha, Ashif Mahmud Joy, Anup Majumder 2024 [2]	YoTransViT: A transformer and CNN method for predicting and classifying skin diseases using segmentation techniques	ViT, MLP, SwinViT	25,331 images, 8 categories	100% - precision 99.97% - accuracy
Yali Nie, Paolo Sommella, Marco Carratù, Mattias O'Nils and Jan Lundgren 2022 [8]	Skin Lesion Classification of Dermoscopic Images Using Focal Loss	CNN (ResNet-50) , ViT, Cross-Entropy (CE), Weighted Cross-Entropy (WCE) Loss, Focal Loss (FL)	ISIC 2018 dataset, 10,015 images, 7 categories	ResNet-50 - 74% Hybrid FL - 89%
Md Abrar Hamim, S hahadat Hossain Sajim, Fahim Ur Rahman, F.M. Tanmoy 2023 [6]	Multiple Skin-Disease Classification Based on Machine Vision Using Transfer Learning Approach.	CNN, Transfer Learning	18550 images (9 types of skin disease)	83%
Raghav Agarwal 2023 [4]	Comparing Different CNN models for skin disease detection	VGG16, VGG19, ResNet, InceptionV3, Dense Net, MobileNet, Xception	25000 images from different Kaggle datasets (8 types of disease)	Resnet 152 (95% train, 73% test)
MWP Maduranga 2022 [10]	Develop an AI-based Mobile Application to detect type of skin disease	MobileNet with Transfer Learning	HAM10000(7 types of disease)	85%

D. N. Vasundara, Swetha Naini, N. Venkata Sailaja, and Sagar Yeruva 2022 [13]	Classification of Skin Diseases Using Ensemble Method.	Ensemble Algorithm	407 images (7 types of skin disease)	96.93%
Mostafiz Ahammed, Md. Al Mamun, Mohammad Shorif Uddin 2022 [11]	A machine learning approach for skin disease detection and classification using image segmentation.	CNN, DNN, KNN, SVM, DT, RF	25331 images (8 types of skin disease)	90%
Syed Imthiyaz 2022 [9]	Skin disease detection using deep learning	Resnet 50	Xiangya-Derm (3 types of skin disease)	87%
Saja Salim Mohammed and Jamal Mustafa Al-Tuwaijari 2021 [15]	Skin Disease Classification System Based on Machine Learning Technique	ML, CNN, Classification	366 images (6 types of skin disease)	94%
Nawal Soliman 2019 [19]	Using CNN and SVM to Predict Skin Disease Using Computer Vision	CNN (Alex Net), SVM	80 images (3 type of disease)	100%
Jessica Velasco 2019 [20]	Making a mobile application using CNN for Skin disease Detection	CNN MobileNet model	3460 images (7 types of disease)	94.4%
Shuchi Bhadula, Piyush Juyal, Chitransh Kulshrestha 2019 [17]	Machine Learning Algorithms based Skin Disease Detection.	Random forest, Naive Bayes, logistic regression, kernel SVM and CNN	3000 images (3 types of skin disease)	96%

TABLE 2.2: RESULT COMPARISON BETWEEN DIFFERENT RESEARCH PAPERS

S. No	Research Paper	Methods	Result
1	[2]	ViT MobileNet InceptionResnetV2	Acc: 99.97% Acc: 98% Acc: 96%
2	[4]	Resnet 152	Acc (Val): 74.24% Acc (Test): 73.45%
3	[6]	VGG 16 VGG 19 MobilenetV2	Acc: 59.33% Acc: 62.24% Acc: 83%
4	[8]	Resnet 50 MobileNet Hybrif+FL	Acc: 74.21% Acc: 83.10% Acc: 89.48%
5	[11]	SVM KNN	Acc: 52% Acc: 42%
6	[13]	Random forest SVM	Acc: 95.09% Acc: 92.63%
7	[17]	Logistic regression Random Forest Kernel SVM Naïve Bayse CNN	Acc: 68% Acc: 67% Acc: 50% Acc: 47% Acc: 96%
8	[18]	Random Forest SVM KNN Naïve Bias	Acc: 82.22% Acc: 85.19% Acc: 79.26% Acc: 65.93%

TABLE 2.3: DATASET USED IN DIFFERENT PAPERS

S. No	Paper	Dataset	Model and Performance
1	[9]	Xiang-Derm	Resnet 50 Acc: 87%
2	[10]	HAM10000	MobileNet Acc: 85%
3	[8]	ISIC 2018	Resnet 50 Acc: 74%

2.2 KEY GAPS IN THE LITERATURE

A number of significant gaps in the literature underscore the need for additional study and development in this field, despite the increased interest in using generative AI to support healthcare.

2.2.1 LIMITED VALIDATION IN THE REAL WORLD:

With little implementation in actual clinical settings, the majority of research on generative AI applications in healthcare is still conducted in controlled or experimental settings. Longitudinal research evaluating the long-term safety, effectiveness, and ethical ramifications of using generative AI to healthcare support are scarce.

2.2.2 ISSUES WITH FAIRNESS AND BIAS:

Biases from training datasets are frequently inherited by generative AI models, which could result in unfair outcomes in healthcare situations. Methods for methodically identifying, mitigating, and preventing biases are not well studied, particularly in diverse and international healthcare populations.

2.2.3 INADEQUATE CLINICAL INTEGRATION STRUCTURES:

There is a dearth of information in the literature about the smooth integration of generative AI into current healthcare procedures, such as electronic health record (EHR) systems. Interoperability issues with existing healthcare systems are not sufficiently addressed.

2.2.4 DATA SECURITY AND PRIVACY ISSUES:

Large volumes of private patient data are used by generative AI, but little is known about protecting privacy and guaranteeing adherence to laws like HIPAA and GDPR. How generative AI systems can safely handle patient data during training and inference is not well covered in research.

2.2.5 INADEQUATE MEASURES OF EVALUATION:

Current generative AI evaluation measures in the healthcare industry place a strong emphasis on technical performance (accuracy, precision, etc.) while ignoring human-centric elements like patient pleasure, usability, and trust. Standardization in evaluating the caliber and applicability of AI-generated outputs in healthcare settings is lacking.

2.2.6 ISSUES WITH SCALABILITY:

The scalability of generative AI systems in healthcare, namely their capacity to manage a variety of medical specialties, languages, and cultural contexts, has not received much attention in research. The computational and infrastructure needs for large-scale generative AI deployment in resource-constrained environments are frequently ignored in research.

2.2.7 IMPLICATIONS FOR ETHICS AND THE LAW:

Few studies go into great detail with the ethical ramifications of generative AI, including decision accountability, culpability for mistakes, and implications for the roles of healthcare providers. There is still much to learn about the legal frameworks controlling generative AI in healthcare, especially with regard to intellectual property rights and malpractice.

2.2.8 LIMITATIONS OF TRAINING DATA:

High-quality, labelled datasets are frequently used in generative AI models, however there aren't enough complete, objective, and representative medical datasets available. Strategies for overcoming data shortage in healthcare, such as data augmentation or synthetic data generation, have not received much attention.

2.2.9 VIEWS OF PATIENTS AND CLINICIANS:

Few studies examine the trust, acceptance, and concerns that patients and physicians have regarding the application of generative AI in healthcare. The dynamics of human-AI interaction in healthcare are still poorly understood.

2.2.10 INSUFFICIENT ATTENTION TO NON-ENGLISH AND LOW-RESOURCE ENVIRONMENTS:

The promise of generative AI in non-English or resource-constrained situations is rarely explored in research, which is mostly focused on English-speaking or well-resourced environments. This worsens healthcare inequities and restricts the current findings' global application.

CHAPTER 3: SYSTEM DEVLOPMENT

3.1 REQUIRMENTS AND ANALYSIS

3.1.1 FUNCTIONAL REQUIREMENTS

- **Skin disease detection:**
  The system should be able to accept an image of a skin lesion and accurately classify it into one of the predefined disease categories using a trained deep learning model (e.g., ResNet, Efficient Net).
- **Disease information generation:**
  Once the disease is detected, the system should fetch or generate relevant information about the disease (such as causes, symptoms, and treatments) using a generative AI model (Google Gemini API).
- **User interface:**
  A web-based interface should be provided for users to upload images and receive results in a readable and informative format.
- **Model integration:**
  The prediction model and the generative API should be seamlessly integrated to provide real-time and coherent results.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

- **Accuracy:**
  The detection model should achieve high accuracy and minimal false positives/negatives, especially for critical conditions like melanoma.
- **Performance:**
  The system should process image input and return both the prediction and information in less than 5 seconds under normal conditions.
- **Scalability:**

The architecture should support adding more disease classes or switching APIs in the future without major changes.

- **Security and privacy:**

  Uploaded images should be securely handled, and user data should not be stored without consent.

- **Responsiveness:**

  The application should be mobile-friendly and responsive across different devices and screen sizes.

3.1.3 TOOLS AND TECHNOLOGIES USED

- Python – For backend logic and model implementation.
- TensorFlow/PyTorch – For training and using the skin disease classification model.
- Google Gemini API – For generating descriptive information about the detected disease.
- Flask – For creating the web-based interface.
- HTML/CSS/JavaScript – For frontend design and user interaction.

3.1.4 PROBLEM ANALYSIS

The dataset contains skin photographs that have been marked with categories of diseases: Melanocytic Nevus (Nv), Melanoma (Mel), Benign Keratosis-like Lesions (bkl), Squamous Cell Carcinoma(scc) and Basal Cell Carcinoma (bcc) and no skin disease detected. However, there are several drawbacks that arise when using this dataset. These involve variation of images size and quality, over representation or under representation of diseases in the sample used, and the need to make fast inference for applications to be useful practically.

3.2 PROJECT DESIGN AND ARCHITECTURE

3.2.1 SYSTEM OVERVIEW

The system is designed to perform two major functions:

1. Detect the type of skin disease from an input image using a deep learning model.
2. Generate relevant information about the detected disease using the Google Gemini API.

This system follows a modular and layered architecture to ensure separation of concerns, scalability, and ease of maintenance.

3.2.2 ARCHITECTURE DIAGRAM

Skin disease detection:

Input Layer: Takes skin disease images as input.

Processing Layer: The system includes a preprocessing pipeline that enhances the contrast of the obtained images and standardizes their size for quality and analytical purposes. The preprocessed images are then fed into an ensemble model for disease classification. This ensemble uses ResNet-50, Efficient Net, Mobile Net, and Dense Net to produce the most accurate predictions possible by leveraging the benefits of a wide range of architectures.

Output Layer: Predicts the class of the image fed into the model in the input layer.

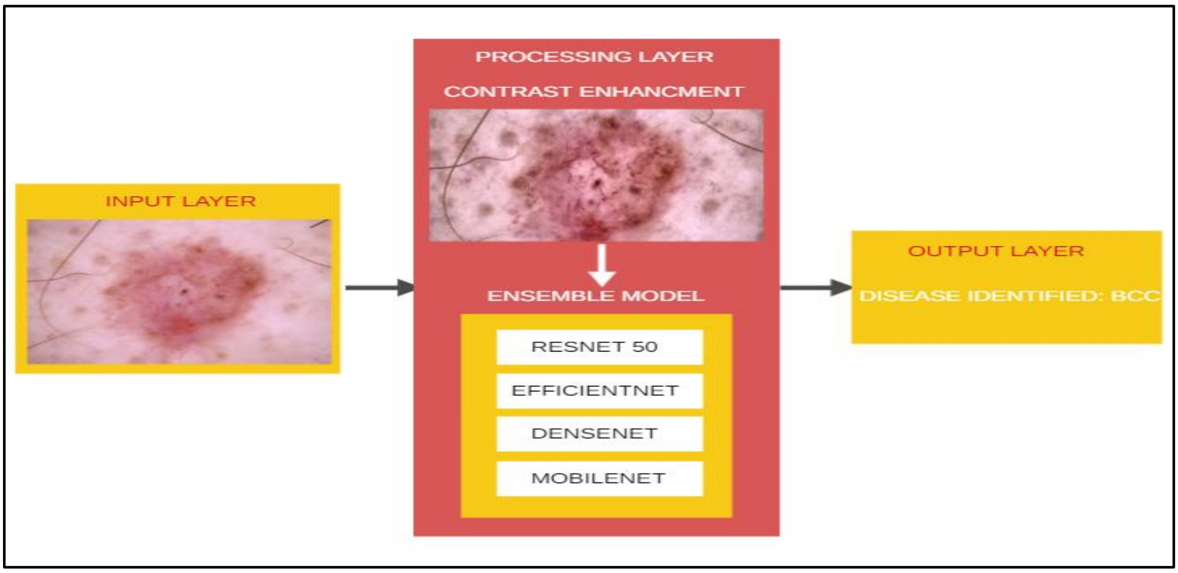

FIGURE 3.1: DISEASE DETECTION PIPELINE

Generating disease information:

Input layer: takes disease name as input.

Processing layer: sends a structured prompt message with the disease information to the google Gemini through API and requests for information-like symptoms, medication etc.

Output layer: displays all the received information about the skin disease in structured format on user interface.

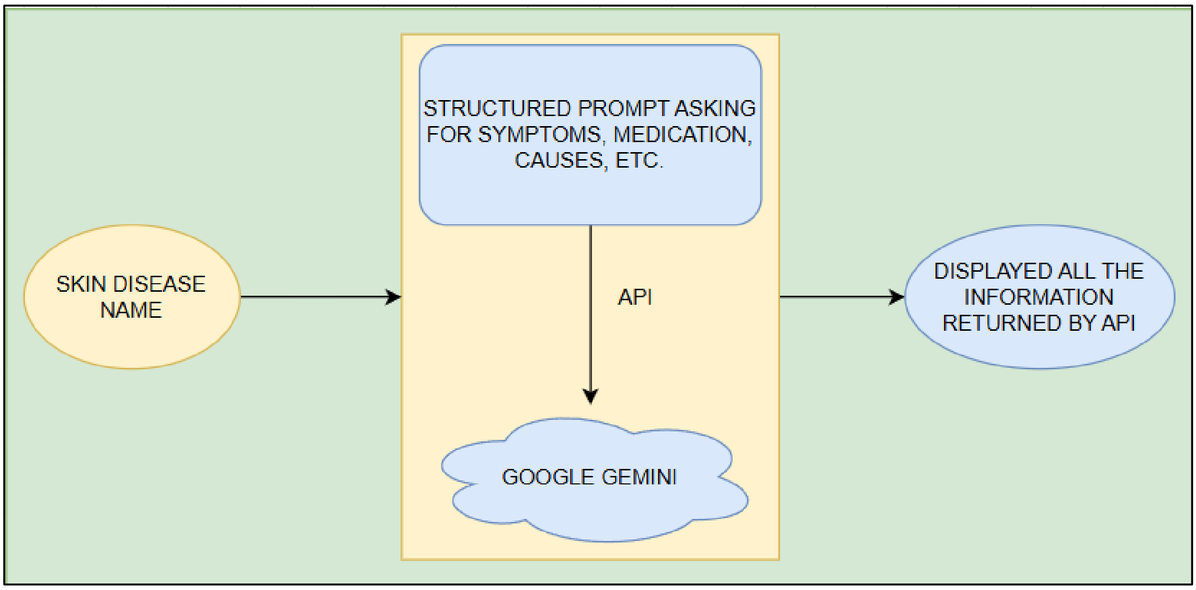

FIGURE 3.2: INFORMATION GENERARTION PHASE

3.3 DATASET PREPRATION

This dataset is collected from ISIC 2019 and HAM10000, which includes 5 diseases categories Melanocytic Nevus (Nv), Melanoma (Mel), Benign Keratosis-like Lesions (bkl), Basal Cell Carcinoma (bcc), and Squamous Cell Carcinoma (scc). The dataset contains a sum of 12,000 images, perfectly balanced in all classes, after augmentation. This data has been split into training, validation, and testing in the ratio of 70:15:15. It includes preprocessing techniques such as resizing all images to a standard dimension of 224x224 pixels, then applying CLAHE on it for contrast enhancement, and normalizing pixel values between 0 and 1. In order to experience the real world's variation and variability while increasing the dataset, data augmentation approaches such as rotation, flipping, zooming, and color jittering are used additionally.

Also, a class (no disease detected) was added with images of random things like walls, roads, room and animals so that the model can detect that the case where the image does not belong to any skin disease category it was trained for. For these 100 images for each category (animals, roads, rooms) were collected and taken from Kaggle and mixed together.

3.4 IMPLEMENTATION

3.4.1 DATA COLLECTION AND PREPROCESSING

The first thing that was done for implementing the project was to develop a dataset, which is a comprehensive dataset of skin images. Various preprocessing techniques were then applied to this dataset to make it consistent and of good quality. Images are resized to standard dimensions of 224x224 pixels, contrast enhancement was applied across images using CLAHE (Contrast Limited Adaptive Histogram Equalization), and normalization made sure the pixel values are scaled between 0 and 1. For variability and robustness improvement, data augmentation techniques included horizontal and vertical flipping and 30% zooming to mimic real-world conditions.

Five skin diseases were identified and then equal number of images for each disease were collected from HAM10000 and ISIC2019 datasets. Diseases which were focused are: Melanocytic Nevus (Nv), Melanoma (Mel), Benign Keratosis-like Lesions (bkl), Basal Cell Carcinoma (bcc), and Squamous Cell Carcinoma (scc) and another class for no disease detected was also added in which random images were added to create a class in case the image has no disease.

After pre-processing, the dataset was split further into three parts-training (70%), validation (15%) and testing (15%)-so that the models can be properly evaluated with the help of these three classifications by splitting the entire dataset into three categories.

3.4.2 MODEL TRAINING

Four pre-trained deep learning models were fine-tuned for skin disease classification: ResNet50, Efficient Net, Dense Net and Mobile Net. Each model was trained separately on a training dataset to learn the distinguishing features of different skin diseases. Then models' performance was tested on test dataset and parameters like precision, recall, f1-score, accuracy was observed for each model individually. Performance graphs were created to compare and find out the best performing model and necessary conclusions and changes like parameters tuning, change in learning rate were made and performance after the changes were observed.

Each model was trained on complete dataset for 10 epochs and after that confusion matrix was compared for each model on testing data.

Code snippet for model training:

```
for epoch in range(num_epochs):
    print(f'Epoch {epoch + 1}/{num_epochs}')
    print('-' * 10)
    # Initialize accuracy for current epoch for both phases
    epoch_train_acc = 0.0
    epoch_val_acc = 0.0
    for phase in ['train', 'val']:
        if phase == 'train':
            model.train()  # Set model to training mode
        else:
            model.eval()   # Set model to evaluation mode
        running_loss = 0.0
        running_corrects = 0
        for inputs, labels in dataloaders[phase]:
            inputs = inputs.to(device)
            labels = labels.to(device)
            optimizer.zero_grad()
```

```
            with torch.set_grad_enabled(phase == 'train'):
                outputs = model(inputs)
                _, preds = torch.max(outputs, 1)
                loss = criterion(outputs, labels)
                if phase == 'train':
                    loss.backward()
                    optimizer.step()
            running_loss += loss.item() * inputs.size(0)
            running_corrects += torch.sum(preds == labels.data)
        epoch_loss = running_loss / len(image_datasets[phase])
        epoch_acc = running_corrects.double() / len(image_datasets[phase])
        print(f'{phase} Loss: {epoch_loss:.4f} Acc: {epoch_acc:.4f}')
        if phase == 'train':
            epoch_train_acc = epoch_acc
        else:
            epoch_val_acc = epoch_acc
            # Save the best model (entire model) based on validation accuracy
            if epoch_acc > best_acc:
                best_acc = epoch_acc
                torch.save(model, model_save_path)
                print(f"Best model saved with validation accuracy: {best_acc:.4f}")
    # Record the accuracies for this epoch
    training_accuracies.append(epoch_train_acc)
    validation_accuracies.append(epoch_val_acc)
```

```
model.safetensors: 100%                                    14.2M/14.2M [00:00<00:00, 23.8MB/s]
Epoch 1/10
----------
train Loss: 0.6254 Acc: 0.7909
val Loss: 1.9075 Acc: 0.6989
Best model saved with validation accuracy: 0.6989
Epoch 2/10
----------
train Loss: 0.3002 Acc: 0.8941
val Loss: 2.0308 Acc: 0.6896
Epoch 3/10
----------
train Loss: 0.1838 Acc: 0.9347
val Loss: 2.2269 Acc: 0.7055
Best model saved with validation accuracy: 0.7055
Epoch 4/10
----------
train Loss: 0.1561 Acc: 0.9469
val Loss: 2.3978 Acc: 0.7032
Epoch 5/10
----------
train Loss: 0.1333 Acc: 0.9515
val Loss: 2.1349 Acc: 0.7203
Best model saved with validation accuracy: 0.7203
Epoch 6/10
----------
train Loss: 0.1275 Acc: 0.9550
val Loss: 3.0717 Acc: 0.6923
```

FIGURE 3.3: EPOCHS FOR MODEL TRAINING

3.4.3 ENSEMBLE MODEL

For improving the classification accuracy further, an ensemble model was built. The ensemble applied a max-confidence technique which accepted four individual models' predictions and class with maximum confidence score as the final result. This basically utilized strengths of each model to yield more accurate and reliable predictions. Rigorous tests were performed on the ensemble model on the testing dataset, which showed its ability to classify skin diseases consistently and accurately after integrating the predictions of various architectures.

Code snippet for ensemble model (max confidence):

```
def ensemble_predict_max_confidence(models, dataloader, device):
    all_labels = []
    all_preds = []
    with torch.no_grad():
        for inputs, labels in dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)
            max_confidence = None
            final_preds = None
            for model in models:
                # Get the softmax probabilities
                outputs = torch.softmax(model(inputs), dim=1)
                confidences, preds = torch.max(outputs, 1)  # Get max confidence and corresponding predictions
                if max_confidence is None:
                    max_confidence = confidences
                    final_preds = preds
                else:
                    # Update predictions where confidence is higher
                    update_mask = confidences > max_confidence
                    max_confidence[update_mask] = confidences[update_mask]
                    final_preds[update_mask] = preds[update_mask]
            all_labels.extend(labels.cpu().numpy())
            all_preds.extend(final_preds.cpu().numpy())
    return np.array(all_labels), np.array(all_preds)
```

3.4.4 DISEASE INFORMATION GENERATION

After the detection of the disease model returns the name for the predicted disease. Using google Gemini API a structured prompt is send asking for various information about the disease detected. For example, causes, symptoms, diagnosis, treatment, complications etc. API returns all the information in a Jason format which is later reorganized and displayed on the webpage in a structured format. Also, similar images for the same disease are also displayed at the bottom of the webpage.

A chat box is also provided thorough which user can ask queries that arise in his mind and the chat box is also connected to google Gemini through backend which provides answers to the queries using API.

Code snippet for disease information generation:

```
def get_disease_info(disease):
    """
    Fetches structured medical information about a disease using Gemini-1.5 Pro.
    """
    prompt = f"""
    Act as a medical expert. Provide structured information about {disease} in **strict JSON
format**.
    ```json
    {{
        "Disease Name": "{disease}",
        "Description": "Brief overview of the disease",
        "Causes": "Main causes",
        "Symptoms": ["Symptom 1", "Symptom 2", "Symptom 3"],
        "Diagnosis": "How it is diagnosed",
        "Treatment": "Available treatments",
        "Prevention": ["Preventive measure 1", "Preventive measure 2"],
        "Severity Levels": "Different levels of severity",
        "Contagious": "Yes/No and how it spreads",
        "Complications": "Possible complications if untreated"
```
```

```
    }}
    ```
    Do not include citations or references. Just return a valid JSON response.
    """

    headers = {"Content-Type": "application/json"}
    data = {
        "contents": [{"parts": [{"text": prompt}]}],
        "generationConfig": {"temperature": 0.3, "maxOutputTokens": 1000}
    }
try:
        print(f"\nDEBUG: Sending request to API: {GEMINI_API_URL}\n")
            response = requests.post(f"{GEMINI_API_URL}?key={API_KEY}", json=data,
headers=headers)

        print("\nDEBUG: API Response Code:", response.status_code)
        print("DEBUG: Full API Response:", response.text[:500])

        if response.status_code == 200:
            result = response.json()
            raw_text = result["candidates"][0]["content"]["parts"][0]["text"]

            json_match = re.search(r"```json\n(.*?)\n```", raw_text, re.DOTALL)
            if json_match:
                return json.loads(json_match.group(1))

            print("DEBUG: JSON not found in API response.")
        else:
            print(f"ERROR: API Request Failed - {response.status_code} - {response.text}")
```
```

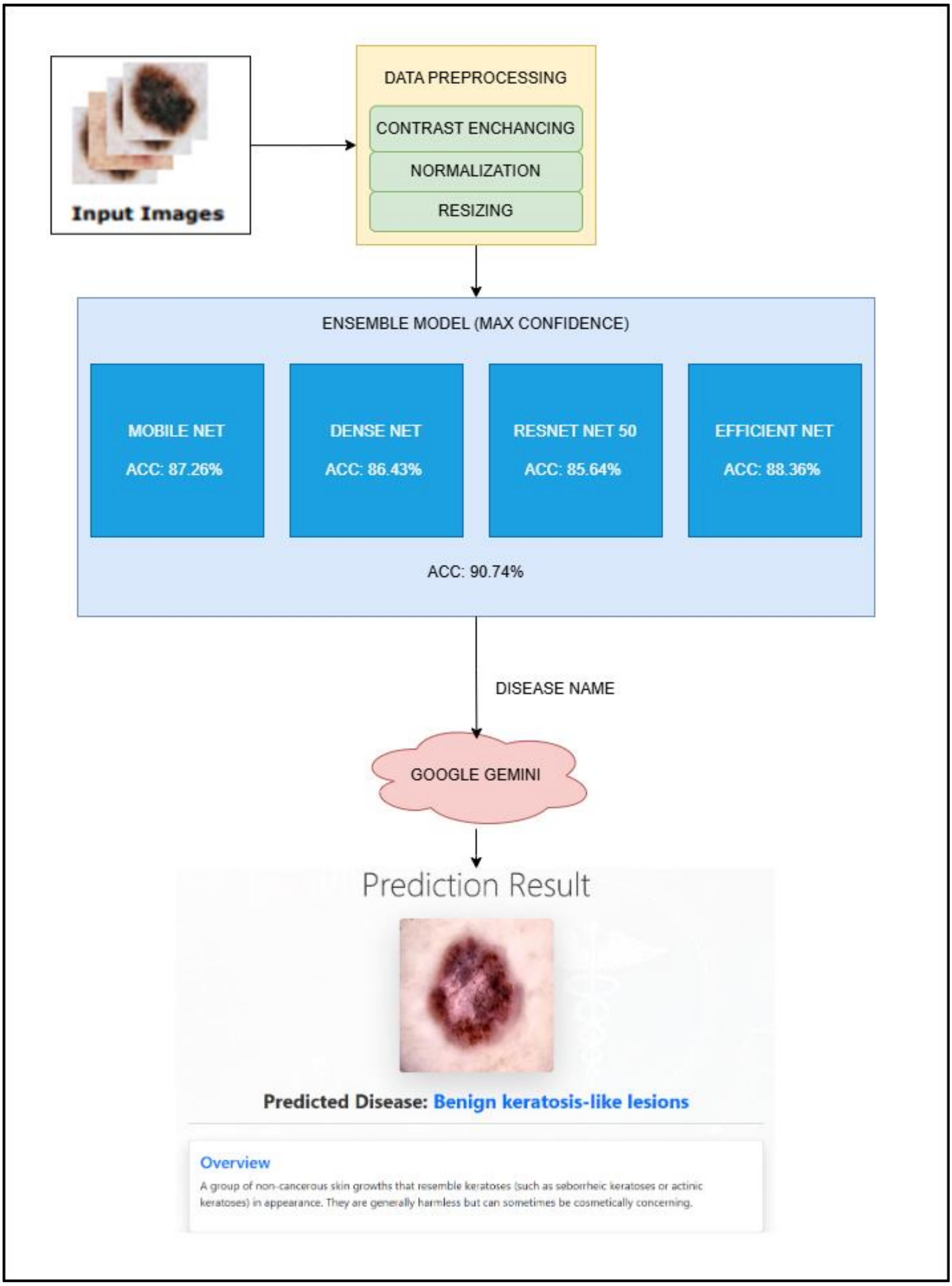

FIGURE 3.4: PROJECT IMPLEMENTATION

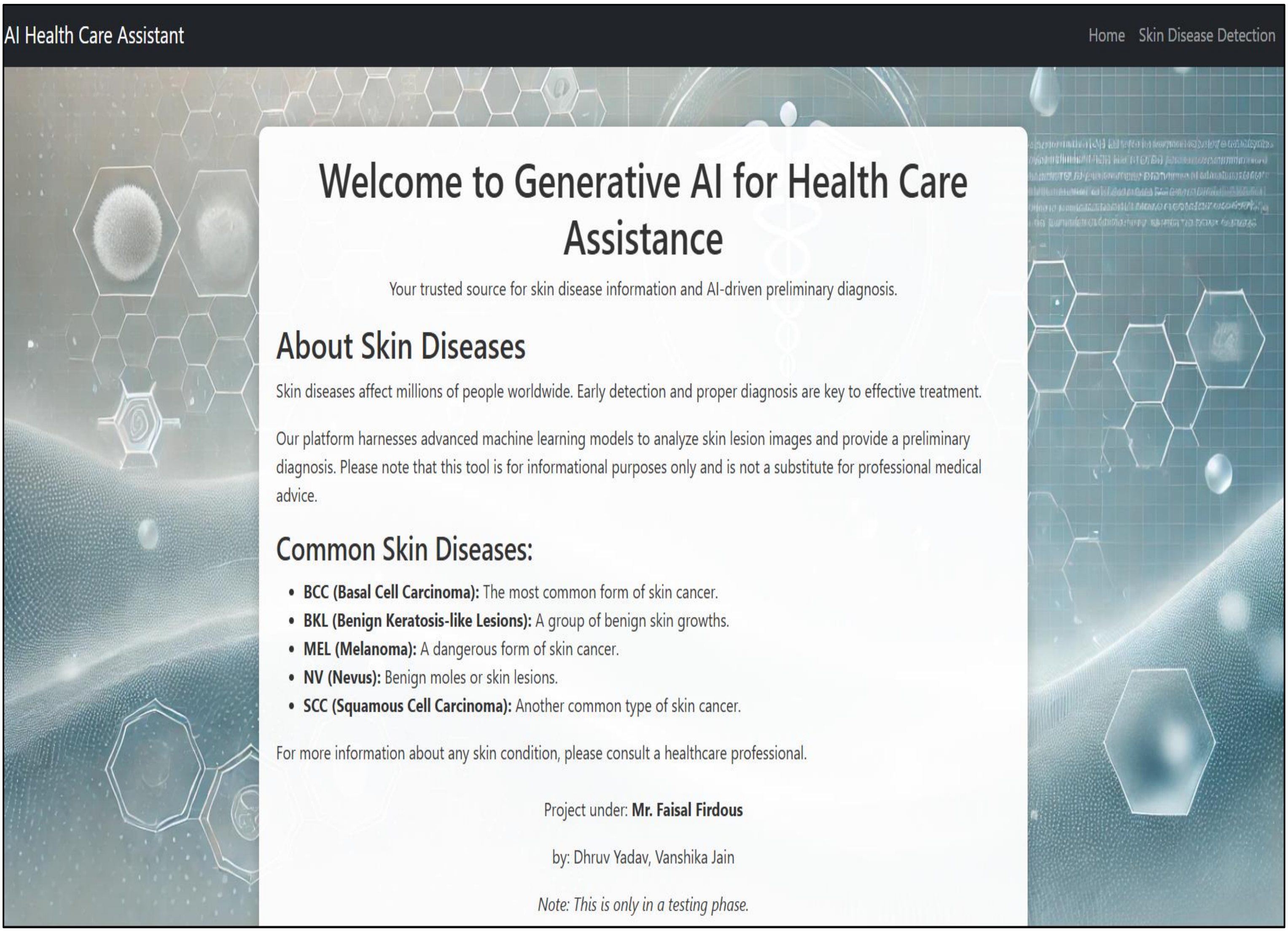

FIGURE 3.5: HOME PAGE

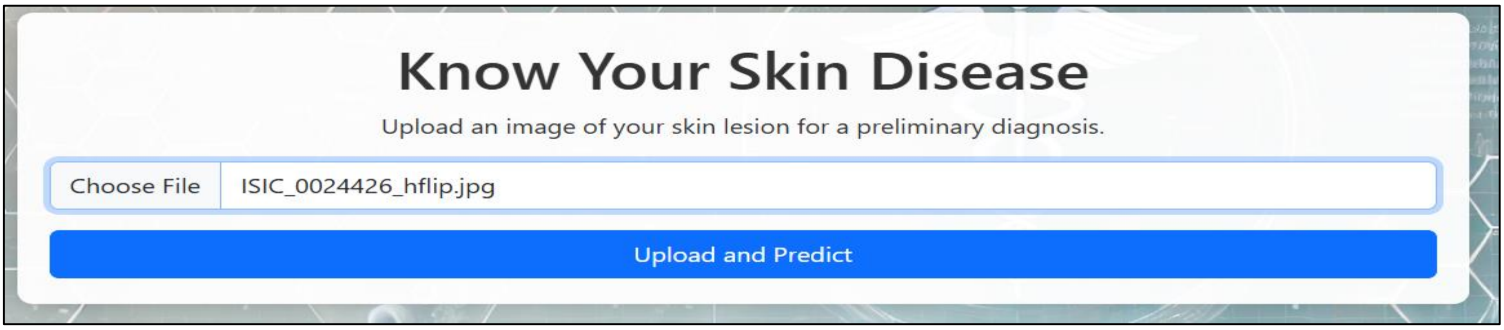

FIGURE 3.6: WEBPAGE TO UPLOAD IMAGE

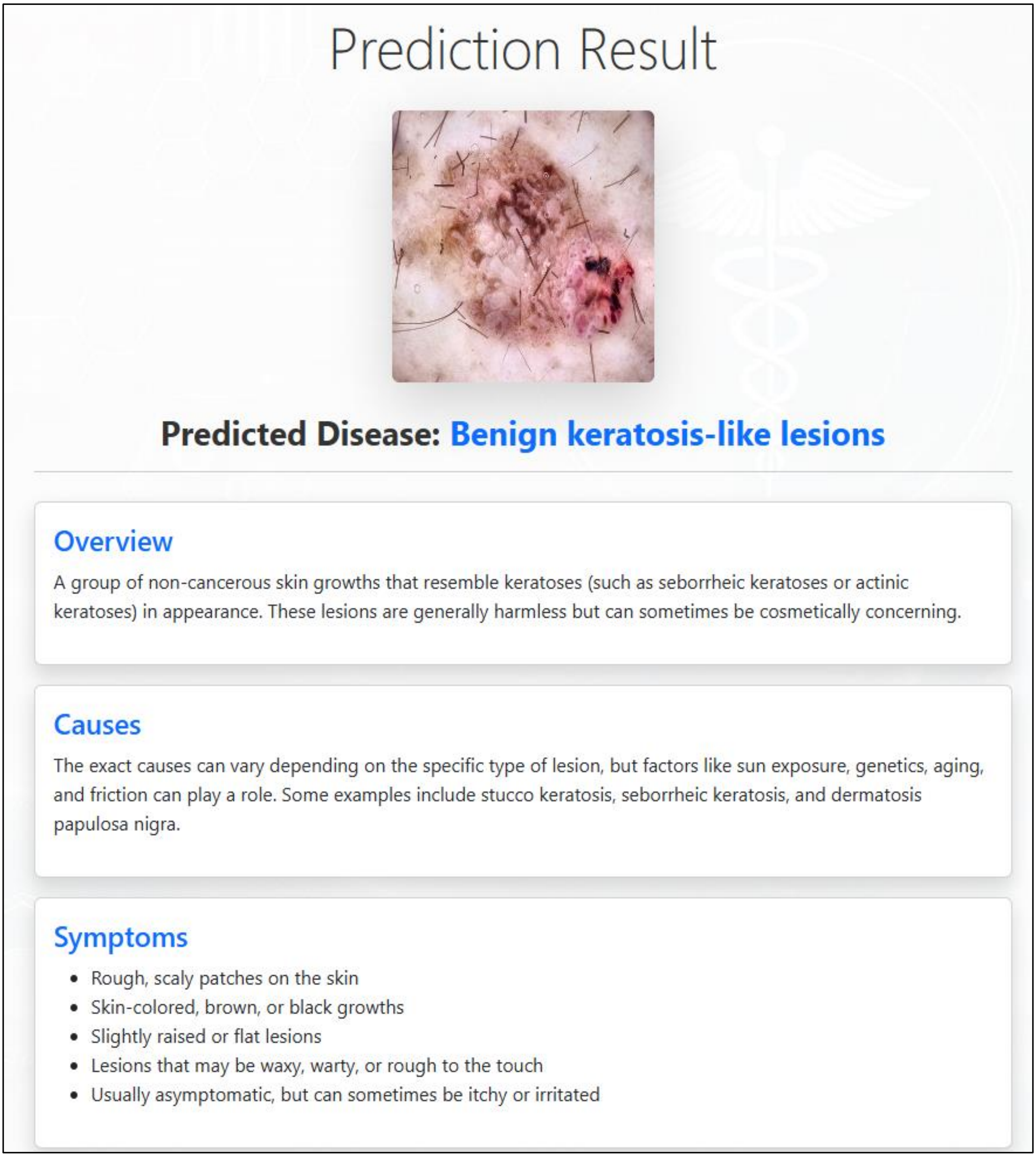

FIGURE 3.7: RESULT WEBPAGE

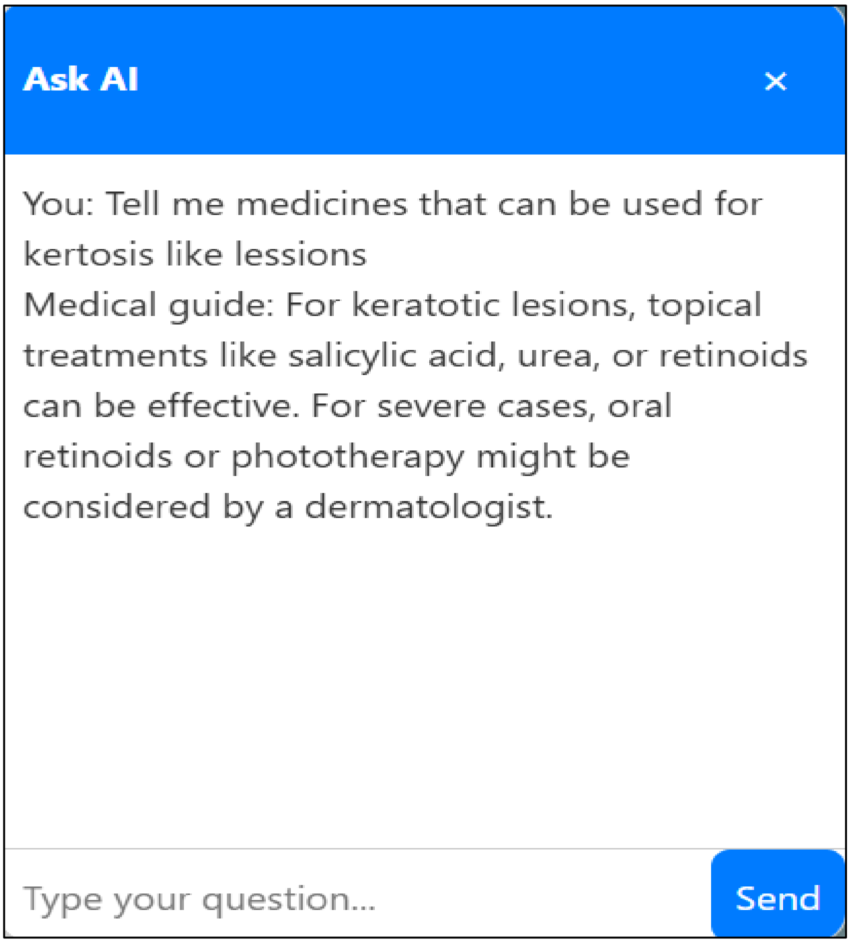

FIGURE 3.8: CHAT BOX

3.4.5 TOOLS AND TECHNIQUES

User Interface (Frontend):

- Developed using HTML, CSS, and JavaScript.
- Allows users to upload skin images and view results.

Backend (Flask Server)

- Receives the image via HTTP POST request.
- Handles image preprocessing and prediction.
- Sends detected disease label to Google Gemini API.

Model training:

- Python:

- PyTorch: Used for constructing, training, and fine-tuning deep learning models
- Torch Vision: A part of PyTorch, utilized for accessing pre-trained models and image preprocessing utilities.
- Google Collab: Provided cloud-based GPUs (e.g., NVIDIA Tesla T4) for efficient model training and testing.

3.5 KEY CHALLENGES

3.5.1 DATASET IMBALANCE

Challenge: Some classes like melanoma (Mel) had a higher number of images than other classes which creates an issue of imbalance data.

Solution: To solve this, we increased the number of images of other classes by oversampling and taking images from different datasets on combining them with our dataset.

3.5.2 COMPUTATIONAL LIMITATIONS

Challenge: Training multiple deep learning models on high-resolution skin images demanded significant computational resources, making it challenging to experiment with different architectures and hyperparameters efficiently.

Solution: This challenge was mitigated by leveraging Google Collab's cloud GPU resources, including NVIDIA Tesla T4 GPUs, to accelerate training. Batch sizes and learning rates were optimized to fit within GPU memory constraints, ensuring smooth model training without exhausting computational resource.

CHAPTER 4: TESTING

4.1 TESTING STRATEGY

To ensure the functionality, accuracy, and robustness of the system, the following testing strategies were adopted:

4.1.1 UNIT TESTING

- Each individual module (image upload, preprocessing, model prediction, API integration) was tested independently.
- Purpose: Ensure each component functions correctly in isolation.

4.1.2 INTEGRATION TESTING

- Modules were tested together to ensure data flow between components (e.g., model prediction passing result to Gemini API) was seamless.

4.1.3 FUNCTIONAL TESTING

- The entire application was tested end-to-end to verify whether it meets functional requirements, such as: uploading an image, getting the correct prediction and displaying correct and accurate disease information.

4.1.4 PERFORMANCE TESTING

- Time taken for prediction and information generation was measured to ensure real time usability.
- Goal: Keep response time under 10 seconds.

4.2 TESTCASE AND OUTCONES

TABLE 4.1: TESTCASES

Test case ID	**Description**	**Input**	**Expected Output**	**Actual Output**	**Result**
TC01	Valid skin image upload	Image of melanoma	Melanoma predicted	Melanoma predicted	PASS
TC02	Upload of image with no clear lesion	Blurred image	"Uncertain" or low-confidence prediction with fallback message	"Cannot detect disease confidently" message shown	PASS
TC03	Gemini API response handling	Disease: "Nevus"	Correct paragraph about Nevus from Gemini	Gemini returned informative paragraph about Nevus	PASS
TC04	Time efficiency test	Any valid image	Results returned in < 10 seconds	Results returned in approx. 7 seconds	PASS

CHAPTER 5: RESULTS AND EVALUATION

5.1 RESULTS

5.1.1 ACCURACY

Accuracy is a performance metric that indicates how well a model correctly predicts the outcomes. It is calculated as the ratio of correctly predicted observations to the total observations. In the context of classification models, it shows the percentage of input samples for which the model's prediction matches the actual label. Higher accuracy reflects better model performance, but it should be evaluated along with other metrics if the dataset is imbalanced.

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$

5.1.2 PRECISION AND RECALL

Precision measures the accuracy of positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives.

$$PRECISION = \frac{TP}{TP + FP}$$

Recall measures the model's ability to detect all actual positive cases. It is the ratio of correctly predicted positive observations to all actual positives.

$$RECALL = \frac{TP}{TP + FN}$$

5.1.1 INDIVIDUAL MODEL PERFORMANCE

Individual models have been evaluated, and there are differences in the amounts of performance exhibited by the models relating to skin disease detection. Efficient Net stands out amongst the models with an impressive testing accuracy of 88.36%. Notably, this model is seen to generalize very well on previously unseen test data. ResNet-50 and Dense Net models presented a strong performance during training, with training accuracies of 98.44% and 97.88%, respectively. However, their accuracies on testing were significantly lower at 85.64% and 86.43%, respectively, indicating that both models were slightly overfitted. Although Mobile Net had a testing accuracy of 87.26%, it showed a fair balance in between the training and the testing performance levels.

5.1.2 ENSEMBLE MODEL PERFORMANCE

According to the successful approach, the development of an Ensemble Model using max confidence, including the outputs of different models, will give higher accuracy results compared to individual models. Testing based on different models yields a maximum result of 90.74%; hence it is successful over other models. Such results make it robust in ensemble learning, especially regarding the variability of data quality and providing consistency in predictions across various disease classes. The resulting performance of the ensemble model sets it apart as one that can draw on the complementary strengths of ResNet-50, Efficient Net, Dense Net, and Mobile Net.

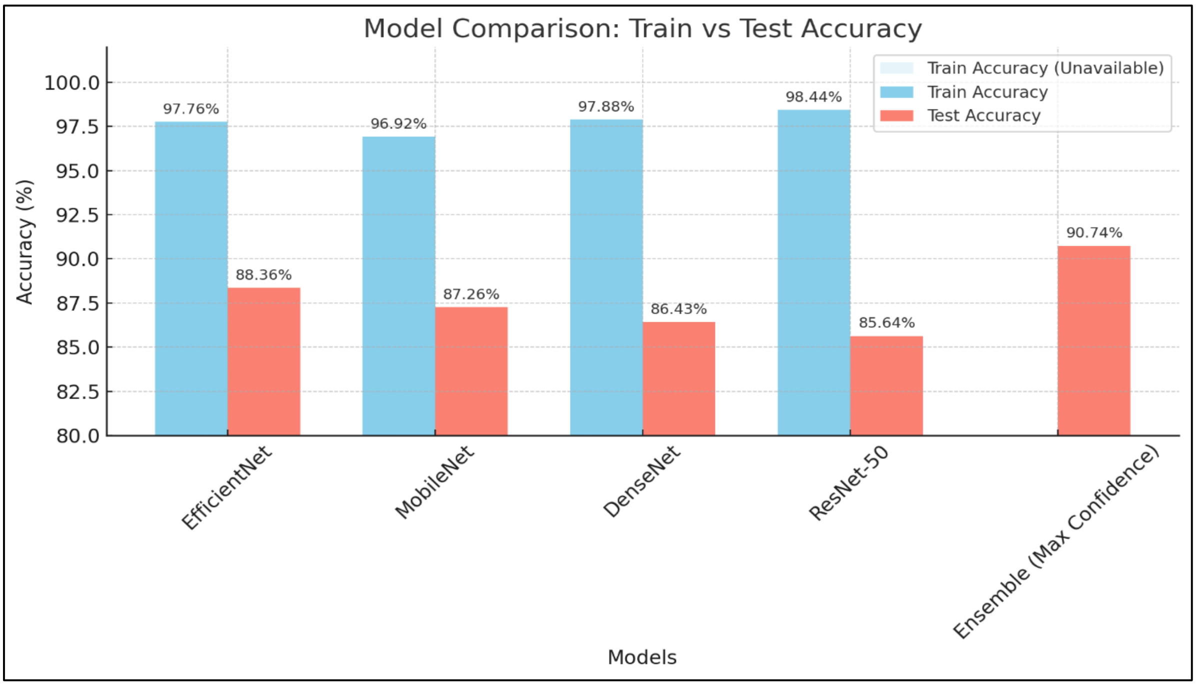

FIGURE 5.1: MODELS PERFORMANCE COMPARISION

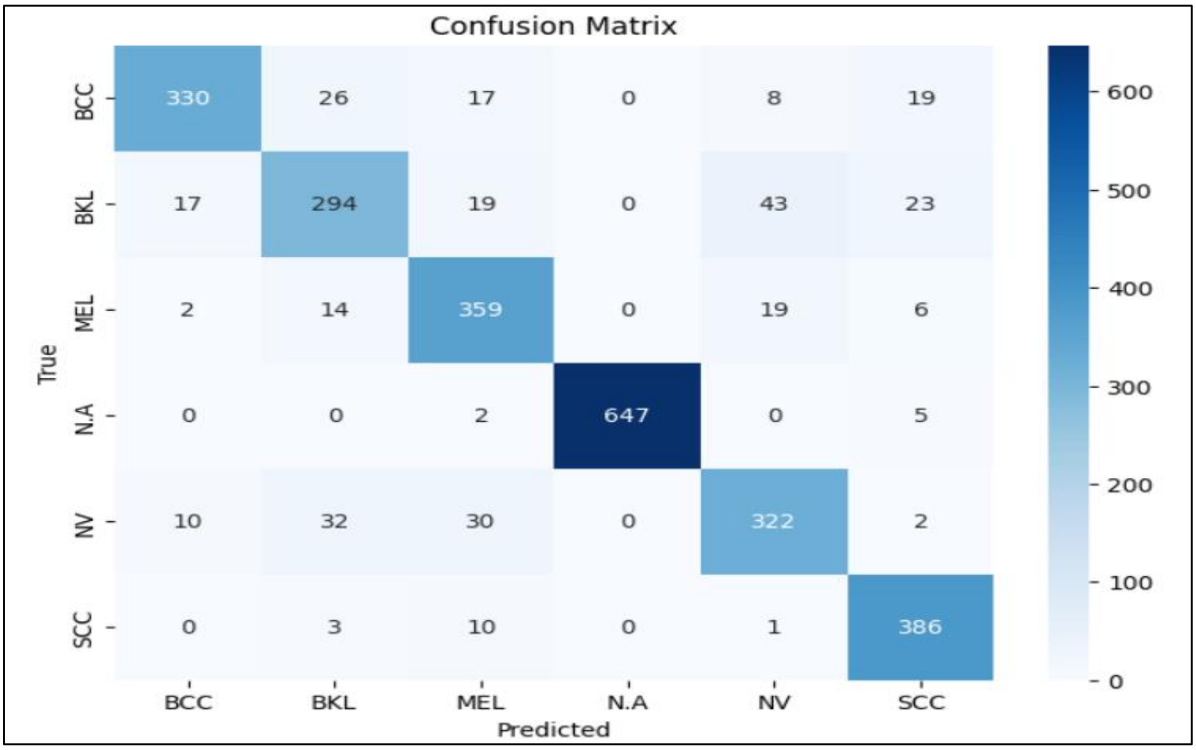

FIGURE 5.2: CONFUSION MATRIX FOR EFFICIENT NET

Classification Report:

	precision	recall	f1-score	support
BCC	0.92	0.82	0.87	400
BKL	0.80	0.74	0.77	396
MEL	0.82	0.90	0.86	400
N.A	1.00	0.99	0.99	654
NV	0.82	0.81	0.82	396
SCC	0.88	0.96	0.92	400
accuracy			0.88	2646
macro avg	0.87	0.87	0.87	2646
weighted avg	0.88	0.88	0.88	2646

Test Accuracy: 0.8836
Total Training Time: 961.86 seconds

FIGURE 5.3: CLASSIFICATION REPORT FOR EFFICIENT NET

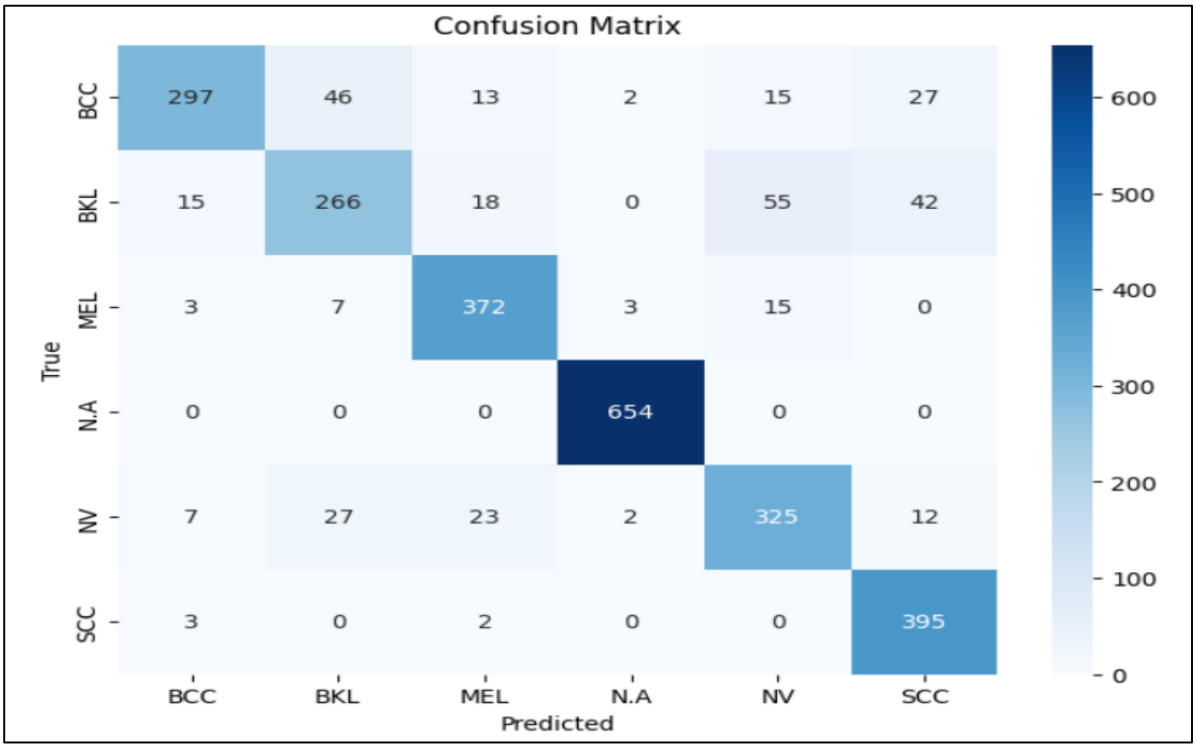

FIGURE 5.4: CONFUSION MATRIX FOR MOBILE NET

```
Classification Report:
              precision    recall  f1-score   support

         BCC       0.91      0.74      0.82       400
         BKL       0.77      0.67      0.72       396
         MEL       0.87      0.93      0.90       400
         N.A       0.99      1.00      0.99       654
          NV       0.79      0.82      0.81       396
         SCC       0.83      0.99      0.90       400

    accuracy                           0.87      2646
   macro avg       0.86      0.86      0.86      2646
weighted avg       0.87      0.87      0.87      2646

Test Accuracy: 0.8726
Total Training Time: 1754.55 seconds
```

FIGURE 5.5: CLASSIFICATION REPORT FOR MOBILE NET

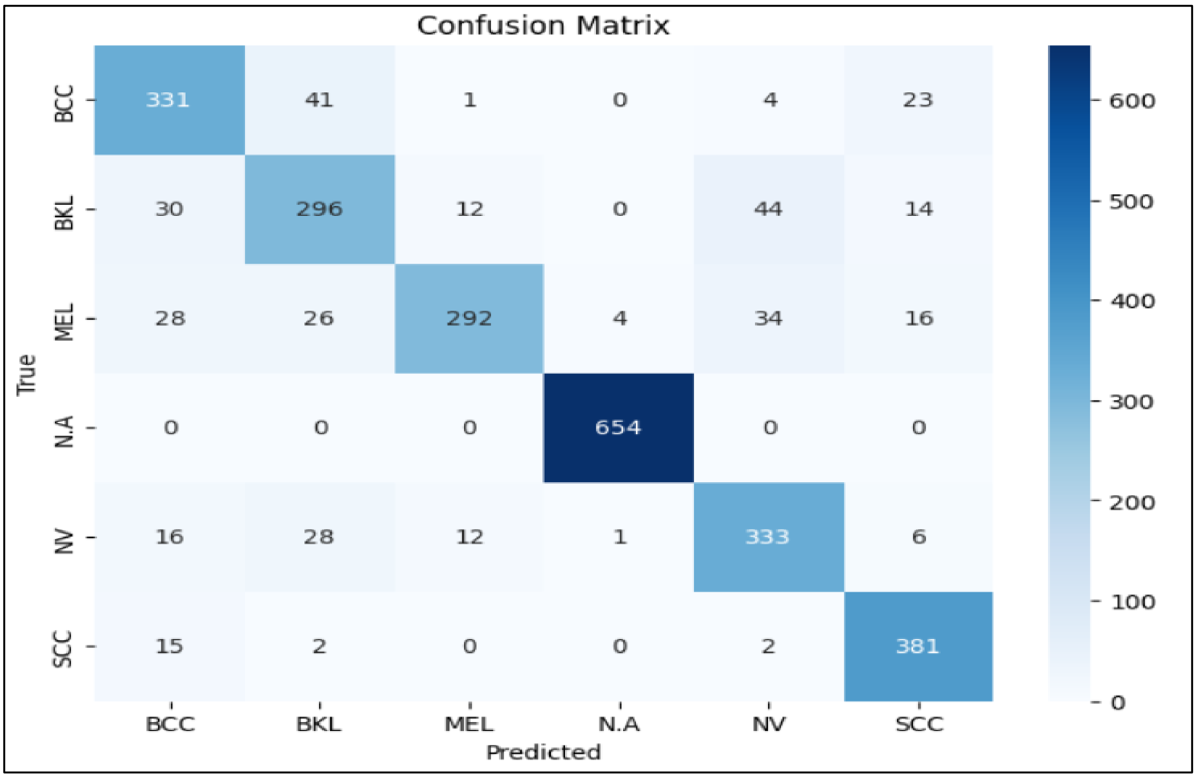

FIGURE 5.6: CONFUSION MATRIX FOR DENSENET

```
Classification Report:
              precision    recall  f1-score   support

         BCC       0.79      0.83      0.81       400
         BKL       0.75      0.75      0.75       396
         MEL       0.92      0.73      0.81       400
         N.A       0.99      1.00      1.00       654
          NV       0.80      0.84      0.82       396
         SCC       0.87      0.95      0.91       400

    accuracy                           0.86      2646
   macro avg       0.85      0.85      0.85      2646
weighted avg       0.87      0.86      0.86      2646

Test Accuracy: 0.8643
Total Training Time: 1174.73 seconds
```

FIGURE 5.7: CLASSIFICATION REPORT FOR DENSENET

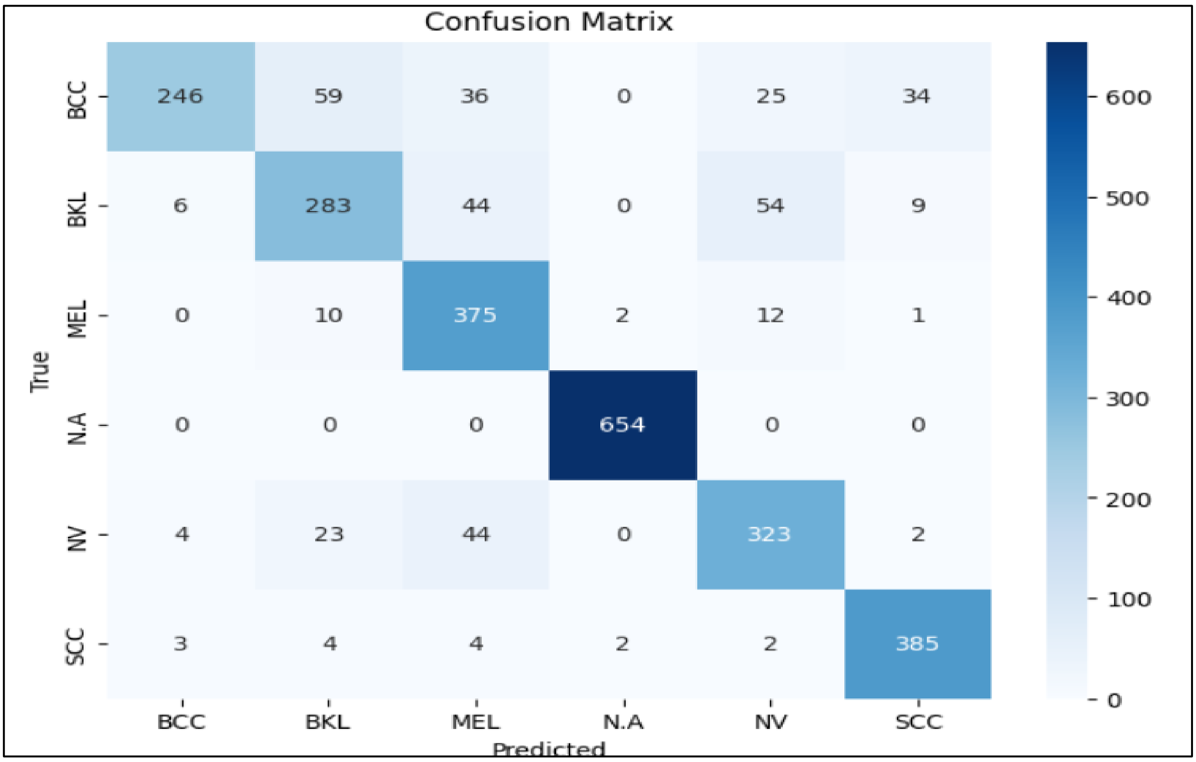

FIGURE 5.8 CONFUSION MATRIX FOR RESNET 50

Classification Report:

	precision	recall	f1-score	support
BCC	0.95	0.61	0.75	400
BKL	0.75	0.71	0.73	396
MEL	0.75	0.94	0.83	400
N.A	0.99	1.00	1.00	654
NV	0.78	0.82	0.80	396
SCC	0.89	0.96	0.93	400
accuracy			0.86	2646
macro avg	0.85	0.84	0.84	2646
weighted avg	0.86	0.86	0.85	2646

Test Accuracy: 0.8564
Total Training Time: 1171.21 seconds

FIGURE 5.9: CLASSIFICATION REPORT FOR RESNET 50

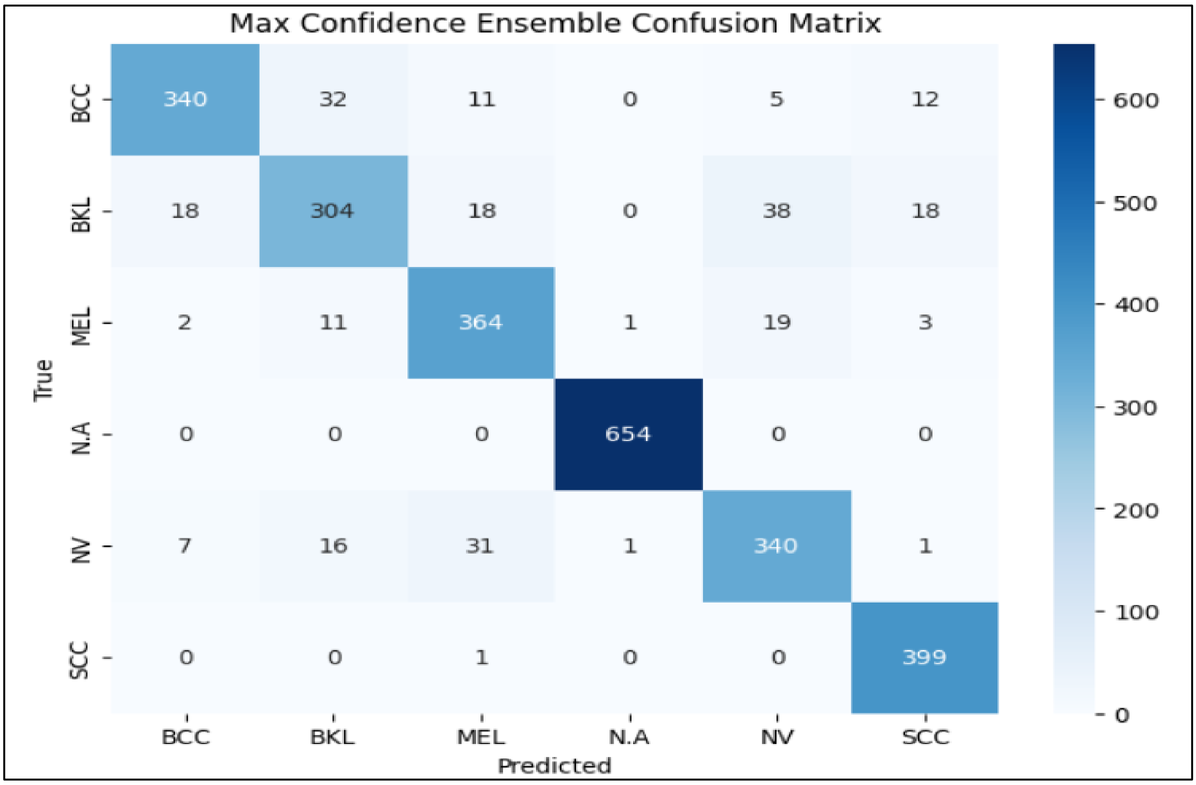

FIGURE 5.10: CONFUSION MATRIX FOR ENSEBLE MODEL

```
Classification Report:
              precision    recall  f1-score   support

         BCC       0.93      0.85      0.89       400
         BKL       0.84      0.77      0.80       396
         MEL       0.86      0.91      0.88       400
         N.A       1.00      1.00      1.00       654
          NV       0.85      0.86      0.85       396
         SCC       0.92      1.00      0.96       400

    accuracy                           0.91      2646
   macro avg       0.90      0.90      0.90      2646
weighted avg       0.91      0.91      0.91      2646

Max Confidence Ensemble Test Accuracy: 0.9074
```

FIGURE 5.11: CLASSIFICATION REPORT FOR ENSEMBLE MODEL

TABLE 5.1: MODELS RESULT COMPARISION

MODEL	**TRAIN ACCURACY**	**TEST ACCURACY**	**PRECISION**	**RECALL**
Efficient Net	97.76 %	88.36%	88%	88%
Mobile Net	96.92%	87.26%	87%	87%
Dense Net	97.88%	86.43%	87%	86%
Resnet 50	98.44%	85.64%	86%	86%
Ensemble model	N. A	90.74%	91%	91%

5.2 COMPARISION WITH EXISTING SOLUTION

The proposed models, including Efficient Net, MobileNet, Dense Net, and ResNet-50, were evaluated and compared based on their training and testing accuracies. Among these, the ensemble model using maximum confidence achieved the highest testing accuracy of 90.74%, outperforming all individual models. This indicates that combining multiple models leads to better generalization and robustness.

Compared to existing solutions reported in prior studies (or baseline models), which typically achieve testing accuracies in the range of 85–88%, our ensemble approach demonstrates a significant improvement in performance.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The project outlines a comprehensive process of developing AI models to address deep medical issues. Emphasis is placed on planning and process such as preparation, augmentation, and splitting to maximize input data for state-of-the-art machine learning, when attempting to ensure that high-performing AI technology is developed in the field. In training, the project achieved remarkable training accuracies of 97.76%, 96.92%, 97.88%, and 98.44% for Efficient Net, Mobile Net, Dense Net, and ResNet50, respectively. The project also achieved remarkable testing accuracies of 88.36%, 87.26%, 86.43% and 85.6%, respectively, for the same models. The capability of the system in detecting diseases such as melanoma and basal cell carcinoma is also demonstrated further by the usage of an ensemble method, which is supports confidence of 90.74%.

After the detection of the disease the project provides information about the detected disease by using generative AI of Google Gemini through API. Information-like symptoms, medication, overview, etc. are provided in a detailed structure which is easy to understand. A chat box is also provided which also uses API to respond to the queries asked by the user.

6.1.1 LIMITATIONS

There are few limitations that can affect the wide scale acceptance of the project:

- **Risk of overfitting:** Models give high training accuracies which is a indication that when provided with real world data may not be able to classify disease properly.

- **Rules and Regulations:** To provide medical advice you have to be qualified and need to have government approval.

6.2 FUTURE SCOPE

1. Currently model predicts 5 skin disease which can be expanded in future with support for more skin diseases.
2. A mobile application can be developed in future.
3. Instead of using google Gemini for information generation we can develop our own disease information generation model.

REFERENCES

[1] R. Pandey, H. Waghela, S. Rakshit, A. Rangari, A. Singh, R. Kumar, R. Ghosal, and J. Sen, "Generative AI-based text generation methods using pre-trained GPT-2 model," *arXiv preprint arXiv:2404.01786*, 2024.

[2] D. K. Saha, A. M. Joy, and A. Majumder, "YoTransViT: A transformer and CNN method for predicting and classifying skin diseases using segmentation techniques," *Informatics in Medicine Unlocked*, vol. 47, p. 101495, 2024.

[3] A. Escalé-Besa, J. Vidal-Alaball, Q. Miró Catalina, V. H. G. Gracia, F. X. Marin-Gomez, and A. Fuster-Casanovas, "The use of artificial intelligence for skin disease diagnosis in primary care settings: A systematic review," in *Healthcare*, vol. 12, no. 12, p. 1192, 2024.

[4] R. Agarwal and D. Godavarthi, "Skin disease classification using CNN algorithms," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 9, no. 1, 2023.

[5] M. Saeed, A. Naseer, H. Masood, S. U. Rehman, and V. Gruhn, "The power of generative AI to augment for enhanced skin cancer classification: A deep learning approach," *IEEE Access*, vol. 11, pp. 130330–130344, 2023.

[6] M. A. Hamim, S. H. Sajim, F. U. Rahman, and F. M. Tanmoy, "Multiple skin-disease classification based on machine vision using transfer learning approach," in *Proc. 14th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–8.

[7] K. Vayadande, A. A. Bhosle, R. G. Pawar, D. J. Joshi, P. A. Bailke, and O. Lohade, "Innovative approaches for skin disease identification in machine learning: A comprehensive study," *Oral Oncology Reports*, vol. 10, p. 100365, 2024.

[8] Y. Nie, P. Sommella, M. Carratù, M. O'Nils, and J. Lundgren, "A deep CNN transformer hybrid model for skin lesion classification of dermoscopic images using focal loss," *Diagnostics*, vol. 13, no. 1, p. 72, 2022.

[9] S. Inthiyaz, B. R. Altahan, S. H. Ahammad, V. Rajesh, R. R. Kalangi, L. K. Smirani, M. A. Hossain, and A. N. Z. Rashed, "Skin disease detection using deep learning," *Advances in Engineering Software*, vol. 175, p. 103361, 2023.

[10] M. W. P. Maduranga and D. Nandasena, "Mobile-based skin disease diagnosis system using convolutional neural networks (CNN)," *International Journal of Image, Graphics and Signal Processing*, vol. 12, no. 3, p. 47, 2022.

[11] M. Ahammed, M. Al Mamun, and M. S. Uddin, "A machine learning approach for skin disease detection and classification using image segmentation," *Healthcare Analytics*, vol. 2, p. 100122, 2022.

[12] M. Leite, W. D. Parreira, A. M. da R. Fernandes, and V. R. Q. Leithardt, "Image segmentation for human skin detection," *Applied Sciences*, vol. 12, no. 23, p. 12140, 2022.

[13] D. N. Vasundara, S. Naini, N. V. Sailaja, and S. Yeruva, "Classification of skin diseases using ensemble method," in *Proc. 2nd Int. Conf. on Advances in Computer Engineering and Communication Systems (ICACECS 2021)*, 2022, pp. 79–87.

[14] R. Mamdouh, N. El-Khamisy, K. Amer, A. Riad, and H. M. El-Bakry, "A new model for image segmentation based on deep learning," *International Journal of Online & Biomedical Engineering*, vol. 17, no. 7, 2021.

[15] S. S. Mohammed and J. M. Al-Tuwaijari, "Skin disease classification system based on machine learning technique: A survey," in IOP Conf. Ser.: Mater. Sci. Eng., vol. 1076, no. 1, p. 012045, 2021.

[16] I. R. I. Haque and J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informatics in Medicine Unlocked*, vol. 18, p. 100297, 2020.

[17] S. Bhadula, S. Sharma, P. Juyal, and C. Kulshrestha, "Machine learning algorithms based skin disease detection," *Int. J. Innov. Technol. Explor. Eng. (IJITEE)*, vol. 9, no. 2, pp. 4044–4049, 2019.

[18] R. D. Seeja and A. Suresh, "Deep learning based skin lesion segmentation and classification of melanoma using support vector machine (SVM)," *Asian Pac. J. Cancer Prev.*, vol. 20, no. 5, p. 1555, 2019.

[19] N. S. A. ALKolifi, "A method of skin disease detection using image processing and machine learning," *Procedia Comput. Sci.*, vol. 163, pp. 85–92, 2019.

[20] J. Velasco, C. Pascion, J. W. Alberio, J. Apuang, J. S. Cruz, M. A. Gomez, B. Molina Jr., L. Tuala, A. Thio-ac, and R. Jorda Jr., "A smartphone-based skin disease classification using mobilenet CNN," *arXiv preprint arXiv:1911.07929*, 2019.

Dhruvmajor2

ORIGINALITY REPORT

14%	10%	10%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hertfordshire Student Paper	2%
2	www.mdpi.com Internet Source	1%
3	Submitted to Kingston University Student Paper	1%
4	Mostafiz Ahammed, Md. Al Mamun, Mohammad Shorif Uddin. "A machine learning approach for skin disease detection and classification using image segmentation", Healthcare Analytics, 2022 Publication	1%
5	harvest.usask.ca Internet Source	1%
6	researchr.org Internet Source	<1%
7	Submitted to Monash University Student Paper	<1%
8	Submitted to University of Tennessee Knoxville Student Paper	<1%
9	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
10	Pamela Hermosilla, Ricardo Soto, Emanuel Vega, Cristian Suazo, Jefté Ponce. "Skin Cancer Detection and Classification Using Neural	<1%

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.