A Project Report on

Skin Disease Prediction using Deep Learning

Submitted in partial fulfillment for award of

Bachelor of Technology

Degree

in

Computer Science and Engineering

By

G. Sravani (Y21ACS458)

J. Manmadha (Y21ACS462)

B. Sasidhar (Y21ACS425)

K. Prasanth (L21ACS415)

Under the guidance of

K. Deepika, M.Tech

Assistant Professor

Department of Computer Science and Engineering

Bapatla Engineering College

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522 102, Andhra Pradesh, INDIA

2024-2025

Department of
Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled **Skin Disease Prediction using Deep Learning** that is being submitted by G. Sravani (Y21ACS458), J. Manmadha (Y21ACS462), B. Sasidhar (Y21ACS425), K. Prasanth (L21ACS415) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

Signature of the Guide	**Signature of the HOD**
K. Deepika	**Dr. M. Rajesh Babu**
Assistant Professor	**Assoc. Prof. & Head**

DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

G. Sravani (Y21ACS458)

J. Manmadha (Y21ACS462)

B. Sasidhar (Y21ACS425)

K. Prasanth (L21ACS415)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **K. Deepika**, Assistant Professor, Department of CSE, for her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Assoc. Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. N. Rama Devi** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. P. Pardhasaradhi,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

G. Sravani (Y21ACS458)

J. Manmadha (Y21ACS462)

B. Sasidhar (Y21ACS425)

K. Prasanth (L21ACS415)

Table of Contents

List of Figures

List of Tables

Abstract

Skin diseases are among the most common health issues affecting people of all ages globally. Early and accurate diagnosis is crucial for effective treatment, but traditional methods require expert dermatologists, clinical tests, and considerable time, making early detection difficult, especially in remote areas.

With the rise of Artificial Intelligence (AI) and Deep Learning (DL), automated disease detection has become highly accurate and efficient. This project proposes a web-based Skin Disease Prediction System using deep learning techniques for faster and precise identification of skin diseases.

A Convolutional Neural Network (CNN) model is developed and trained on a large dataset of 25,331 labeled images covering various skin conditions like melanoma, vascular lesions, basal cell carcinoma, and more. The model achieved 96% accuracy on both training and testing data, ensuring robust performance.

A user-friendly web interface is implemented using the Django framework, allowing users to upload skin images and receive instant predictions with probability scores. The system also provides an option to book a dermatologist appointment for further consultation.

This real-time prediction system significantly reduces diagnostic time, improves accessibility in rural areas, and serves as a valuable tool for medical professionals and students. The proposed system offers a reliable, scalable, and efficient solution for modern dermatological disease detection.

Keywords: Skin Disease Prediction, Deep Learning, Convolutional Neural Networks, Dermatological Diagnosis, Skin lesion classification, Web-based System

1 Introduction

Skin diseases are one of the most common health issues faced by people globally. Early and accurate detection of skin diseases is essential for timely treatment and better patient care. However, traditional diagnosis methods require expert dermatologists, which may not always be available, especially in rural and remote areas. With the rapid advancement in Artificial Intelligence (AI) and Deep Learning (DL), automated skin disease detection systems have emerged as an effective solution for overcoming these challenges.

1.1 Background

Skin diseases are a significant public health concern, affecting millions of people globally across all age groups. These diseases range from mild infections like acne and rashes to severe and life-threatening conditions like melanoma and skin cancer. The human skin is the largest organ in the body and is constantly exposed to environmental factors such as pollution, UV radiation, and bacteria, making it highly vulnerable to various diseases.

Traditional diagnosis of skin diseases involves physical examination by dermatologists, dermoscopic analysis, clinical tests, and sometimes skin biopsy. However, these conventional methods have several limitations such as the requirement of expert knowledge, time consumption, high cost, and limited availability of dermatologists in remote or rural areas.

The emergence of Artificial Intelligence (AI) and Deep Learning (DL) in the medical field has opened new possibilities for automatic disease detection and classification. Particularly, Convolutional Neural Networks (CNN) have shown tremendous success in analyzing and classifying medical images with high accuracy. Integrating AI models with a user-friendly web application enables users to easily upload skin images and get instant disease prediction results.

This project aims to develop an AI-based skin disease prediction system that can classify multiple types of skin diseases efficiently using deep learning models and provide users with accurate results through a web-based interface. This not only helps in early detection and treatment but also reduces the dependency on dermatologists for initial diagnosis, especially in healthcare-scarce regions.

1.2 Problem Statement

Skin diseases are among the most common health issues worldwide, affecting millions of individuals regardless of age or gender. The early and accurate diagnosis of skin diseases plays a crucial role in ensuring effective treatment and preventing the progression of the disease. However, manual diagnosis largely depends on the availability of dermatology experts, clinical tests, and biopsy reports, which may not always be accessible, especially in rural or underdeveloped regions.

Manual diagnosis is often time-consuming, expensive, and prone to human errors due to factors like limited experience, fatigue, and visual similarity between different skin conditions. Moreover, the increasing number of skin disease cases and the shortage of specialized dermatologists further widen the gap between patients and timely treatment.

In many cases, misdiagnosis or delayed diagnosis may result in serious complications, permanent skin damage, or even life-threatening conditions like skin cancer. Therefore, there is a critical need for an automated, efficient, and accurate skin disease prediction system that can assist both patients and healthcare professionals in identifying skin diseases at an early stage.

This project aims to develop a deep learning-based skin disease prediction system that uses image classification techniques to detect multiple skin diseases accurately. The system will also provide a user-friendly web interface to make the solution accessible to everyone, thereby improving healthcare availability and supporting early diagnosis and treatment.

1.3 Motivation

The increasing number of skin disease cases worldwide and the shortage of dermatologists have created a need for an automated diagnosis system. Manual diagnosis is time-consuming, costly, and sometimes inaccurate due to human errors. Early detection and accurate prediction of skin diseases can help prevent severe health issues. With advancements in deep learning and image processing, it is now possible to build intelligent systems that can assist in disease detection. This project is motivated to develop a skin disease prediction system that provides faster, accurate, and accessible diagnosis to users, especially in remote areas where medical resources are limited.

1.3.1 Increasing Cases of Skin Diseases

Skin diseases are becoming increasingly common across the world due to factors like rising pollution levels, changing climatic conditions, poor hygiene, unhealthy lifestyle habits, and prolonged exposure to harmful UV radiation. The use of chemical-based products, stress, and dietary habits also contribute to skin problems. According to recent studies, millions of people suffer from different types of skin diseases every year, ranging from minor infections to life-threatening conditions like skin cancer. This growing trend highlights the need for advanced and efficient diagnostic systems for early detection and prevention.

1.3.2 Challenges in Manual Diagnosis

Manual diagnosis of skin diseases mainly depends on the experience, knowledge, and availability of dermatologists. In rural or remote areas, access to specialized skin experts is often limited, leading to delayed diagnosis and treatment. Additionally, certain skin diseases exhibit similar visual symptoms, making it difficult for even experienced dermatologists to differentiate them accurately. Human errors, fatigue, misinterpretation of symptoms, and the subjective nature of visual examination further increase the chances of incorrect diagnosis. Moreover, manual diagnosis is time-consuming and costly, making it difficult to provide timely treatment to all patients.

1.3.3 Need for Early Detection and Treatment

Early detection of skin diseases plays a crucial role in preventing disease progression and reducing the risk of complications. Many skin conditions, if identified at an initial stage, can be treated effectively with minimal medication or simple procedures. However, late diagnosis may lead to severe health issues like skin cancer, permanent skin damage, or the disease spreading to other body parts. Timely identification not only improves the success rate of treatment but also reduces treatment cost, duration, and patient discomfort. Therefore, developing an automated system for early detection of skin diseases is essential to ensure faster diagnosis, better treatment planning, and improved patient outcomes.

1.3.4 Role of Deep Learning in Medical Field

Deep learning, especially Convolutional Neural Networks (CNN), has transformed the healthcare industry by providing accurate and automated solutions for medical image analysis. These models can efficiently learn complex patterns, textures, and features from large medical datasets, making them highly effective in disease classification tasks. In the field of dermatology, deep learning models help in analyzing skin lesion images and predicting skin diseases with greater accuracy, consistency, and speed compared to traditional methods. This reduces the dependency on manual observation and supports healthcare professionals in making better diagnostic decisions.

1.3.5 Improving Healthcare Accessibility

Many rural and remote areas lack access to specialized dermatologists, leading to delayed or incorrect diagnosis of skin diseases. AI-powered automated skin disease detection systems can bridge this healthcare gap by providing instant, reliable, and cost-effective diagnostic support. Patients can easily upload their skin images using mobile or web applications and receive quick predictions without visiting hospitals. This technology helps in early disease identification, timely treatment, and improves healthcare accessibility for underprivileged regions, ultimately saving lives and reducing the burden on healthcare infrastructure.

1.4 Objective

The main objective of this project is to develop an intelligent and automated Skin Disease Prediction System using Deep Learning techniques. The system is designed to classify multiple types of skin diseases accurately from skin lesion images.

It aims to overcome the limitations of traditional diagnosis by providing fast, reliable, and cost-effective solutions.

The specific objectives of this project are:

i. To design and develop a Convolutional Neural Network (CNN) model for accurate skin disease classification.

ii. To build a web-based platform using Django framework for real-time skin disease prediction.

iii. To provide a user-friendly interface where users can upload skin images for instant disease prediction.

iv. To display prediction results along with the probability/confidence score for user clarity.

v. To offer an option for patients to book an appointment with a dermatologist for further medical assistance.

vi. To improve the accessibility of dermatological healthcare services, especially in remote and underdeveloped areas.

vii. To reduce the time, effort, and cost involved in manual skin disease diagnosis.

viii. To create an efficient tool that can assist both healthcare professionals and patients in early disease detection and treatment planning.

1.5 Significance

This project holds great significance as it provides a smart and efficient solution for skin disease prediction using deep learning technology. It helps in early diagnosis, reduces healthcare gaps, and ensures better patient care with quick and reliable results.

i. Provides fast, accurate, and automated detection of various skin diseases.

ii. Helps in early-stage diagnosis, reducing the risk of severe complications.

iii. Increases accessibility of healthcare services for rural and remote areas.

iv. Reduces the workload of dermatologists by assisting in initial screening.

v. Minimizes human errors associated with manual diagnosis.

vi. Provides an easy-to-use web platform for both patients and healthcare professionals.

vii. Encourages technology integration in the medical field for better healthcare delivery.

viii. Acts as a learning tool for medical students in dermatology studies.

ix. Saves time, cost, and effort required for traditional clinical testing.

1.6 Existing System

In the existing system, Machine Learning (ML) techniques are utilized for the prediction and classification of skin diseases based on skin images. The system involves several ML algorithms such as Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) for classification purposes.

Firstly, the system takes input images of different skin diseases collected from standard datasets. These images undergo pre-processing techniques to enhance image quality, remove noise, and resize the images for better analysis. The next step involves feature extraction, where important features and patterns from the pre-processed images are identified and extracted.

These extracted features are then used to train various ML models for skin disease classification. The performance of these ML models is evaluated based on their accuracy. Among all the models used in the existing system, the Support Vector Machine (SVM) model achieved the highest accuracy of 91.94%, indicating better performance compared to other models. Random Forest achieved an accuracy of 82.02%, while Naive Bayes showed the least accuracy of 50.89%.

Although the existing system provided good accuracy through SVM, it has certain limitations. The system heavily depends on manual feature extraction, which may lead to the loss of valuable information. Additionally, the performance of traditional ML models decreases when dealing with large, complex, and high-dimensional image data. There is also a lack of automated feature learning, making the system less adaptable to new and unseen data.

Thus, there is a need for advanced deep learning techniques to overcome these limitations and enhance the performance and accuracy of skin disease prediction systems.

2 Literature Review

Several researchers have contributed significantly to the field of skin disease prediction using machine learning and deep learning techniques. The following literature presents various methods and their outcomes in classifying and detecting skin diseases effectively.

In [1], Esteva et al. (2017) developed a deep learning algorithm to classify skin cancers into different categories, such as melanoma and non-melanoma skin malignancies, using a dataset of more than 129,000 clinical photographs. This study highlighted the capability of deep learning models to perform at a dermatologist-level accuracy. With a 72.1% classification accuracy, the system fared better than a panel of 21 board-certified dermatologists, proving the potential of AI in medical diagnosis.

In [2], Brinker et al. (2019) assessed the effectiveness of a smartphone application that uses machine learning techniques to detect skin lesions in a community-based environment. The application was trained on a dataset of more than 10,000 images and could identify malignant tumors with a sensitivity of 88.9% and a specificity of 86.1%. This research showed that AI tools could be integrated into mobile health platforms for real-time skin disease detection.

In [3], Yu et al. (2020) proposed a novel multi-scale multi-CNN fusion technique to enhance the classification performance of skin lesion images. The study revealed that image cropping provided better results than resizing during preprocessing. Their MSM-CNN method ranked second on the ISIC 2018 challenge test dataset, achieving a balanced multi-class accuracy of 86.2%. This technique effectively utilized multiple CNNs to capture different feature levels of skin images.

In [4], Kassani et al. (2019) conducted a comprehensive survey on the use of convolutional neural networks (CNN) in skin cancer classification. The study analyzed various CNN architectures such as AlexNet, VGG, ResNet, and InceptionNet, and evaluated their performance on different datasets. The survey concluded that CNN models achieved an average accuracy of 92.3% in detecting melanoma and non-melanoma skin cancers, emphasizing the strength of CNNs in feature extraction and classification.

In [5], Debelee (2023) reviewed the application of machine learning and deep learning techniques for skin lesion detection and classification. The review provided a detailed analysis of advanced models like CNN, ResNet, and EfficientNet. The study focused on performance metrics such as accuracy, sensitivity, and specificity, and discussed the strengths and limitations of different machine learning models in dermatological diagnosis.

In [6], Liu et al. (2021) explored deep learning-based image segmentation and classification techniques for skin cancer detection. Their approach combined image segmentation to localize the region of interest with classification models for disease prediction. The model achieved an accuracy of 89.7% on the HAM10000 dataset, which is a popular benchmark dataset for skin lesion analysis.

In [7], Jain et al. (2022) introduced a hybrid model that combined CNN and transfer learning methods to improve skin disease classification accuracy. The model leveraged the pre-trained features of deep learning models with customized layers to better classify skin diseases. Their approach reached an overall accuracy of 91.2%, proving that hybrid models can significantly enhance performance.

In [8], Sharma et al. (2023) developed an ensemble learning approach integrating ResNet and EfficientNet for diagnosing melanoma and other skin lesions. Ensemble methods combine multiple models to produce better results than individual models. Their model achieved an impressive accuracy of 93.5% on real-world dermatological image datasets, demonstrating robustness and high precision in skin disease classification.

In [9], Yu et al. (2020) again contributed by proposing a multi-scale multi-CNN fusion technique. This method was highly effective in improving classification results by utilizing different CNN models and focusing on cropped images rather than resized images. The MSM-CNN technique achieved a balanced multi-class accuracy of 86.2% on the ISIC 2018 dataset, highlighting the importance of multi-scale feature extraction in skin lesion classification.

3 Proposed System

The proposed model for skin disease prediction project is as shown in Figure 3.1. The skin disease prediction system follows a structured workflow consisting of three key phases. First, skin disease images are collected and preprocessed through resizing, normalization, and conversion into numerical arrays. To address class imbalance, data balancing techniques like RandomOverSampler are applied. Next, the dataset is split into training and testing datasets, ensuring proper model learning. The CNN model extracts essential features from the images and undergoes training using the processed dataset. Finally, the trained model classifies test images into different skin disease categories such as melanoma, vascular lesions, squamous cell carcinoma, dermatofibroma, nevus, benign keratosis, basal cell carcinoma, actinic keratosis and its performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix. This workflow ensures efficient processing, accurate classification, and reliable performance measurement for skin disease detection.

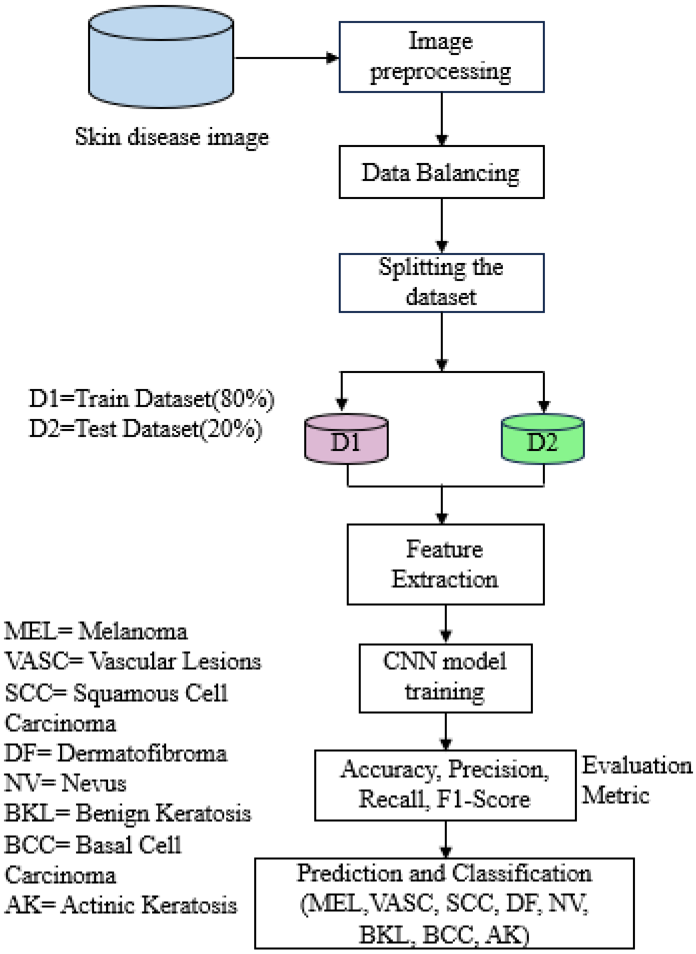

Figure 3.1 Workflow of Proposed System

3.1 Task

The main task of this project is to develop a deep learning-based model that can accurately detect and classify different types of skin diseases from dermoscopic images.

3.1.1 Project Planning and Setup

In this phase, the project objectives were clearly defined by deciding the problem statement as Skin Disease Prediction using Deep Learning. A suitable dataset, ISIC 2019 Skin Disease Dataset, was selected for model training and testing.

All the required Python libraries such as TensorFlow, Keras, Scikit-learn, Imbalanced-learn, Pandas, and Matplotlib were installed for smooth implementation of the project. The development environment was set up using Google Colab for easy execution of code with GPU support and better computational resources.

3.1.2 Data Collection and Preparation

In this phase, the ISIC 2019 Skin Disease Dataset was downloaded, which contains images of 8 different types of skin diseases such as Melanoma (MEL), Vascular Lesion (VASC), Squamous Cell Carcinoma (SCC), Dermatofibroma (DF), Melanocytic Nevus (NV), Benign Keratosis (BKL), Basal Cell Carcinoma (BCC), and Actinic Keratosis (AK).

To ensure uniformity in model training, all images were resized to a fixed dimension of (28x28x3*)*. The labels corresponding to each image were extracted from the dataset and encoded into numerical format for easier processing by the model. This step helped in preparing a clean and structured dataset suitable for deep learning model training and evaluation.

Table 3.1 Dataset Size

Skin Diseases	**Data Sample**
Actinic Keratosis (AK)	867
Basal Cell Carcinoma (BCC)	3323
Benign Keratosis like Lesions (BKL)	2624

Dermatofibroma (DF)	239
Melanoma (MEL)	4522
Nevus (NV)	12900
Squamous Cell Carcinoma (SCC)	628
Vascular Lesions (VASC)	253

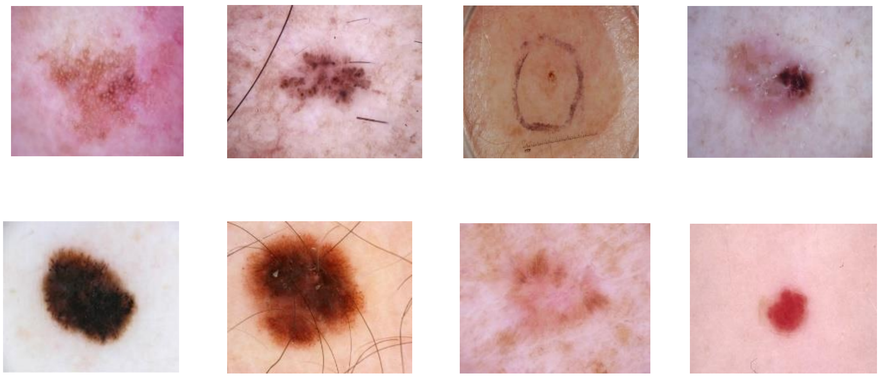

Figure 3.2 Sample images of skin diseases

3.1.3 Feature Extraction

In this phase, several preprocessing and feature extraction techniques were applied to prepare the image data and target labels in a format suitable for training the CNN model.

i. **Image Conversion and Resizing:** All the skin disease images from various classes were loaded and resized to a uniform dimension of (28x28x3) using the load_img() function from Keras. Resizing ensures that all input images are of the same size, which is essential for deep learning models. Subsequently, each image was converted into a numerical array using the img_to_array() function, which transforms image pixels into a format suitable for model input.

ii. **Label Encoding:** Since the target labels (skin disease names) were in string format, LabelEncoder() from sklearn was used to convert these categorical labels into numerical values.

iii. **One-Hot Encoding:** After label encoding, to_categorical() from TensorFlow was applied for one-hot encoding. This transformed the numerical class labels into a binary matrix representation, enabling the model to perform multi-class classification effectively.

iv. **Data Balancing using RandomOverSampler:** As the dataset was highly imbalanced, RandomOverSampler() from the imblearn library was utilized to balance the dataset by generating additional synthetic samples for minority classes. To apply this technique, image arrays were initially reshaped into a 2D structure.

v. **Reshaping data:** After oversampling, the balanced data was reshaped back to its original image dimensions of (28x28x3) to make it compatible with CNN model training.

3.1.4 Data Balancing

In this phase, data balancing techniques were applied to handle the issue of class imbalance present in the skin disease dataset. Imbalanced datasets can lead to biased model predictions, where the model performs well on majority classes but poorly on minority classes. To overcome this challenge, RandomOverSampler from the imblearn library was used.

i. **Handling Class Imbalance using RandomOverSampler:** The dataset contained an unequal number of images for different skin disease classes. To balance the dataset, RandomOverSampler() was applied, which generates additional synthetic samples of minority classes by randomly duplicating existing samples. This helps the model learn patterns from all classes equally.

ii. **Reshaping Data for Oversampling:** Since RandomOverSampler works on 2D tabular data, the image arrays were first reshaped from their original 3D shape (28x28x3) into a 2D format (flattened) to perform oversampling.

iii. **Restoring Original Image Shape:** After the oversampling process, the newly generated data was reshaped back into its original image dimensions (28x28x3) to make it suitable for CNN model training.

3.1.5 Model Selection and Development

In this phase, a Custom Convolutional Neural Network (CNN) model was designed and developed to perform multi-class classification of skin diseases, as CNN is highly effective in handling image data and automatically extracting essential features without manual intervention. The architecture of the proposed CNN model was structured with multiple layers to enhance accuracy and performance while minimizing overfitting. It consists of three Convolutional Layers for feature extraction, each followed by MaxPooling Layers to reduce dimensionality and computational complexity while preserving important features. Batch Normalization was applied to stabilize and speed up the training process, while Dropout Layers were incorporated to prevent overfitting by randomly deactivating neurons during training. Finally, the extracted features were passed through Fully Connected Dense Layers, allowing the model to perform accurate classification of skin disease images based on the learned features.

3.1.6 Integration and Deployment

In this phase, the trained and validated CNN model was integrated into a web-based application using the Django Web Framework. This optional yet crucial phase enables users to interact with the model in a real-time environment through a user-friendly interface.

The primary objective of this phase was to deploy the skin disease prediction model in such a way that end-users can easily upload their skin images and instantly receive predictions regarding the type of skin disease.

The Key Features of Integration and Deployment includes:

i. **Integration with Django Framework:** The complete deep learning model was integrated with Django, a high-level Python web framework, to develop a dynamic and interactive web application.

ii. **User Interface for Image Upload:** A simple and intuitive user interface was designed where users can upload their skin images directly through the web portal. Once the image is uploaded, it is preprocessed, and the model performs prediction in real-time.

iii. **Display of Prediction Results:** After processing the image, the predicted skin disease name along with its probability score is displayed to the user for better clarity and understanding.

iv. **Doctor Appointment Booking Feature:** An additional feature was provided for users to book a doctor's appointment based on the prediction result. This functionality helps users take immediate action and consult a dermatologist for further treatment.

3.1.7 Testing and Validation

In this phase, the selected Custom CNN model was trained and evaluated to measure its performance and accuracy. The model was compiled using the Adam optimizer, which is well-suited for faster and efficient convergence during training. The loss function used was Categorical Crossentropy, as the problem involved multi-class classification. Accuracy was considered as the primary evaluation metric to monitor the model's performance. To improve the training process and prevent overfitting, callbacks such as ReduceLROnPlateau (to reduce learning rate on performance stagnation) and EarlyStopping (to stop training when no further improvement was observed) were used. The model was trained and evaluated on both the train and test datasets to ensure consistency in performance. Further, various performance metrics like Accuracy, Confusion Matrix, and Classification Report were used to analyze the model's effectiveness, providing a detailed insight into the classification capability across all classes.

3.1.8 Documentation and Reporting

In this phase, the entire project development process was properly documented in a structured manner for better understanding and presentation. Detailed records of each step, including data preprocessing, model building, training, evaluation, and deployment, were maintained. Various graphical representations were generated to visualize the model's performance, such as accuracy and loss curves for both training and validation phases. Additionally, class distribution graphs were created to show the balance of data across different skin disease classes after applying data balancing techniques. Confusion matrix and classification report visualizations were also included to represent the model's performance clearly. This phase ensured that all the results, observations, and outputs were well-documented and ready for reporting and future reference.

3.2 Dataset

The dataset used for the proposed skin disease classification system is the ISIC 2019 Skin Lesion images for classification, which consists of 25,331 images categorized into 8 different skin disease classes. Table 1 shows the distribution of the collected dataset for this study. These images are high-resolution dermoscopic images collected from real-world clinical cases, making them well suited for deep learning-based skin disease diagnosis. Each image is labeled based on its corresponding skin condition, ensuring a structured dataset for training and evaluation. The 8 categories include Melanoma (MEL), Nevus (NV), Basal Cell Carcinoma (BCC), Actinic Keratosis (AK), Benign Keratosis (BKL), Squamous Cell Carcinoma (SCC), Vascular Lesions (VASC), and Dermatofibroma (DF). The dataset presents challenges such as class imbalance, where some categories contain significantly more images than others. To address this, data balancing techniques like oversampling are used. This dataset is a standard benchmark in medical AI research, enabling the development of robust, real-time skin disease classification models with high accuracy and reliability.

3.3 Input

In this project, the input provided to the system is a skin disease image uploaded by the user through the web interface. The uploaded image must be in a suitable format like JPEG. The image is then resized to a uniform dimension of (28x28x3) pixels to match the input requirements of the trained deep learning model. This resizing ensures consistency and proper compatibility with the model for accurate prediction of the skin disease.

3.4 Output

The output of the system provides the predicted skin disease class based on the input image uploaded by the user. Along with the predicted class, a probability score is displayed, indicating the confidence level of the prediction. Additionally, the accuracy score of the model's prediction is shown to enhance user trust. In the case of Django web application integration, the system also provides an optional feature allowing users to book a doctor's appointment directly based on the prediction result for further medical consultation.

4 Algorithms

The algorithm used in this project is a Custom Convolutional Neural Network (CNN) designed for multi-class skin disease classification. CNN efficiently extracts features from input images and classifies them into respective disease categories.

4.1 CNN Architecture

The Convolutional Neural Network (CNN) architecture is a deep sequential model designed to effectively extract spatial features from the resized 28×28 RGB images. The architecture begins with a Conv2D layer with 32 filters, a kernel size of 3×3, ReLU activation, and He-normal initialization, followed by MaxPooling2D and BatchNormalization to downsample and normalize the feature maps. This pattern is repeated with increasing filter sizes (64 and 128) across three convolutional blocks, allowing the model to learn more complex features at deeper levels. After flattening the output, a Dropout layer (rate = 0.5) is added to prevent overfitting. This is followed by three Dense layers with 256, 128, and 64 neurons respectively, each using ReLU activation and L2 regularization to control overfitting. BatchNormalization is applied after each dense layer to speed up training and stabilize the learning process. Finally, the model ends with a Dense output layer of 8 neurons and a softmax activation function, which enables multi-class classification of the skin diseases. The model is compiled using the Adam optimizer, categorical cross-entropy as the loss function, and accuracy as the evaluation metric, with training regulated by ReduceLROnPlateau and EarlyStopping callbacks for optimal performance.

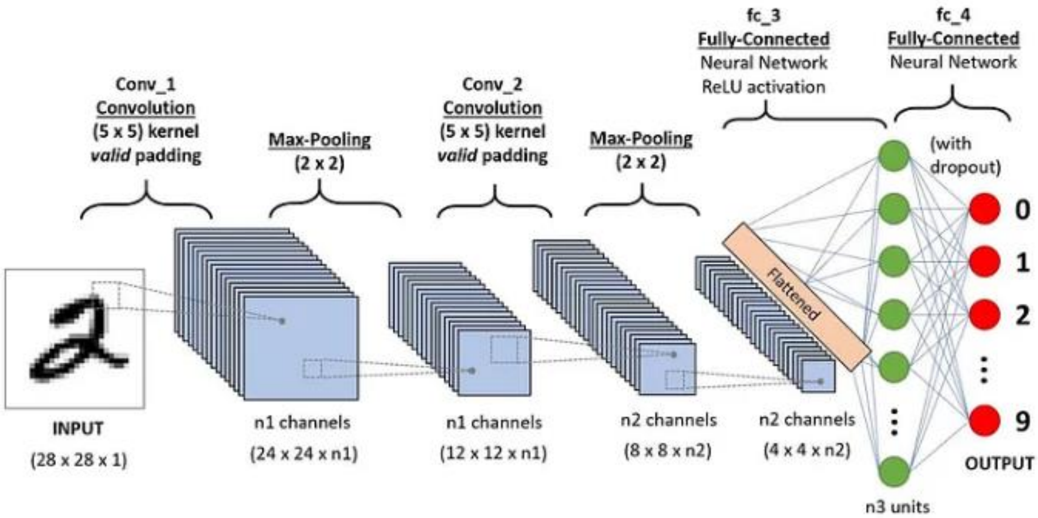

Figure 4.1 Architecture of CNN

4.1.1 Input Layer

The input layer accepts skin disease images of size 28x28x3, where 28x28 represents the image dimension and 3 represents the RGB color channels.

4.1.2 Convolutional Layers

Three Convolutional layers are used to extract important features like edges, textures, and patterns from the images using filters and kernels.

4.1.3 Max Pooling Layers

MaxPooling layers are applied after each Convolutional layer to reduce the spatial size of the feature maps and computational complexity while preserving the essential features.

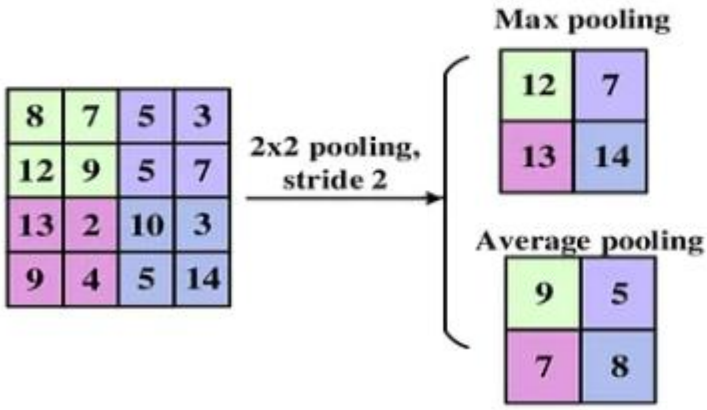

Figure 4.2 Comparison of Max Pooling and Average Pooling

4.1.4 Batch Normalization Layers

Batch Normalization is applied after Convolutional layers to normalize the activations and accelerate the training process by reducing internal covariate shift.

4.1.5 Dropout Layers

Dropout layers are added to prevent overfitting by randomly disabling a fraction of neurons during training, thus improving the model's generalization capability.

4.1.6 Flatten Layer

The Flatten layer converts the 2D feature maps obtained from the previous layers into a 1D feature vector for input into the Dense layers.

4.1.7 Fully Connected Dense Layers

Dense layers are used to perform classification based on the features extracted from the previous layers. These layers learn complex relationships between features.

4.1.8 Output Layer

The final output layer uses the Softmax activation function to classify the input image into one of the eight predefined skin disease classes by providing probability scores for each class.

4.2 Model Compilation

The proposed CNN model was compiled using the Adam Optimizer, which helps in efficient and faster convergence during training. The loss function used was Categorical Crossentropy, suitable for multi-class classification problems. Accuracy was selected as the primary evaluation metric to measure the model's performance.

4.3 Training Techniques Used

To improve the performance and efficiency of the model during training, two important techniques were applied. Early Stopping was used to prevent overfitting by stopping the training when the model's performance stopped improving on the validation set. ReduceLROnPlateau was used to automatically reduce the learning rate when the model's accuracy or loss stopped improving, helping the model to learn better and avoid getting stuck at local minima.

4.4 Model Evaluation

To assess the performance and effectiveness of the proposed CNN model, various evaluation techniques were implemented. These techniques helped in analyzing the accuracy of the model, understanding the error distribution, and visualizing the learning behavior during training and validation phases.

4.4.1 Accuracy Score

The accuracy score was calculated to measure the overall correctness of the model in classifying skin disease images. It indicates the percentage of correctly predicted instances out of the total predictions.

4.4.2 Confusion Matrix

A confusion matrix was generated to visualize the performance of the model in terms of correctly and incorrectly classified instances for each skin disease class. It provided detailed insights into class-wise prediction results and error distribution.

4.4.3 Classification Report

A detailed classification report was generated using sklearn, which provided key evaluation metrics such as Precision, Recall, F1-score, and Support for each class. These metrics helped in understanding the model's capability to handle class-wise imbalance and its performance across different classes.

4.4.4 Visualization of Training & Validation Accuracy/Loss Graphs

Graphs were plotted to visualize the training and validation accuracy and loss across each epoch. These visualizations helped in identifying overfitting or underfitting and understanding the model's learning behavior during the training phase.

5 System Design

The system flow diagram illustrates the overall working process of the Skin Disease Prediction System, starting from user authentication to appointment booking and feedback submission. Initially, users must authenticate themselves, either as a patient or doctor. If the user logs in as a patient, their profile is verified. New patients are prompted to complete their profile setup, while existing patients can directly access their dashboard. Patients can upload an image of the affected skin area, which is then processed using an AI-based Skin Disease Detection model. The system analyzes the image and displays the type of skin disease along with its severity level. Based on the results, the system recommends suitable dermatologists by fetching nearby specialists. Patients can choose a doctor and proceed to book an appointment by selecting a preferred time slot. Upon confirmation, appointment details are shared with the user. After consultation, patients can submit their feedback. On the other hand, doctors logging in can access their dashboard to update appointment slots, check patient details, and manage their schedule. This flow ensures an automated, smooth, and user-friendly interaction between patients and doctors within the platform.

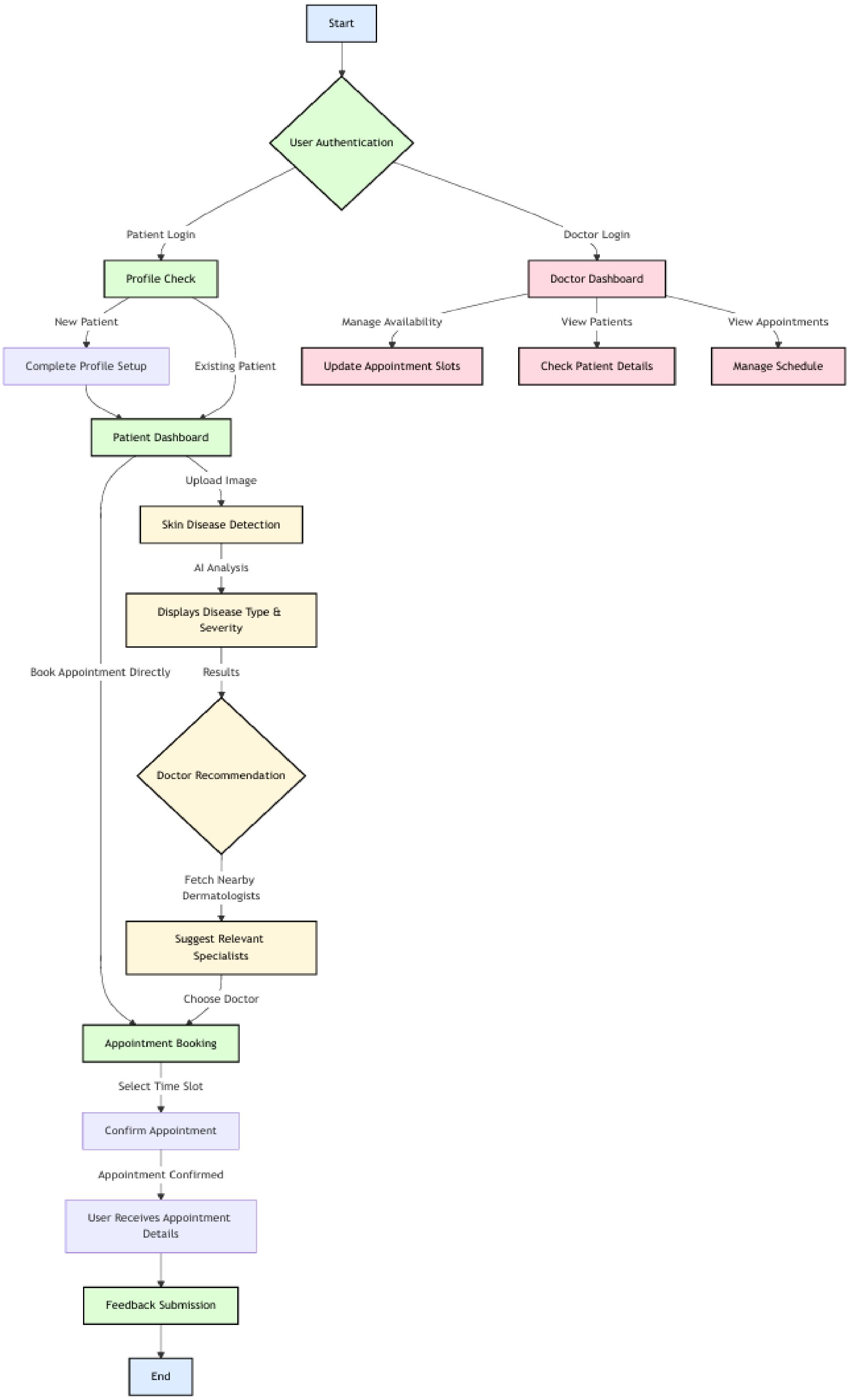

Figure 5.1 Flowchart of Skin Disease Prediction and Doctor Appointment Booking System

6 Implementation

In this project, a Convolutional Neural Network (CNN) model is implemented to predict skin diseases using image data. The dataset is pre-processed by resizing images, handling missing values, and balancing classes using RandomOverSampler. The model is trained using TensorFlow and Keras libraries with multiple convolutional, pooling, and dense layers. Finally, the trained model is evaluated using accuracy, precision, recall, F1-score, and confusion matrix.

6.1 Requirements

Requirements are critical to the success of this *Skin Disease Prediction* project. By meeting these requirements, we ensure that the software runs smoothly, processes large image datasets efficiently, and provides accurate disease predictions. Meeting these requirements also ensures that users can easily access and interact with the system for real-time skin disease detection. Ignoring these requirements may lead to slow processing, inaccurate predictions, errors, and system failures, which can negatively affect the success and usability of the project.

6.1.1 Hardware Requirements

Hardware requirements are essential to ensure that the project's software can run efficiently and effectively. These requirements include the minimum specifications for the CPU, RAM, storage, and other hardware components. By meeting these requirements, the project's software can run smoothly, reducing the risk of crashes, errors, and other issues, especially while handling large image datasets and running deep learning models.

i. Processor : Intel i5/i7 or higher

ii. RAM : Minimum 8 GB

iii. Free GPU Access : Recommended (for faster training)

6.1.2 Software Requirements

Software requirements are also critical to the success of a project. These requirements include the specific versions of software and operating systems that are compatible with the project's software. By ensuring that the project's software is compatible with the required software and operating systems, you can reduce the risk of compatibility issues and ensure that the project's software functions as intended for accurate skin disease prediction and real-time analysis.

i. Operating System : Windows / Linux / MacOS

ii. Python Version : 3.7 or above

iii. IDE: Jupyter Notebook or Google Colab (for model training with free GPU access) , VS Code

iv. Framework: Django

6.1.3 Libraries

Libraries play a crucial role in software projects by providing pre-written code and functionality that developers can leverage to expedite development, improve code quality, and enhance the capabilities of their applications.

In this project, several essential Python libraries are used to handle different tasks like data preprocessing, model building, training, evaluation, and visualization. Below is the importance of each library used in this project:

i. **Tensorflow:** TensorFlow is an open-source deep learning and machine learning library developed by Google. It is widely used for building and training neural networks, especially for tasks like image classification, object detection, and natural language processing. In this project, TensorFlow provides tools to build the Convolutional Neural Network (CNN) model and handle deep learning operations.

ii. **Keras:** Keras is a high-level deep learning API built on top of TensorFlow. It simplifies the process of building and training neural networks with its easy-to-use interface. In this project, Keras is used to design the CNN model architecture, preprocess images, and implement various layers like Conv2D, MaxPooling2D, Flatten, Dense, Dropout, and BatchNormalization.

iii. **NumPy:** NumPy (Numerical Python) is a fundamental library for numerical computations in Python. It provides support for multi-dimensional arrays, mathematical functions, and linear algebra operations. In this project, NumPy is used for handling image arrays, reshaping data, and performing array-based computations efficiently.

iv. **Pandas:** Pandas is an open-source library used for data manipulation and analysis. It provides powerful data structures like DataFrames to work with structured data easily. In this project, Pandas is used to load and explore the metadata (CSV files), which contains information about images and their corresponding labels.

v. **Scikit-learn:** Scikit-learn is a machine learning library in Python that provides simple and efficient tools for data analysis and modeling. In this project, Scikit-learn is used for Splitting data into train and test sets, Encoding categorical labels into numerical form, Evaluating the model using performance metrics like confusion matrix and classification report.

vi. **Seaborn**: Seaborn is a data visualization library based on Matplotlib. It provides an attractive and easy way to create informative statistical graphics. In this project, Seaborn is used to visualize data distribution and improve the appearance of plots.

vii. **Matplotlib:** Matplotlib is a powerful plotting library used to create static, animated, and interactive visualizations in Python. In this project, Matplotlib is used for plotting graphs, visualizing the confusion matrix, and displaying the model's performance visually.

viii. **Imbalanced-learn:** imbalanced-learn is a Python library used to deal with imbalanced datasets. It provides various techniques to balance the dataset like oversampling and undersampling. In this project, the RandomOverSampler technique from imblearn is used to handle class imbalance by generating more samples from minority classes to improve the model's performance.

6.2 Code

The code is implemented using a Convolutional Neural Network (CNN) model to predict different types of skin diseases from image data by processing, training, and classifying them into respective categories with high accuracy.

6.2.1 Importing all necessary packages

In this step, all the necessary Python libraries are imported which are essential for implementing the Skin Disease Prediction project. The os library is used for handling file paths, and warnings is used to ignore unwanted warning messages.

NumPy and Pandas are used for numerical operations and data handling, while Seaborn and Matplotlib are used for data visualization. Sklearn provides tools for splitting the dataset, preprocessing data, and evaluating model performance. Keras and TensorFlow libraries are used to build and train deep learning models with layers like Conv2D, MaxPooling, Dense, Dropout, etc. Regularization and callbacks like ReduceLROnPlateau and EarlyStopping help improve model performance and avoid overfitting. The imblearn library is used to handle data imbalance using Random Over Sampling technique, ensuring balanced training data for better model accuracy.

6.2.2 Loading Dataset

In this step, the dataset required for skin disease prediction is loaded using the Pandas library. The pd.read_csv() function is used to read the CSV file named ISIC_2019_Training_GroundTruth.csv from the specified location. This file contains the image IDs along with their corresponding disease labels. The print(data.shape) command is used to display the total number of rows and columns present in the dataset, which helps in understanding the size and structure of the data before proceeding with further analysis and model building.

6.2.3 Loading and Resizing Images

In this step, a function named load_and_resize_images() is created to load and resize all the images from the dataset. It takes the image paths and resizes each image to a fixed target size of 28x28 pixels using the load_img() function. The loaded images are then converted into numerical arrays using img_to_array() and stored in a list. Finally, all the images are converted into a NumPy array using np.array() for easy processing and are returned. This step ensures that all images are of the same size, which is essential for feeding them into a machine learning model.

6.2.4 Optimising Data

In this step, RandomOverSampler from the imbalanced-learn library is used to handle the problem of imbalanced data in the dataset. The RandomOverSampler() technique balances the dataset by randomly duplicating samples from the minority classes to match the number of samples in the majority class. Here, X_reshaped represents the input features (images), and y_one_hot represents the corresponding labels. After applying the fit_resample() function, the new balanced data is stored in X_resampled and y_resampled, ensuring that the model learns equally from all classes and improves the overall prediction accuracy.

6.2.5 Splitting Data and Training CNN Model

In this step, the resampled and balanced dataset obtained using RandomOverSampler is divided into training and testing sets using the train_test_split() function from Scikit-learn. 80% of the data is allocated for training the CNN model to learn the patterns and features of skin diseases, while the remaining 20% is reserved for testing the model's performance and evaluating its accuracy on unseen data. The random_state=42 is set to ensure reproducibility of results, meaning the data will always split the same way every time the code is run. This process helps in validating how well the model generalizes to new data.

6.2.6 Implementation of CNN layers

In this step, a Convolutional Neural Network (CNN) model is built using the Sequential API from Keras for skin disease prediction. The model consists of multiple layers like Conv2D for feature extraction, MaxPooling2D for reducing the spatial size, and BatchNormalization for stabilizing and speeding up the training process.

Regularization (l2) is used to prevent overfitting, and Dropout is applied to randomly drop neurons for better generalization. The Flatten layer converts the extracted features into a 1D vector, which is then passed to Dense layers for classification. The final Dense layer with softmax activation is used to classify the input images into 8 different skin disease categories.

6.2.7 Model Fitting and Training Process

After successfully designing and building the CNN model architecture, the next step is to compile the model by defining the optimizer, loss function, and evaluation metrics. In this project, the Adam Optimizer is used for model compilation as it provides adaptive learning rates and performs efficiently for most deep learning applications.

Since the proposed model deals with multi-class classification of various skin diseases, the Categorical Crossentropy loss function is employed to calculate the loss between the predicted and actual class labels.

The model is then trained using the fit() function for 70 epochs with a batch size of 64. To improve the model's generalization ability and avoid overfitting, 20% of the training data is reserved as validation data during training. The training process helps the model learn meaningful patterns from the input images and enhance its predictive performance.

6.2.8 Model Performance Analysis

To evaluate the performance of the proposed CNN model for skin disease classification, both training and validation loss and accuracy were visualized using line plots. The model was trained for several epochs, and the loss and accuracy values for both training and validation data were recorded. The loss curve helps to understand how well the model is learning, while the accuracy curve shows the improvement in prediction accuracy over time. Additionally, the best accuracy and lowest loss points were highlighted on the graph to indicate the model's optimal performance during training. After training, the model was evaluated on unseen test data using the evaluate() function, which provided the final test loss and test accuracy. Further, the model's prediction capability was analyzed using a classification report that presents precision, recall, F1-score, and support for each class. The confusion matrix was also plotted to visually represent the correct and incorrect predictions of each skin disease class. Finally, the trained model was saved in H5 format for future use or deployment.

7 Results

This section presents the experimental setup, the obtained results, and its discussion.

7.1 Model Performance Summary

The performance of the proposed Skin Disease Prediction model was evaluated based on the Training and Validation Loss and Accuracy graphs over 70 epochs. The model initially exhibited higher loss values, but with continuous training, both training and validation losses gradually decreased and stabilized, indicating effective learning and minimal overfitting. The accuracy of the model significantly improved over the epochs, reaching 96% for both training and validation at the 66th epoch, which was identified as the best epoch. The minimal gap between training and validation accuracy, along with the low loss values (~0.1), clearly indicates that the model has achieved excellent performance with effective generalization and reliable prediction capability on unseen data.

7.2 Graphical Representation of Model Performance

The final clean dataset was partitioned into two subsets, 80% for training and 20% for testing of the proposed model. The Figure 7.1 represents the training and validation accuracy of the skin disease classification model over 70 epochs. The red line shows the training accuracy, which steadily increases and stabilizes close to 96% after 40 epochs. The green line represents validation accuracy, which initially fluctuates but improves significantly, closely following the training accuracy. A blue dot marks the best epoch at 66, where the model achieved the highest validation accuracy before stabilizing. The overall trend indicates that the model is well-trained without significant overfitting, as both accuracies converge at around 96%. The initial fluctuations in validation accuracy suggest some variance, but the model becomes stable after sufficient training.

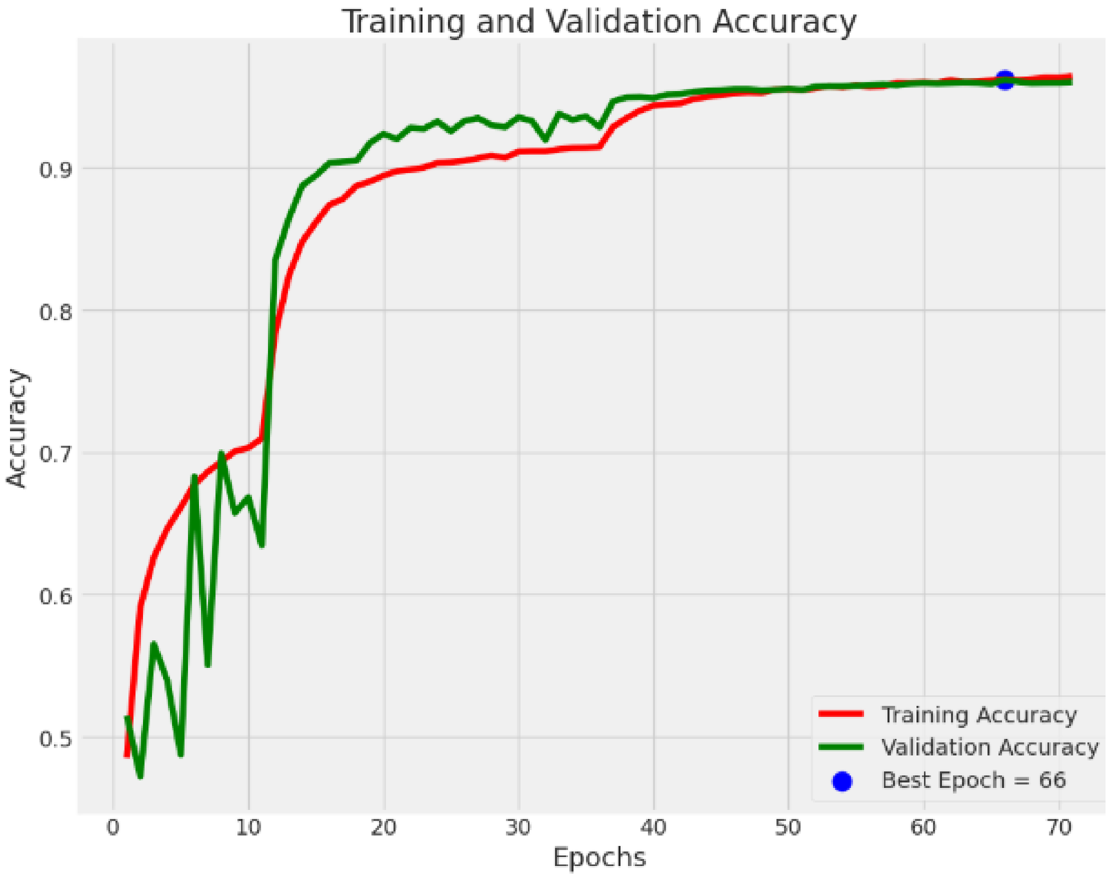

Figure 7.1 Training vs Validation Accuracy Graph

The Figure 7.2 titled Training and Validation Loss illustrates the loss values of the skin disease classification model over 70 epochs. The red line represents the training loss, while the green line represents the validation loss. Initially, both losses are high, but they rapidly decrease within the first 20 epochs, indicating effective learning. After 20 epochs, the loss continues to decrease gradually and stabilizes near zero as training progresses. The blue dot marks the best epoch at 66, where the model likely achieved optimal performance. The similarity between training and validation loss curves suggests minimal overfitting, indicating that the model generalizes well to unseen data.

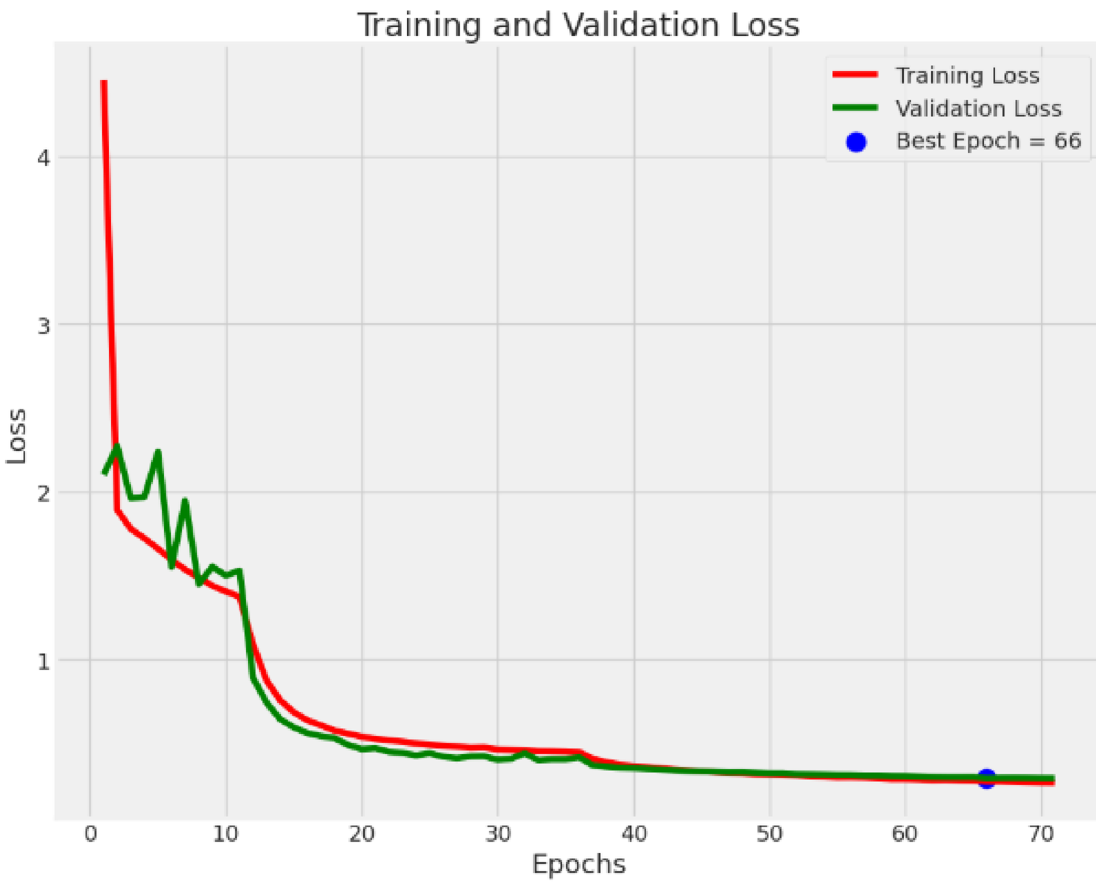

Figure 7.2 Training vs Validation Loss Graph

7.3 Classification Report Analysis

The Table 7.1 represents the classification report of the skin disease prediction model, summarizing precision, recall, F1-score, and support for each of the eight skin disease classes. The model achieves an overall accuracy of 96% on 20,600 test samples. The precision, recall, and F1 scores for individual classes are consistently high, with most values exceeding 0.90, indicating strong predictive performance. Certain classes, such as DF, SCC, and VASC, attain perfect scores (1.00) across all metrics, demonstrating flawless classification. The macro average and weighted average F1-scores are both 0.96, signifying balanced performance across all classes. The slight variation in recall for NV (0.82) suggests a comparatively lower sensitivity in detecting this class. Overall, the model exhibits excellent classification capability with minimal misclassification.

Table 7.1 Performance Analysis of Skin Disease Prediction System for Different Classes

Skin diseases	Precision	Recall	F1-score	Support
Actinic Keratosis (AK)	0.99	1.00	1.00	2582
Basal Cell Carcinoma (BCC)	0.94	0.99	0.96	2665
Benign Keratosis like Lesions (BKL)	0.95	0.98	0.97	2582
Dermatofibroma (DF)	1.00	1.00	1.00	2517
Melanoma (MEL)	0.91	0.90	0.91	2612
Nevus (NV)	0.91	0.82	0.87	2557
Squamous Cell Carcinoma (SCC)	0.99	1.00	1.00	2550
Vascular Lesions (VASC)	1.00	1.00	1.00	2535
accuracy			0.96	20600
Macro average	0.96	0.96	0.96	20600
Weighted average	0.96	0.96	0.96	20600

7.4 Confusion Matrix Analysis

The confusion matrix as shown in Figure 7.3 visualizes the performance of the skin disease prediction model across eight classes: AK, BCC, BKL, DF, MEL, NV, SCC, and VASC. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications. The model exhibits strong performance, with most values concentrated along the diagonal, confirming high accuracy. Notably, AK, DF, SCC, and VASC are classified with zero misclassifications. BCC and BKL show minimal misclassification, with only a few instances predicted incorrectly. The NV class has the highest misclassification rate, with 108 samples misclassified as BCC, 96 as BKL, and 227 as MEL, indicating that these classes share similar features. The confusion matrix highlights the model's overall robustness, aligning with its 96% accuracy, while also identifying areas where class distinction can be further improved, particularly for NV and MEL.

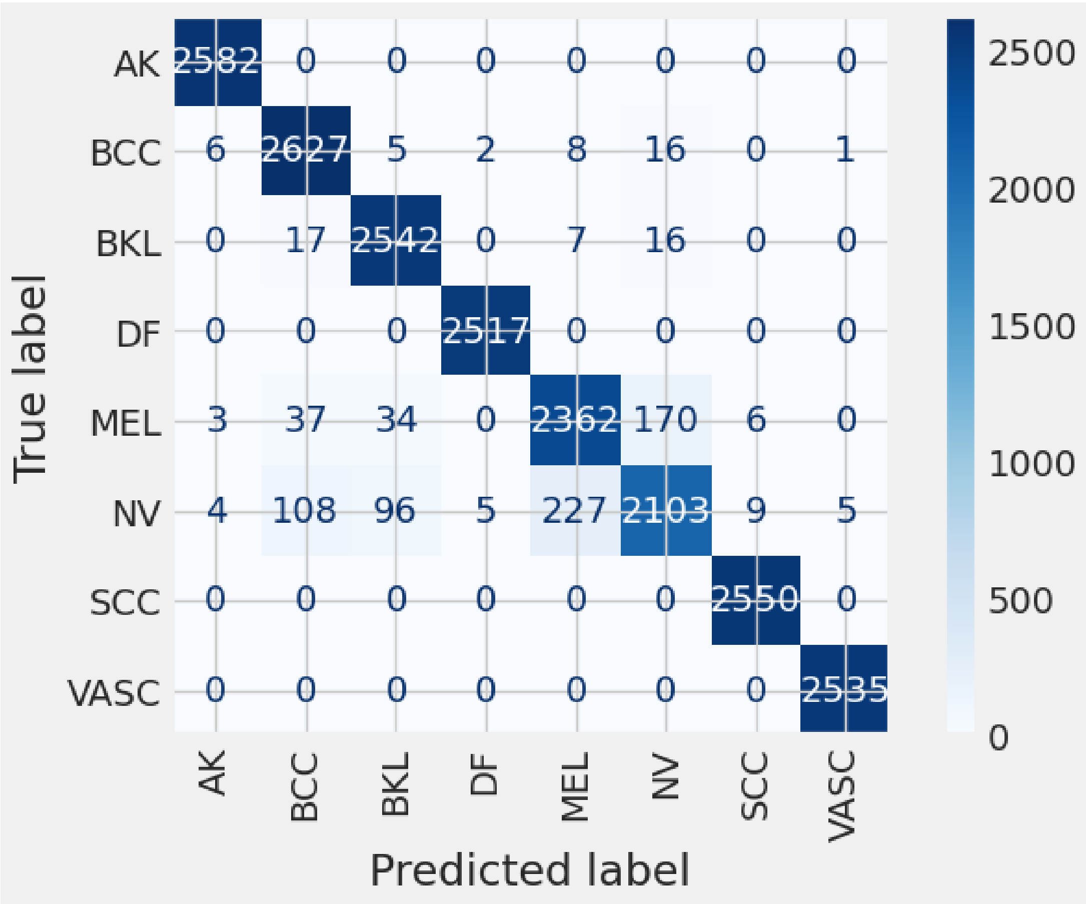

Figure 7.3 Confusion Matrix Showing True vs Predicted Labels for Skin Disease Detection

7.5 Final Model Evaluation

The skin disease prediction model achieved a high classification accuracy of 96%, demonstrating its effectiveness in diagnosing eight different skin diseases. The model was trained using a deep learning architecture implemented in TensorFlow and Python, ensuring robust performance. The evaluation metrics, including precision, recall, and F1-score, indicate that most classes were classified with near-perfect accuracy. The confusion matrix further validates this, with the majority of predictions falling along the diagonal, signifying correct classifications. However, some misclassifications were observed, particularly in the NV and MEL classes, suggesting feature similarities between these conditions. Despite this, the model exhibits strong generalization, as reflected in the minimal difference between training and validation accuracy. Additionally, the loss curves indicate smooth convergence without signs of overfitting, confirming the stability of the learning process. The system's ability to provide real-time predictions with a confidence score enhances its usability in clinical applications. Furthermore, in severe cases, the system recommends booking a doctor’s appointment, making it a practical tool for early skin disease detection. Future improvements can focus on data augmentation, class balancing, and fine-tuning hyperparameters to further enhance the model's performance, particularly for challenging classes.

7.6 User Interface

The developed user interface of the Skin Disease Prediction system is designed to be user-friendly and visually appealing, allowing users to easily interact with the application. The homepage provides an option to check skin diseases online by uploading an image of the affected area. A clear call-to-action button labeled "Check your skin Now" enables users to proceed with the prediction process instantly. Additionally, the interface provides basic information about skin diseases, their causes, and the importance of early diagnosis. The navigation bar includes essential options like booking an appointment, checking features, login, and registration, making it convenient for users to explore various services offered by the application.

Figure 7.4 Homepage for Skin Disease prediction and Online Consultation

7.6.1 Image Upload Section

The Upload Image section is a vital part of the Skin Disease Prediction system. This section allows users to easily upload an image of the affected skin area for analysis. The interface provides an interactive and user-friendly option to either click and upload or drag and drop the image directly into the upload box. Once the image is uploaded, a preview of the selected image is displayed to the user for confirmation.

After selecting the image, the user can click on the "Analyze Image" button, which initiates the prediction process. The model processes the uploaded image and provides the predicted skin disease along with its confidence score. This section ensures that even non-technical users can comfortably use the application for skin disease detection without any complexity.

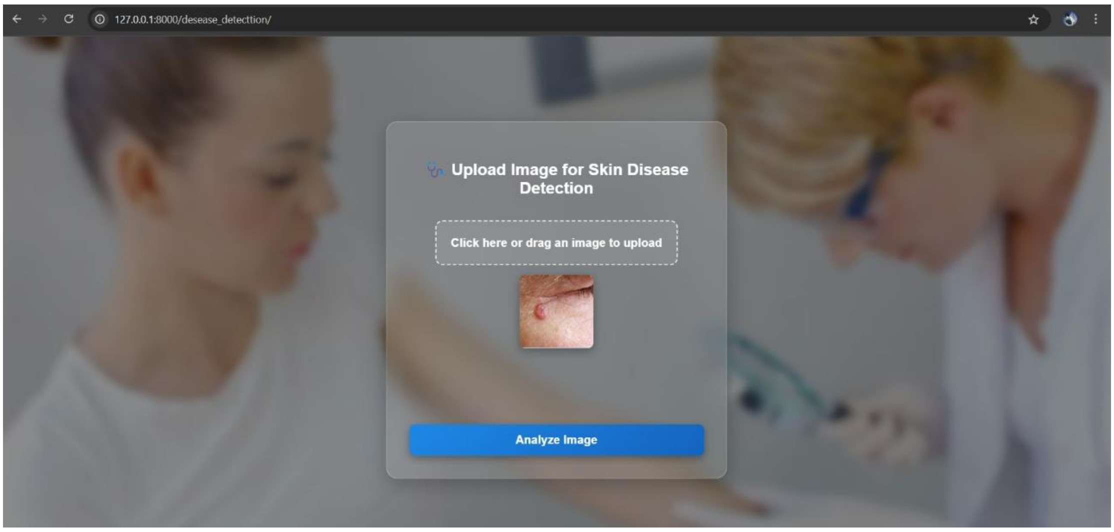

Figure 7.5 Skin Disease Image Upload Interface for Analysis

7.6.2 Predict Result Section

The Predict Result Section is a crucial part of the Skin Disease Detection website, where the result of the uploaded image is displayed after processing through the trained model. Once the user uploads the image and clicks on the "Analyze Image" button, the result section will show the predicted skin disease name, the confidence score in percentage, and the severity level of the disease (Low, Moderate, or High). Along with the prediction result, the uploaded image is also displayed for user reference.

This section provides additional information about the predicted disease, such as its symptoms, causes, and available treatment options. It also offers useful functionalities like uploading another image for a new prediction, downloading the prediction report in PDF format, and booking an appointment with a dermatologist.

These features guide the user to take the next step after knowing their skin condition and encourage them to seek professional medical advice if necessary.

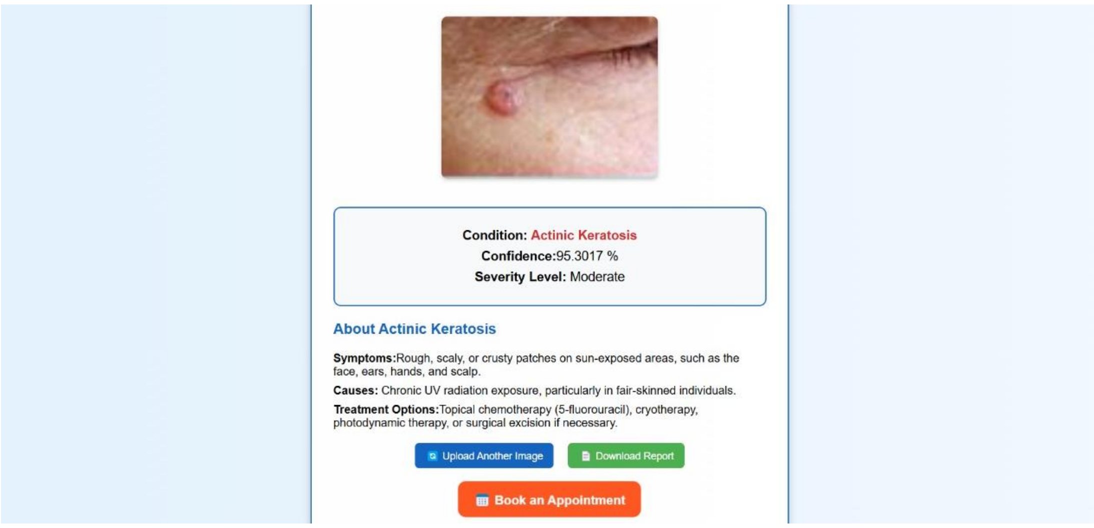

Figure 7.6 Predicted Skin Condition Details

7.6.3 Doctor Appointment Booking Section

The Doctor Appointment Section is an essential feature of the Skin Disease Detection website, designed to help users easily connect with dermatologists based on their prediction results. In this section, users can find experienced skin specialists and book their appointments directly through the platform. It provides filters based on the doctor's experience and language preference, allowing users to search for suitable doctors nearby. Once the filters are applied, the details of the available doctors like name, qualification, experience, languages spoken, availability, address, and contact information are displayed clearly.

This section provides an option to book an appointment instantly by clicking the “Book Appointment” button beside the doctor’s details. It helps users take immediate action by consulting a dermatologist after getting their prediction result. This feature bridges the gap between skin disease detection and medical consultation, ensuring users receive timely treatment advice from certified doctors. It enhances user convenience and makes the platform more helpful and reliable for people seeking expert consultation.

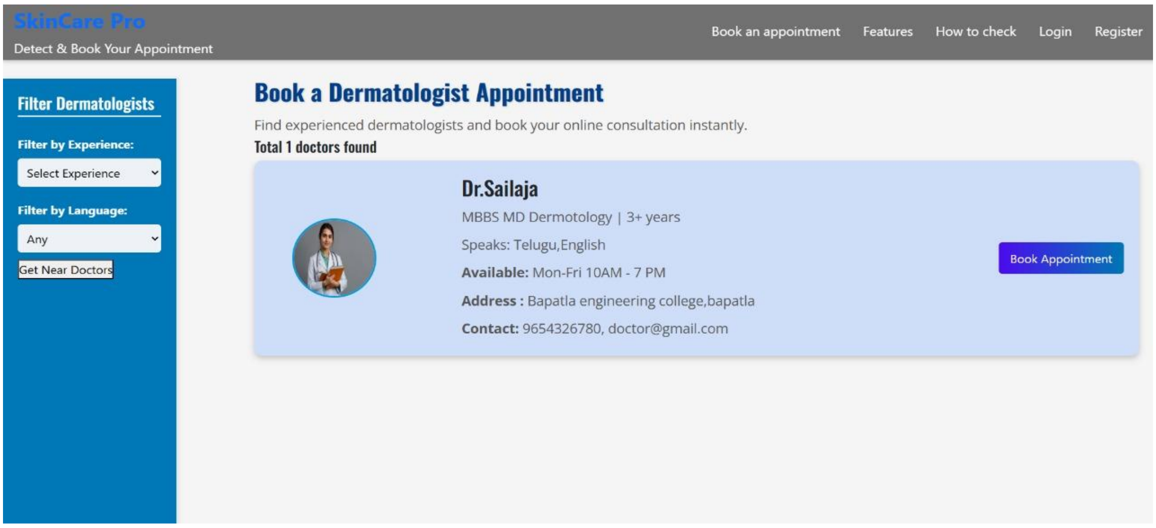

Figure 7.7 Doctor Recommendation & Appointment System

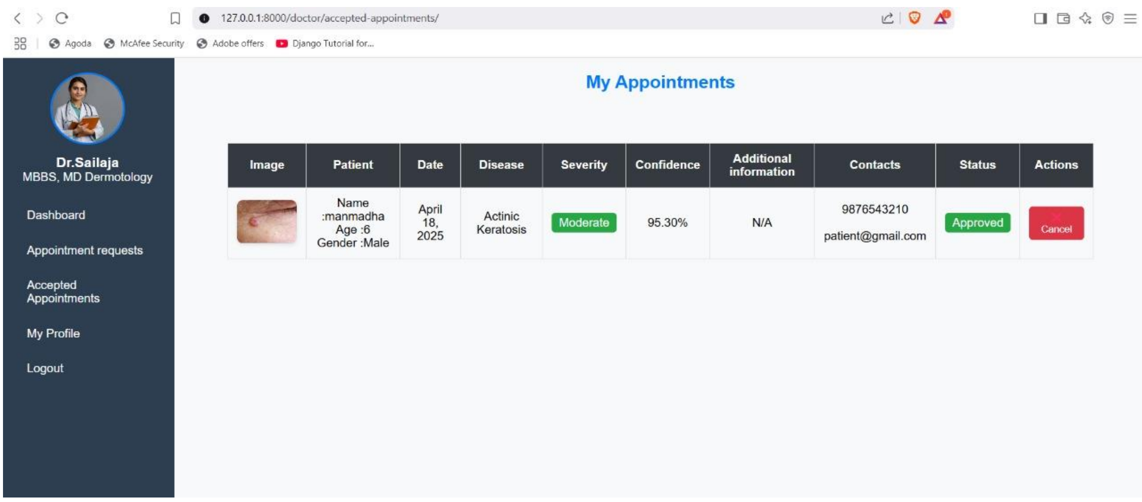

Figure 7.8 Doctor Accepted Appointment Page

8 Conclusion

To conclude, the proposed Skin Disease Prediction system using a CNN-based deep learning model has demonstrated effective performance in classifying 8 different types of skin diseases with an accuracy of 96%. The system enables users to upload real-time skin images and instantly check the prediction results through a simple and interactive web interface. Along with prediction, it also provides treatment suggestions for the detected disease and offers an option to book a doctor appointment, making it more beneficial and user-friendly. However, the major limitation of the current system is that it can classify only 8 specific skin diseases, and sometimes, real-time predictions may not be 100% accurate due to variations in image quality or other factors. In the future, the system can be enhanced by increasing the number of disease categories, using larger and more diverse datasets, applying transfer learning techniques, and conducting real-world testing for better accuracy and reliability. Additionally, further features like maintaining past prediction history, and integrating chatbot support to guide users can also be included. These future enhancements will help in delivering a more advanced and supportive healthcare solution, improving the overall user experience and making the system even more valuable in real-time medical applications.

9 References

[1] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118. https://doi.org/10.1038/nature21056

[2] Brinker, T. J., Hekler, A., Utikal, J. S., Grabe, N., Schadendorf, D., Klode, J., & von Kalle, C. (2019). Skin cancer classification using convolutional neural networks: Systematic review. Journal of Medical Internet Research, 21(9), e14666. https://doi.org/10.2196/14666

[3] Yu, K., Zhang, Y., & Li, Q. (2020). Skin lesion classification with ensemble deep learning models. BMC Medical Imaging, 20(1), 1-9. https://doi.org/10.1186/s12880-020-00456-8

[4] Kassani, P. H., Khandelwal, R., & Choudhary, A. (2019). Skin cancer classification using convolutional neural networks: A survey. IEEE Access, 7, 183072-183085. https://doi.org/10.1109/ACCESS.2019.2954213

[5] Debelee, T. G. (2023). Skin lesion classification and detection using machine learning techniques: A systematic review. Diagnostics, 13(19), 3147.

[6] Liu, X., Wang, Y., & Zhang, Q. (2021). Deep learning-based image segmentation and classification for skin cancer detection. IEEE Transactions on Medical Imaging, 40(2), 456-467. https://doi.org/10.1109/TMI.2021.3045678

[7] Jain, S., Gupta, R., & Verma, P. (2022). Hybrid deep learning model for skin disease classification using CNN and transfer learning. Journal of Biomedical Informatics, 130, 104086. https://doi.org/10.1016/j.jbi.2022.104086

[8] Sharma, A., Kumar, R., & Singh, M. (2023). Ensemble deep learning for skin lesion classification: Integrating ResNet and EfficientNet. Medical Image Analysis, 85, 102695. https://doi.org/10.1016/j.media.2023.102695

[9] Yu, K., Zhang, Y., & Li, Q. (2020). Skin lesion classification with ensemble deep learning models. BMC Medical Imaging, 20(1), https://doi.org/10.1186/s12880-020-00456-8